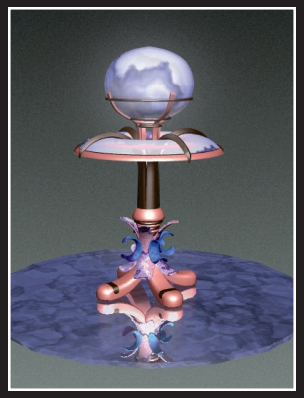
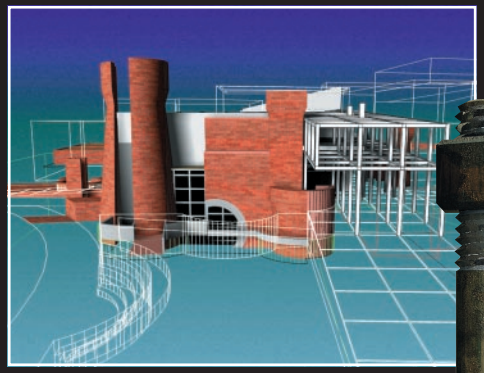








autodessys INC

2011 Riverside Drive
Columbus, Ohio 43221
www.formz.com

Product information & support
(614) 488-9777

TEL : (614) 488-8838
FAX : (614) 488-0848
formz@formz.com

formZ RenderZone

Demo Mini User's Manual

form•Z RenderZone Demo
Mini User's Manual

Table of Contents

autodessys INC

2011 Riverside Drive
Columbus, Ohio 43221
www.formz.com

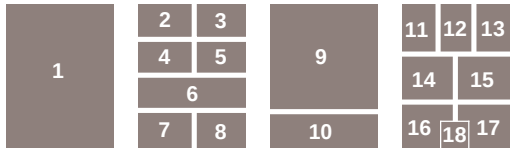
Phone: (614) 488-9777
Fax: (614) 488-0848
formz@autodessys.com

form•Z RenderZone Demo Mini User's Manual
THIRD EDITION: April 1997

CREDITS: This Manual was created electronically using Adobe PageMaker® on an Apple Macintosh®.
All illustrations in this Manual were created and rendered using **form•Z RenderZone**.

PRINTED in the United States of America

COVER CREDITS



1. Cover design by Alberto Scirocco, auto•des•sys, Inc., Columbus, Ohio.
2. Orange Basket; 3. Toaster; and 4. Coins, by Mark Schneider, Eureka, Missouri.
5. Indy Car, by Todd Siechen, RealEyz Imaging, Aptos, California
6. Harbor Point Apartments, designed by Harden, Reid, Jahnes, and Luchtenberg Architects, computer modeling and rendering by Pete Scott, digital Sculpture, Columbus, Ohio.
7. Beach Bar, by Richard Reynolds, Malibu, California.
8. Tintox and Milox, by Christophe Moulin, de Visu, France.
9. Samples of precaptured texture maps shipped with **form•Z RenderZone**.
10. Samples of tree images shipped with **form•Z RenderZone**.
11. Pharaoh Head, by Roger Harris, Kent, England.
12. Lamp, by Michael Del Grosso, auto•des•sys, Inc., Columbus, Ohio.
13. Pots and Pans, by Roger Harris, Kent, England.
14, 15. Landscapes; 16. Projector Light; and 17. Wexner Center, by Markus Bonn, auto•des•sys, Inc., Columbus, Ohio.
18. Bolt, by Alberto Scirocco, auto•des•sys, Inc., Columbus, Ohio.

All models built and rendered with **form•Z RenderZone**

About form•Z RenderZone Demo

WELCOME to **form•Z RenderZone Demo**, the demonstration of the **form•Z** version that includes advanced rendering provided by LightWorks®. Please note that this is a partially disabled version of the program that is not intended for production work. While it allows you to open files, including image files that may be used to map textures on objects, it does not allow you to save. Also note that this Mini Manual only covers the rendering features. The modeling operations are discussed in the **form•Z Mini Manual.**

form•Z RenderZone overview

form•Z RenderZone offers seven levels of rendering: ***flat***, ***Gouraud***, ***Phong***, ***preview z-buffer***, ***full z-buffer***, ***preview raytrace***, and ***full raytrace***. Flat rendering results in the fastest renderings, with limited quality. Raytrace gives the best image quality, but takes the most time to execute. Most of the time, z-buffer produces excellent results at very reasonable speeds; thus it is the rendering type that is expected to be used most frequently. Given that accurate transparencies and reflections can only be generated with raytracing and not with the z-buffer algorithm, when such effects are applied and a z-buffer rendering is produced, a mixed method is actually applied. While the efficiency of the z-buffer is used to determine the closest pixel, reflections and transparencies are generated by casting rays from pixels on reflective and transparent surfaces.

All levels can be rendered with one or more lights, which can be of four types: ***distant*** (sun), ***cone***, ***point*** or ***projection*** (gel) light. Full z-buffer and full raytrace are ***antialiased***. Antialiasing at other rendering levels can be simulated by producing images significantly larger than the size at which they will actually be printed. ***Smoothly shaded*** images are produced by all levels except the flat shaded rendering. ***Highlights*** are produced by all the levels above and including Phong. ***Shadows*** are produced by all the levels above Phong. ***Texture mapping***, which can be ***procedural*** or ***pre-captured***, ***environment mapping***, ***bumps***, ***depth effects***, and ***backgrounds*** can be applied starting with the preview z-buffer level. ***Transparencies*** can be applied at the full z-buffer and the full raytrace levels.

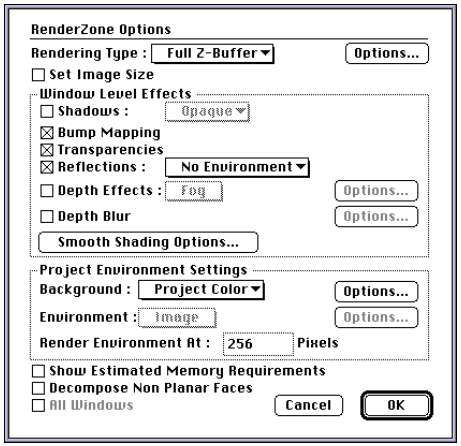
How the rendering effects are created

Rendering with **form•Z RenderZone** involves the following steps:

- Surface styles are created or selected from a set of predefined surface styles and are displayed in the Surface Styles palette. Surface styles are what colors are in the regular version of the program, except that they incorporate textures.
- The active surface style is assigned to an object when it is created. Surface styles can also be assigned to objects or faces of objects with the Color tool.
- The Texture Map tool is used to specify the exact position and size of a texture, and the type of texture mapping (***flat, cubic, cylindrical, spherical, UV,*** or ***parametric***). When objects are created, a default texture mapping is assigned.
- Surface styles can also be attached to surfaces of objects as ***decals*** on top of and in addition to previously assigned surface styles, using the Decal tool.
- One or more lights are generated and positioned.
- Once the surface styles are defined and assigned to the objects and/or their surfaces, and the lights are positioned, a rendered image is created by selecting the **RenderZone*** item from the **Display** menu. Rendering options are selected from the **RenderZone Options** dialog that can be invoked directly from this item.

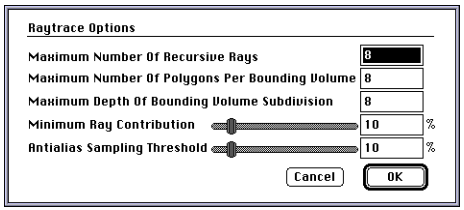
Producing a rendered image

The model displayed in the active window of **form•Z** can be rendered by clicking on the **RenderZone*** item in the **Display** menu. The rendering parameters are set in the **RenderZone Options** dialog, invoked by in the usual manner (click on the item while pressing the ***option*** key on the Macintosh or the ***ctrl+shft*** keys on Windows). This dialog allows you to select the type of rendering and to turn on or off a variety of rendering effects.



Rendering types: One of seven available rendering types is selected from the **Rendering Type** pop up menu at the top of the dialog: **Flat**, **Gouraud**, **Phong**, **Preview Z-Buffer**, **Full Z-Buffer**, **Preview Raytrace**, and **Full Raytrace**.

When one of the z-buffer or raytrace methods is used, additional parameters can be selected by clicking on the **Options...** button next to the **Rendering Type** menu. These parameters control the performance of raytraced images.



By default, the raytrace parameters are set to their optimal values for an average scene. Therefore, in general it is not necessary to further manipulate them. However, for certain scenes, increased performance or quality can be achieved by carefully adjusting the raytrace parameters.

Setting the image size: Normally, the complete modeling scene, as is displayed in a window, will be rendered. The **Set Image Size** option can be selected to render only part of the scene. When it is, before starting the execution of the rendering, **form•Z** allows you to delineate the area of the model you wish to render.

Decomposing non planar faces: The **Decompose Non Planar Faces** option is available at the bottom of the **RenderZone Options** dialog. When selected, all non planar faces that may be contained in the modeling scene will be decomposed into triangles. If not selected, all surfaces will be rendered as they are.

form•Z RenderZone will render non planar surfaces and does not require them to be triangulated. However, such surfaces are not guaranteed to be rendered correctly. It is therefore recommended to turn on the **Decompose Non Planar Faces** option when a scene is known to contain non planar surfaces. On the other hand, if a scene contains planar faces only, turning off the **Decompose Non Planar Faces** option typically results in significant speed gains, since the program does not have to check the planarity of all the faces in the scene.

Setting rendering and environment effects

The **RenderZone Options** dialog contains two groups of options, titled **Window Level Effects** and **Project Environment Settings** that are used to turn on and off rendering effects. The first group of options applies at the ***window*** level, while the second applies at the ***project*** level. That is, options of the first group, such as shadows and bumps, are set individually for each window that may be open. Thus, the same scene may be rendered in one window with shadows, and in another without. The options in the second group apply at the project level and affect all the windows of a project. For example, the same background will appear in all windows.

Shadows: When this option is selected, and the rendering level is set to **Preview Z-Buffer** or above, objects that have been assigned the shadow attribute will cast shadows from lights whose shining and shadow attributes are on. When hard (raytraced) shadows are cast and an object is transparent, its shadow may be **Opaque** or **Transparent** which is selected from the **Shadows** pop up menu.

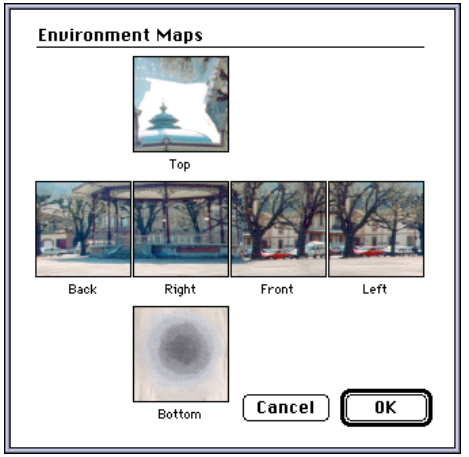
Bump Mapping: When this option is on, all surfaces which have been assigned a surface style that includes bumps will be rendered with bumps.

Transparencies: When this option is selected, all surfaces which have been assigned a surface style that includes transparencies will be rendered transparent.

Smooth Shading Options...: Clicking on this button invokes the **Smoothing Options** dialog, where you can choose to use the object smoothing attributes, to override them, or to combine them with additional smoothing parameters selected from this dialog.

Reflections: When this option is off, all surfaces are rendered without reflections. If it is on, the type of reflection must be selected from the pop up menu next to it. The available options are **No Environment**, **Cubic Environment**, **Spherical Environment**, and **Rendered Environment**.

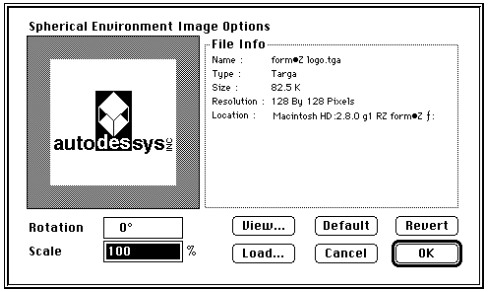
When **Cubic** or **Spherical Environment** is selected, reflective faces will reflect procedural patterns or images mapped onto an imaginary cube or sphere, respectively. These patterns or images are selected from the **Environment** pop up menu found in the **Project Environment Settings** group of options. If **Image** is selected, precaptured images need to be placed on the faces of the environment cube or on the surface of the environment sphere. This is done through the **Environment Maps** (for cubic) or the **Spherical Environment Image Options** (for spherical) dialogs. They are invoked by clicking on the **Options...** button, when **Image** is selected in the **Environment** menu.



When using a z-buffer or raytrace method to render an image, all reflective surfaces will show the portion of the environment toward which they are oriented. The environment cube or sphere themselves are not rendered. They only appear in the reflections of the surfaces.

Loading and manipulating image maps

form•Z RenderZone is shipped with 75 precaptured textures (see samples on back cover), and additional texture images may be produced by the user. Precaptured images may be used by a variety of **form•Z RenderZone** procedures. They may be used to create environment maps by placing them on the surfaces of the environment cube or sphere; they may be used as a background (page 8); they may be used to attach precaptured images to surfaces of objects by using one of the **Color** shaders (page 12); they may be used as transparency filters by one of the **Transparency** shaders (page 15); and they may be used to delineate bumps through one of the **Bump** shaders (page 16). In all these cases, the loading of an image and other available operations are executed through an **Image Options** dialog. This dialog consists of an image view area, a file information area, **Cancel** and **OK** buttons, and four buttons that allow you to locate an image file, to reset the image, and to view and crop the currently loaded image.



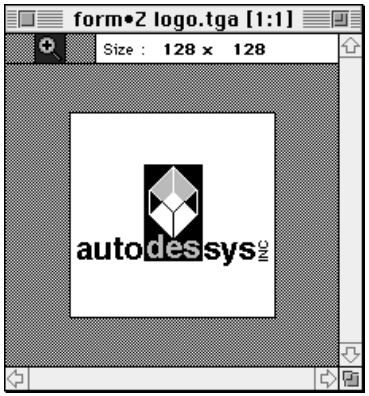
File Info is a non editable area with information about the image.

Default resets the loaded image to the system default image file.

Revert resets all the parameters to the values they had when the dialog was opened.

Load... invokes the Open File dialog to select the image to be loaded.

Clicking on **View...** invokes the **View** window which displays the currently loaded image bounded by a rectangular frame marked with little square handles. The three tools on the top of the window are used to manipulate the image, as follows:



With this tool selected,

- clicking on a corner handle and dragging it repositions the corner;
- clicking on a mid-segment handle and dragging it repositions the segment;
- clicking on a segment (away from a handle) and dragging it moves the whole rectangle.

Clicking inside the image enlarges it.

This icon appears when pressing the ***option*** key on the Macintosh or the ***ctrl+shft*** keys on Windows, and reduces the image.

Clicking and dragging pans the image.

All moves are restricted to within the boundaries of the image.

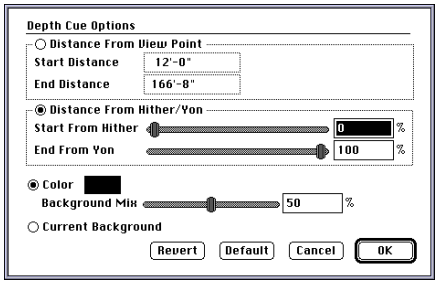
Environmental reflections

Accurate reflective effects are produced with raytracing, when using one of the reflection shaders, such as **Mirror**. The **Environment** shader is also available, which produces simulated reflections and is generally faster.

When the **Environment** shader has been included in the surface styles assigned to one or more objects in a scene, and the **Rendered Environment** item is selected in the **Reflections** menu in the **Window Level Effects** group of options, reflections are simulated through the use of the environment cube. Using as view point the center of all the surfaces to which the **Environment** shader has been assigned, views of the other objects (those that are not assigned the **Environment** shader) are generated looking into the direction of each of the squares of the environment cube. These directions are the X, Y, Z axes of the world space, and views are taken in both their positive and negative directions. These images are then mapped onto the squares of the environment cube, the same way precaptured images are mapped when the **Cubic Environment** option is used. Note that raytraced reflections do not have additional memory requirements.

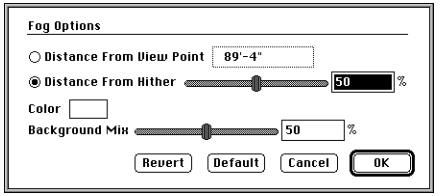
Depth effects

When **Depth Effects** is off, no depth effects are applied to the rendering. If it is on, the type of effect is selected from the pop up menu next to it.

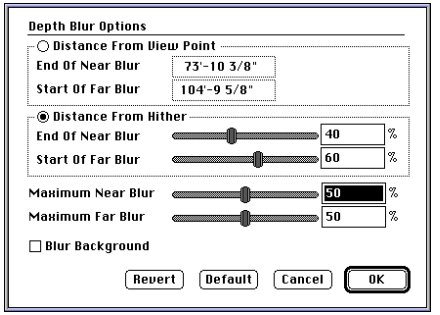


- **Depth Cue** blends a color with the surfaces of objects that lie within a region defined either relative to the view point or relative to the clipping (hither and yon) planes. One of two methods can be selected from the **Depth Cue Options** dialog.

Distance From View Point defines the extents of the depth region as distances from the viewpoint. **Distance From Hither/Yon** defines the depth region relative to the hither and yon plane.

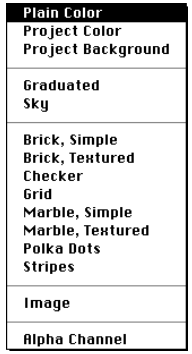


- **Fog** produces fog effects. In the **Fog Options** dialog the basis of the fog may be set as a distance from the view point, or relative to the position of the hither plane. The fog effect is the strongest when its base is close to the hither plane (**Distance From Hither** = 0%). The farther from the hither plane, the weaker the fog effect is.



- **Depth Blur** creates a blur effect and objects in front and beyond a focused area appear fuzzy. The blur region can be defined relative to the viewpoint or the hither plane.

Setting the background



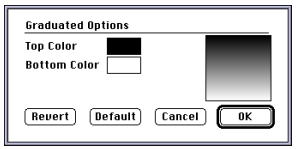
A background color or image is generated from the **Background** pop up menu. Background colors or textures, which can be either procedural or precaptured, are generated by shaders, the same way shaders are used to define surface styles that are mapped onto objects. The parameters of the currently selected background are set in the dialog invoked by clicking on the **Options...** button next to it.

• **Plain Color** produces a background with a uniform color, which can be changed in the Color Picker dialog, invoked by clicking on the **Options...** button.

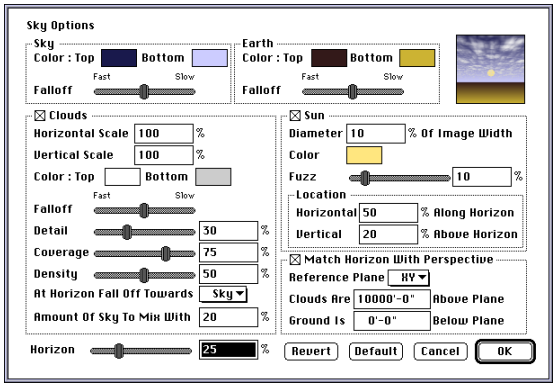
• **Project Color** also sets the background to a plain color, which is the same as that selected from the **Color** dialog invoked from the **Options** menu.

• **Project Background** causes the background to be rendered as it is in any other rendering modes, including grid, axes, and underlay. Its one option of **Render Grid And Axes As 3D Lines** renders the grid and axes as 3D entities. As a result, if the grid is at a location where it passes through the middle of an object, the rendering will show the grid lines intersecting properly with the object.

• **Graduated** produces a background which is a mix of two colors, the **Top** and **Bottom Color** set in the **Graduated Options** dialog. These colors are changed by clicking on them to invoke the Color Picker dialog. The two colors gradually phase into each other.



• **Sky** renders a background that consists of sky and earth areas and may optionally include clouds and sun. For each there are a number of parameters which combine to produce a large variety of sky images.



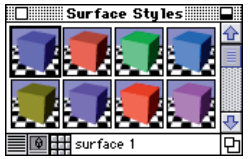
• **Image** generates a background from an image file. Clicking on the **Options...** button invokes the **Background Map Options** dialog. The background image may be scaled to fit the window, or may be tiled.

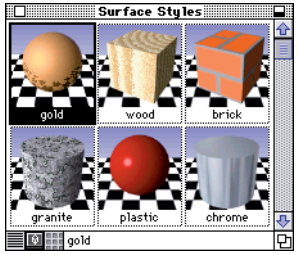
• **Alpha Channel** is a special background which appears black on the screen but is actually empty (transparent). This background also stores with the image the transparency ratio for pixels which facilitates compositing rendered images with a variety of backgrounds in image processing applications.

Surface styles

In **form•Z RenderZone**, surface styles are the equivalent to the colors in the regular version of the program. Plain color is actually the simplest form of a surface style. A complete surface style is displayed by superimposing on a color additional types of attributes, which control how a surface interacts with lights and the environment. Specifically, a surface style consists of four classes of attributes applied through distinct sets of ***shaders: color, reflection, transparency,*** and ***bumps.***

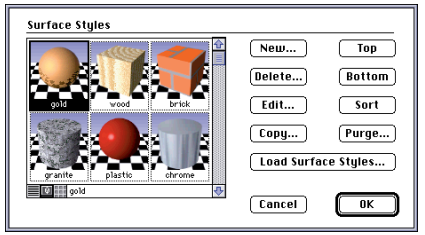
Color shaders are used to define the color of a surface, which may be a plain color, a procedural texture, or a pre-captured image. The behavior of a surface when illuminated by a light is determined by a ***reflection shader***. Shaders of this class may be thought of as defining the finish of a surface, and are used to simulate material properties such as matte, metal or plastic. ***Transparency shaders*** define how transparent or opaque a surface is, and how much light passes through it. Small surface disturbances can be created using ***bump shaders***. Typically, a bump shader gives an otherwise smooth surface an irregular or indented appearance.





Surface styles are displayed in the **Surface Styles** palette. When the program is launched for the first time, the surface styles appear in their default plain colors. If a previously saved project file, for which surface styles were defined, is opened, those styles are displayed. Three formats are available for showing the surface styles: a ***large image***, a ***small image***, and a ***name list*** display. They are selected from icons on the lower margin of the palette. In the palette, surface styles are displayed on 3D objects (planes, cubes, cylinders, or spheres) which are determined at the time a surface style is defined. One of the surface styles is the ***active*** surface style and is highlighted.

Surface styles are managed through the **Surface Styles** dialog (invoked by ***option*** clicking (Macintosh) or ***ctrl+shft*** clicking (Windows) in the Surface Styles palette), which contains a display area identical to the palette, and a number of buttons: **New...** creates a new surface style by invoking the **Surface Style Parameters** dialog (see next page). New surface styles can also be created from the Surface Styles palette.

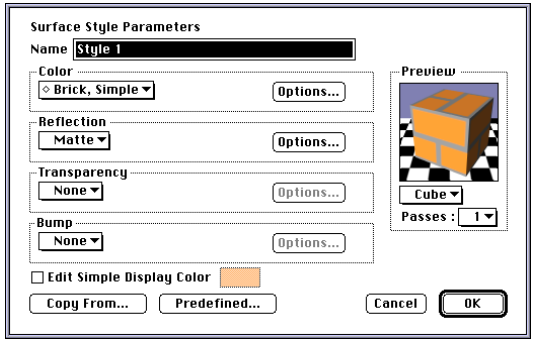


Delete... deletes the active surface style. **Edit...** displays the parameters of the active surface style in the **Surface Style Parameters** dialog, where they can be changed. **Copy...** duplicates the active surface style.

The surface styles are initially stored in the order in which they are created. This order can be changed: **Top** and **Bottom** move the active surface style to the top or bottom position in the list. **Sort** sorts the list of surface styles alphabetically, according to their names. Surface styles not in use are deleted with **Purge...**.

Load Surface Styles... invokes the Open File dialog for selecting a **form•Z** file, whose surface styles will be brought into the active project.

Editing surface styles



Surface styles are set in the **Surface Style Parameters** dialog, invoked by the **New...** and **Edit...** buttons (**Surface Styles** dialog). It can also be invoked by double clicking on a surface style icon or name in the Surface Styles palette.

The **Surface Style Parameters** dialog consists of the following parts:

• A **Name** text field at its top, in which a surface style name can be typed to replace that defaulted by the system. A name can be at most 31 characters long.

• Four classes of shaders, each consisting of a pop up menu from which a shader can be selected. Clicking on the **Options...** button next to each menu invokes a dialog with the parameters that affect the currently selected shader.

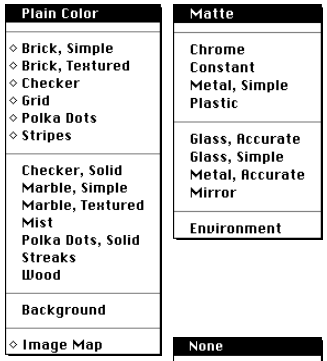
• A **Preview** area displaying the effects of the currently selected shader.

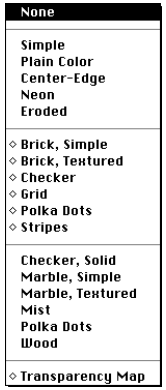
• An **Edit Simple Display Color** option from which the color used by lower end rendering modes, such as wire frame, can be defined.

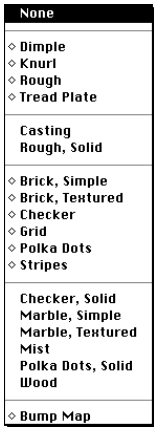
• A **Copy From...** and a **Predefined...** button command.

Types of shaders: There are four classes of shaders that control ***color***, ***reflection***, ***transparency***, and ***bumps***. The effects producing shaders are either ***procedural*** or ***precaptured texture maps***. The procedural shaders are further distinguished into two types: ***wrapped*** (flat) or ***solid shaders***. The precaptured shaders are always flat. In the shaders menus, the wrapped shaders are denoted with a diamond to the left of their name.

Previewing the effects of the shaders: At the upper right corner of the **Surface Style Parameters** dialog, there is a preview image which displays the combined effects of the shaders and the parameters which are currently selected. Each time a new shader is selected or a shader parameter is changed, the image is re-rendered and displays the new effects. The surface styles are previewed by applying them to the surface of a **Plane**, a **Cube**, a **Cylinder**, or a **Sphere**, which is selected from a menu under the preview image. The preview object is shown on a checkered floor, which helps preview the effects of reflective and transparent surface styles. The preview image can be generated through a single rendering pass, or up to four passes may be used. This is selected from the **Passes** pop up menu.







How procedural textures are mapped: The wrapped texture shaders produce two dimensional patterns which are mapped onto the surfaces of objects. Exactly how they are mapped onto a surface can be determined using the Texture Map tool (page 18). If this tool is not used, a global default texture mapping is used. The default texture mapping parameters are set in the **Texture Map Options** dialog.

The wrapped textures are mapped onto a surface by repeating a rectangular tile which carries a module of a texture. Arranging a number of modular tiles horizontally and vertically produces the overall pattern of a texture. These tiles can be of different sizes, which is determined by the number entered in the **Wrapped Textures Horizontal** and **Vertical Tiling Size** fields in the **Texture Map Options** dialog. The values entered in these **Size** fields have an effect only when a wrapped surface is positioned by default. When the Texture Map tool is used to accurately position a wrapped texture on an individual object, the **Size** parameters that are set in the **Texture Map Controls** dialog for that object override the global default **Size** parameters entered in the **Texture Map Options** dialog.

Solid shaders produce 3D patterns that result from the intersection of an object's surfaces with the 3D textures. Thus solid textures show continuity around edges and appear to have been carved out of a solid block of the material which the shader is generating. The unit size or scale of a solid texture is controlled by the **Solid Textures: Size** parameter in the **Texture Map Options** dialog. Contrary to the tile size of the wrapped textures where independent X and Y dimensions can be defined, a single uniform size applies to the three dimensions of the solid textures. As with the wrapped textures, the value entered in the **Solid Size** field of the **Texture Map Options** dialog has an effect only when a solid texture is positioned by default. When the Texture Map tool is used to accurately position a solid texture on an individual object, the **Size** parameter that is set in the **Texture Map Controls** dialog overrides the default global **Size** parameter entered in the **Texture Map Options** dialog.

The shaders dialogs: Next to each of the four menus of shaders there is an **Options...** button. Clicking on it invokes the dialog of the active shader. Each shader has a distinct dialog, but all dialogs share some common characteristics.

Three types of dialogs are invoked. The **Plain Color** shaders found in the **Colors** and **Transparency** menus invoke the standard Color Picker dialog. The texture map shaders, namely **Image Map**, **Transparency Map** and **Bump Map**, invoke a texture map dialog which includes a preview of the texture map and a command for loading a new texture. The dialogs of the procedural shaders contain a two dimensional preview of the texture. Four types of parameters are used:

• The ***scale*** parameters contain scaling factors whose default value is 100%. Using a value other than 100% scales a texture up or down.

• The ***texture size*** parameters, found in dialogs of wrapped textures, define the size of the texture module, are expressed as percentages, and are relative to the tile size.

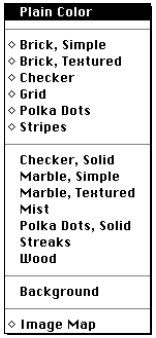
• ***Color*** boxes displaying colors used to generate a texture are found in most of the shader dialogs. Clicking on them invokes the Color Picker dialog.

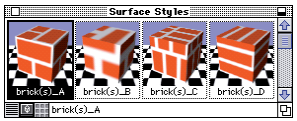
• The ***slider*** parameters are set by sliding a knob on a bar, or by clicking on a position on the bar, or by entering a number in the field next to them.

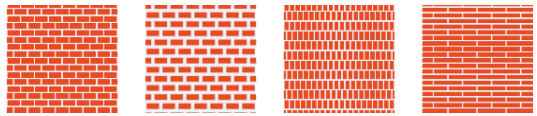
All shader dialogs also contain button commands, namely **OK**, **Cancel**, **Default**, and **Revert**. **Default** resets all the parameters in the dialog to their default values. **Revert** resets all the parameters to the value they had when the dialog was opened.

Color shaders

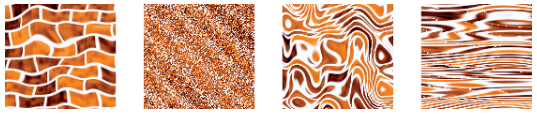
The **Color** shaders in the **Surface Style Parameters** dialog contain five groups of shaders: ***plain color***, ***wrapped procedural textures***, ***solid procedural textures***, ***background color***, ***and precaptured image textures.*** Examples of a few shaders and the textures produced by different combinations of parameters are shown below. We recommend that you experiment with all the shaders and observe the results produced by different settings.



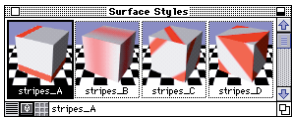


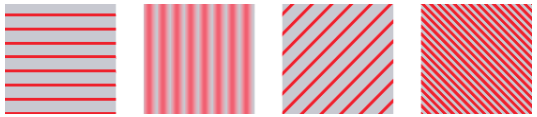


Brick, Simple	1	2	3	4
Scale	100	100	100	100
Brick Width	90	90	30	200
Brick Height	40	40	60	30
Mortar Thickness	10	25	10	10

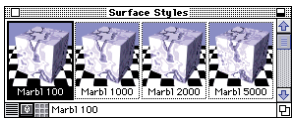


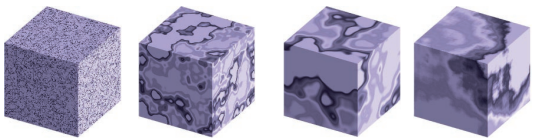
Brick, Textured	1	2	3	4
Scale	200	200	200	200
Brick Width	90	90	90	500
Brick Height	40	40	40	10
Mortar Thickness	10	10	20	10
Rough Scale	500	50	500	500
Rough Amplitude	500	5000	5000	5000



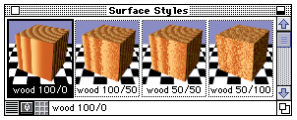


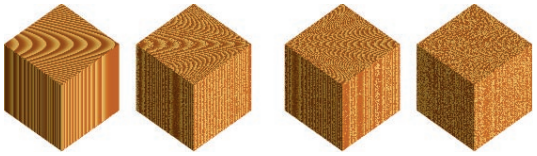
Stripes	1	2	3	4
Size	100	100	100	40
Width	20	20	20	50
Fuzz	0	100	10	10
Stripe Angle	0	90	45	-45





Marble, Simple	1	2	3	4
Scale	100	1000	2000	5000
Detail	30	30	30	30

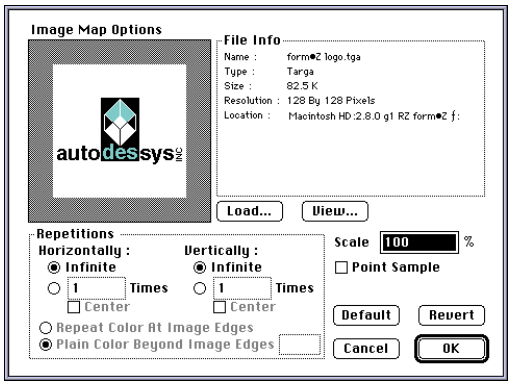




Wood	1	2	3	4
Scale	100	100	50	50
Detail	0	50	50	100

Texture maps

Selecting the **Image Map** item from the **Color** menu invokes the **Image Map Options** dialog. The upper part of this dialog is identical to the dialog discussed on page 6. The lower part includes some additional options.



The **Repetitions** parameters determine how many times a texture map is repeated, how it is centered, and the color used beyond the repeated textures. When **Infinite** is selected, the image is repeated as many times as necessary to cover a surface, in the respective direction. If **Times** is on, the number of repetitions is typed in its field.

Texture Map Control:					
Center	**Horizontal:**	No	No	No	Yes
	Vertical:	No	No	No	Yes
Image Map Options:					
Center	**Horizontal:**	No	Yes	No	Yes
	Vertical:	No	Yes	No	Yes
Times:	**Horizontal:**	2	2	3	3
	Vertical:	2	2	3	3

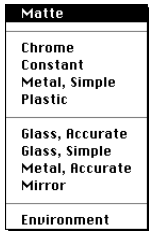
The **Center** option is available only when **Times** is on for the respective orientation (horizontal or vertical). When off (default), the image map texture is generated by starting at the origin and repeating the tiles in the positive direction only. When on, the tiles are repeated in both the positive and negative directions. Note that the arrangement of the tiles is also affected by the **Center** options in the **Texture Map Controls** dialog, which determine whether the center of a tile or the lower left corner of the tile is placed on the texture's origin when the texture is generated. When the **Center** option of the **Image Map Options** dialog is on, and an even number is entered in the **Times** field, then half of the tiles will be arranged in the positive and half in the negative direction. If an odd number is entered in the **Times** field, then one more tile will be arranged in the positive direction than in the negative. For example, if 5 is entered in the **Times** field, then 3 tiles will be arranged in the positive direction and 2 tiles in the negative.

For a perfect centering of an image map, the **Center** parameter in the **Image Map Options** dialog should be correlated with the **Center** parameter in the **Texture Map Controls** dialog. To center an even number of tiles, the **Texture Map Controls**' **Center** should be off. To center an odd number of tiles, it should be on. These conditions apply independently for the horizontal and vertical orientations.

Whenever the generated texture does not cover a surface completely, two options available in the **Image Map Options** dialog determine how the portion of the surface that is not covered by the texture will be treated. **Repeat Color At Image Edges** colors the area of a surface which is not covered by a texture with the color of the pixels at the edge of the image. **Plain Color Beyond Image Edges** fills the surface beyond the texture with the plain color shown in the color box. This color can be changed by clicking in the color box to invoke the Color Picker dialog.

When **Point Sample** is selected, the texture map is point sampled when rendered. This results in a lower rendering quality but faster speed. If this option is off (default), the texture map is area sampled, resulting in better image quality.

Reflection and transmission shaders



Shaders in this class determine how light is reflected off of a surface. The desired shader is selected from the **Reflections** pop up menu, which contains four groups:

- The first group contains the single shader, **Matte**, which is the default and produces no reflections.
- The second group contains four shaders that produce no reflections but depict specific materials, such as metal and plastic.
- The third group contains four shaders that produce true reflections, calculated by a ray tracing procedure, even when a z-buffer rendering is produced.
- The fourth group contains a single shader, **Environment**, which produces simulated reflections through the use of an environment cube (see page 5).

All shaders in this class use some combination of the following parameters:

Ambient Reflection indicates how much of the ambient light intensity is reflected off a surface. Since ambient light is distributed uniformly in a scene, if this factor is high, the contrast between illuminated areas and areas in shade decreases.

Diffuse Reflection determines the percentage of light that is reflected uniformly in all directions. Rough surfaces usually have a high diffuse reflection value, whereas smooth and glossy surface have low diffuse reflection values.

Specular Reflection determines how much of the light is reflected at the incoming angle. This reflection generates highlights on curved surfaces.

Glow causes surfaces to appear brighter than if only ambient, diffuse, or specular reflection parameters were used.

Specular Color is the color at the center of specular highlights. At the perimeter of a hot spot, the specular color blends with the surface color.

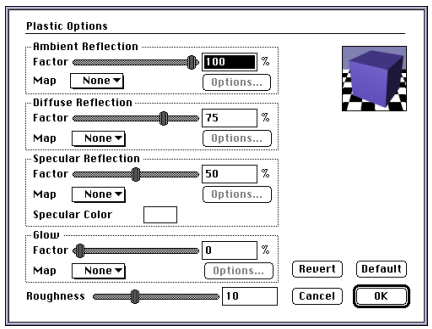
Reflection determines the level of reflection for reflective shaders. 100% causes a surface to receive all of its color from the surrounding environment.

Transmission causes a ray to pass through a surface, rather than being reflected off it. Transmitted rays are bent according to the refraction parameter.

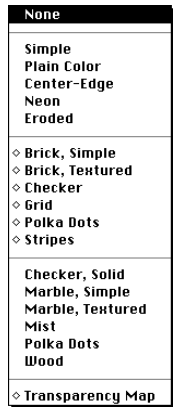
Roughness controls the size of the highlights. With high roughness, highlights become larger, whereas with low roughness they are small.

Refraction determines how rays that are transmitted through a surface are bent. A value of 1.0 indicates no refraction. Water's index of refraction is about 1.5, and no real material has a refraction value greater than 2.5.

All of these parameters are typically applied uniformly across a surface. However, a shader can also be selected from the **Map** pop up menu. The shader generates a gray scale pattern, which is used to distribute the effects of a parameter. Black areas of the pattern have no effect, while white areas have full effect. Such shaders can be applied to all the parameters except the last two. An example of a dialog for a reflection shader is shown to the right.

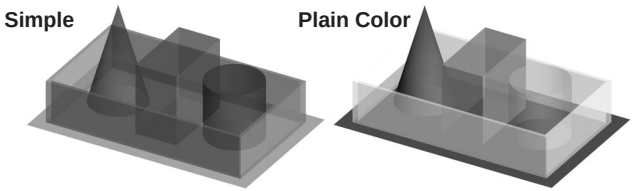


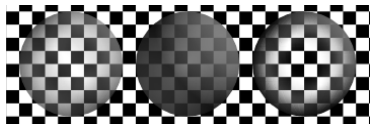
Transparency shaders

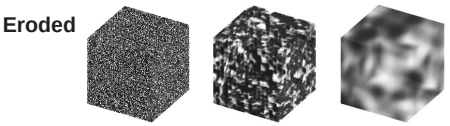


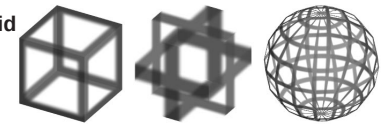
Transparency shaders are selected from the **Transparency** pop up menu, which contains five categories. The first group contains the default **None**, which produces no transparency. The second group contains five shaders, which, contrary to the transparencies in groups three and four, are set independently from the color shader used for the same surface style. **Simple** and **Plain Color** produce uniform transparencies. **Edge** and **Neon** produce graduated transparencies based on the angle between the surface normal and the view direction. **Eroded** uses a solid shader and produces a pattern of irregular transparent and opaque areas that resemble rusted surfaces.

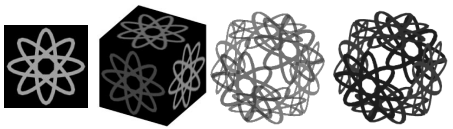
The third and fourth groups contain all the wrapped and solid color shaders, respectively. They can be used to add transparency to patterns that are also generated by the color shaders. However, this is not a necessary condition, and these shaders can also be used independently. The fifth group contains the single **Transparency Map** shader, which produces transparent patterns from precaptured images.

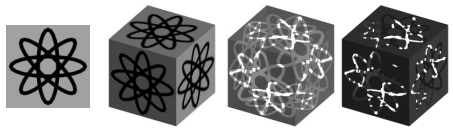












Transparency Maps

Bump shaders

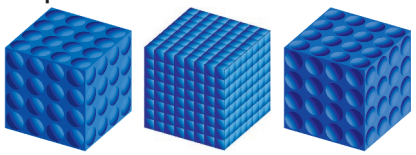
A ***bump*** is a three dimensional effect which is added to a surface through rendering. Bump shaders are selected from the **Bump** pop up menu, which contains six groups.

The first group contains the default **None**, which produces no bumps. The second group contains four wrapped bump shaders, and the third group two solid shaders, which do not have a color shader counterpart.

The fourth and fifth groups contain all the wrapped and solid color shaders, which can be used to add bumps to patterns that are also generated by the color shaders. However, this is not a necessary condition, and these shaders can also be used independently. The sixth group contains the single **Bump Map** shader, which produces bumps from precaptured images.

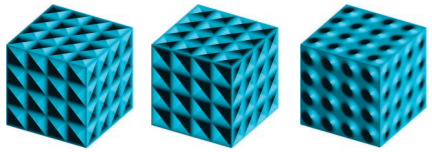
Most of the bump shaders use two parameters that control the character of the bumps. **Amplitude** controls the three dimensionality of the bumps. Positive values produce raised bumps, while negative values produce indented bumps. **Blend** controls the transition from a bump to the surface it is on. Small values produce sharper transitions, and higher values produce smoother transitions.

Dimple



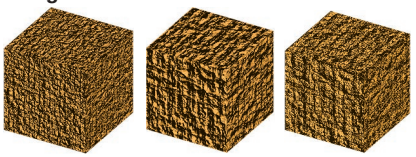
Scale	100	100	100
Separation	100	50	100
Radius	50	50	50
Center Depth	25	25	-25
Blend	10	10	10

Knurl



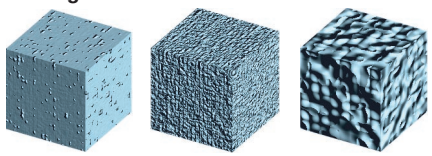
Scale	100	100	100
Blend	10	10	100
Amplitude	100	-100	100

Rough Solid



Scale	500	1000	1000
Amplitude	10	-100	100
Detail	30	30	100
Sharpness	10	10	10

Casting



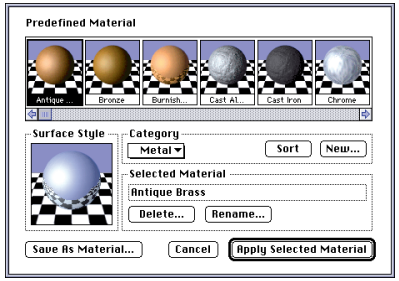
Scale	400	400	1000
Casting Amplitude	5	100	-100
Dented Amplitude	50	50	100
Dented Scale	100	100	100
Dented Threshold	80	80	10
Detail	20	20	10

Copying and invoking predefined surface styles

Two button commands available at the lower right end of the **Surface Style Parameters** dialog allow you to copy the parameters of one surface style into another, and to pick the parameters from a pre-stored surface style.

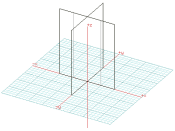
When **Copy From...** is selected, a dialog is invoked which displays the palette with the list of surface styles. Any of the surface styles displayed can be selected by clicking on it. Clicking on **OK** to close the dialog copies the parameters of the highlighted surface style into the surface style active in the Surface Styles palette.

Clicking on **Predefined...** invokes the **Predefined Material** dialog. A ***predefined material*** is a surface style that has been defined earlier, has been assigned a name, has been grouped into a category of materials, and has been saved into a file from which it can be retrieved. These predefined surface styles represent commonly used materials.



Included with **form•Z RenderZone** is a folder called **form•Z Materials**, which contains four files called Metal, Organic, Stone, and Synthetic. Each file contains a number of predefined surface styles, which become available to the user when invoking the **Predefined Material** dialog. The user can also create new categories and save new definitions of materials.

Rendering trees



You can complement your rendering with images of trees, which are produced with texture mapping techniques, as follows:

- Generate two single segment extrusion objects, positioned perpendicular to each other, as shown.

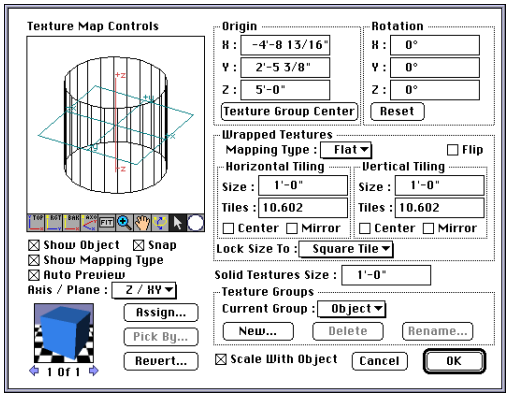
- Define a surface style where the same tree image map is assigned both as a **Color** shader and as a **Transparency** shader. Assuming that your tree image has been captured with a black background, turn **Make Only Black Areas Transparent** on in the **Transparency Map Options** dialog. You may also select **Constant** for **Reflection** to help the image preserve its original colors.

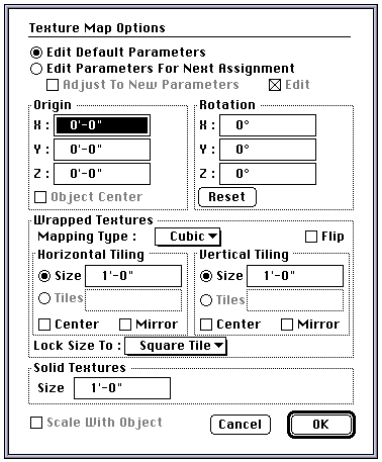
- Use the Texture Map tool to set the proper mapping parameters (see next section). Use **Flat** mapping, rotate the texture axes so that the XY plane sits on the surface object (with X horizontal and Y vertical), position the texture's origin on the lower left point of your object, and enter 1 in both **Tiles** fields. Repeat the texture mapping process for both surface objects, and make certain that the tree image is mapped on both in consistent directions.

The tree used in the example to the right is a birch tree whose image was derived photographically. **form•Z RenderZone** is shipped with eight precaptured tree images (see samples on back cover) and additional tree images can be produced by the user.

Texture Map tool

All **form•Z** objects are assigned the active surface style and a global default mapping at the time they are generated. The Color tool can then be used to assign a different surface style, and the Texture Map tool to assign a different mapping type and to manipulate the positioning of a surface style.

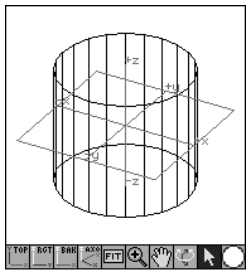




When you click on the object whose texture you wish to map, the **Texture Map Controls** dialog is invoked where the mapping parameters are set. Most of these parameters can also be preset in the **Texture Map Options** dialog, before executing the mapping operation. This dialog (invoked by double clicking on the Texture Map tool) carries two sets of parameters, only one of which is displayed at a given time. Which set is displayed depends on the option selected. **Edit Default Parameters** displays the global default texture mapping parameters. These are used by objects that were not specifically assigned individual texture mapping controls. **Edit Parameters For Next Assignment** displays parameters that will be assigned to an object the next time the Texture Map tool is applied.

Texture mapping involves the following processes, each of which corresponds to a portion of the dialog that contains the respective controls:

- The texture coordinate system is moved and/or rotated to position and orient it relative to an object or its faces.
- One of the available mapping methods is selected.
- The size of the wrapped texture tile or solid texture module is defined.
- Groups of faces may be defined as texture groups and the mapping process repeated for each of these groups.



Previewing and positioning textures

Most textures, particularly wrapped textures, have directions. Detailed mapping of textures is frequently required in order to position the textures properly. As the position of the texture is manipulated, it is viewed in the preview window in the upper left portion of the **Texture Map Controls** dialog. This window works as the other preview windows in **form•Z**.

The images in the preview window are color coded:

- The bounding shapes and two of the axes of the texture coordinate system are shown in the **form•Z** grid color, which is light green by default.
- The third axis is shown in the world axes color, which is red by default. This is the axis about which rotations are executed and is selected from the **Axis/Plane** menu.
- The faces of the active texture group are shown in black. Faces of other texture groups are drawn in the project's ghost color, which is gray by default.

The three check boxes under the preview window determine what is displayed and how the display is refreshed: When **Show Object** is on (default) the object is drawn. When **Show Mapping Type** is on (default), the texture coordinate system and the bounding shape that represents the mapping type are drawn. When **Auto Preview** is selected (default), the object is always redrawn in the current rendering mode after a parameter is changed.

When **Snap** is on and you move the mouse after you have initiated a ***move origin*** or a ***rotate axes*** operation (see below), the mouse snaps to points, midpoints, and segments of the object. You can temporarily cancel the snapping by pressing the ***shift*** key. Likewise, if **Snap** is off, you can temporarily activate it by pressing ***shift***.

The view in the preview window can be manipulated using tools from the preview tool palette located under the preview window. All these tools, except the following two, are the same as those in other preview windows.

Is used to move or rotate a texture and to select faces for the current group.

Pressing the mouse on this icon pops out a menu for selecting a display type.

- **Wireframe** (default) produces a wire frame image of the object.
- **Tiled** shows how the texture tiles are mapped on the current texture group.
- **Rendered** produces a z-buffer rendering of the object.

Textures can be positioned onto objects using either numeric or graphic input. For both methods, two basic operations are available: ***moving*** the origin of the texture coordinate system, and ***rotating*** the texture coordinate system about one of its axes.

Numeric input: The **Origin X**, **Y**, and **Z** fields represent the position in the world coordinate system to which the origin of the texture coordinate system will be moved. Clicking on **Object Center** moves the origin to the center of the object. The **Rotation X**, **Y**, and **Z** fields are the angles by which the texture will be rotated relative to each axis. **Reset** cancels all the rotations (sets **X**, **Y**, and **Z** fields to 0°).

Graphic input: The texture can also be positioned graphically, by moving and rotating its coordinate system directly in the preview window. When the Arrow tool ( ) is selected and the cursor is moved inside the preview window, its icon changes according to which part of the texture coordinate system or the object it is closest to.

Changes to this icon when it is on or close to the texture's ***origin***.

Changes to this icon when it is on or close to an ***axis*** that can be rotated.

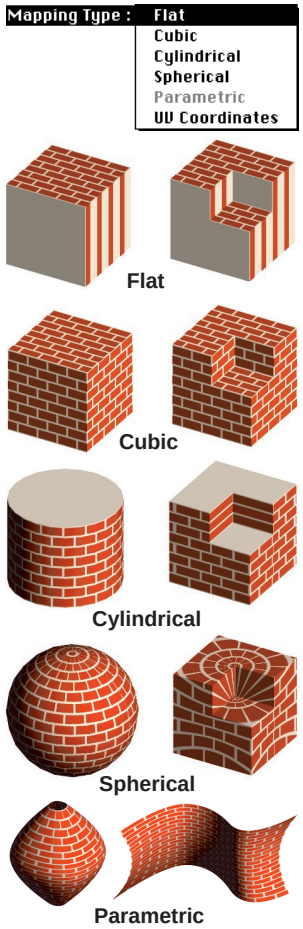
Changes to this icon when you press the button to execute either a move origin or a rotate axis operation.

Changes to this icon when it snaps. As the mouse is moved, if **Snap** is on, it snaps to points, segments, or segment mid-points of the object. If **Snap** is off, it moves freely, relative to the plane selected in the **Axis/Plane** pop up menu. Both states of the **Snap** can be temporarily reversed by pressing the ***shift*** key.

Texture mapping methods

Wrapped textures can be mapped onto objects using one of six available methods. The fifth method can be used only with c-meshes. These types are selected from the **Mapping Type** pop up menu.

Note that the texture mapping methods apply only to wrapped textures, and are ignored by the solid textures. The sixth method (UV) is not derived independently but is derived from the other mapping methods, which need to be applied first.



Flat initially places a texture on the XY plane of the Cartesian coordinate system. This position can be changed with the positioning tools.

Cubic maps textures by using a cube which bounds the object. The texture is generated on each of the faces of the cube, and is projected in a direction perpendicular to the face of the cube on which it was generated. Each face of the object receives its texture from that face of the bounding cube to which it is most parallel.

Cylindrical maps the textures using a cylinder which bounds the object. The texture is generated on the round surface of the cylinder, and is projected towards its central axis.

Spherical maps textures using a sphere which bounds the object. The sphere's origin is at its center, and its pole is on the Z axis of the texture coordinate system. The texture is generated on the surface of the sphere starting with its largest diameter, which corresponds to the horizontal direction. As the lines in the Y direction converge to the poles of the sphere, the texture is scaled as rows are laid out. The spherical texture is projected towards the center of the sphere.

Parametric mapping can be applied only to c-mesh objects, which can be surface or solid objects. The length of the c-mesh corresponds to the X direction of the texture tile, and the depth corresponds to the Y direction. The number of times the tile is repeated along the length and depth of a c-mesh is determined by the values entered in the **Horizontal** and **Vertical Tiles** fields of the **Texture Map Controls** dialog, respectively.

The parametric texture mapping procedures are based on NURBS, and subsequently, only those **form•Z** c-meshes that can be expressed as NURBS can accept parametric texture mapping. These are the NURBS, B-Splines, Bezier, continuous Bezier, quick cubic and quick quadratic curves. Broken Bezier and tangent curves cannot be accurately simulated. In addition, all control lines of the c-mesh should have the same smoothing type assigned to them, including the same degree. Parametric texture mapping will not work with c-meshes with rounded ends.

UV Coordinates freezes and permanently stores the textures on an object at the vertices of that object. This can be used after any one of the five other types of texture maps have been assigned to an object. While UV coordinate mapping offers extensive possibilities for transporting objects with their texture parameters to other rendering or animation applications, it is less flexible than the other mapping types when objects are geometrically transformed.

Defining the size of textures

When a texture is generated by its default parameters, the size of the texture tile is determined by the values entered in the **Wrapped Textures Size** or the **Solid Textures Size** fields of the **Surface Style Parameters** dialog. When the Texture Map tool is used, the size of the tiles is determined by the values entered in the **Size** fields of the **Texture Map Controls** dialog. Note again that distinct **Size** parameters are available for wrapped and solid textures.

Transforming textures with objects

When one of the geometric transformations (move, rotate, scale or mirror) is applied to an object to which one or more textures have been applied, or when such an object is copied, the transformed object carries the texture and the mapping of the original object. That is, the textures and their positions are transformed with the object.

However, when objects with textures are scaled, the textures are resized only when **Scale With Object** is selected at the time a texture is mapped onto an object. If this option is not selected, the texture tile size is not affected by the Scale operation, and remains the same, while the number of tiles placed changes. When a solid texture is applied to an object which is then resized using the Uniform Scale tool, the same two options are available. The Independent Scale does not affect solid textures that may have been assigned to an object.

Texture groups

Textures may be assigned and mapped onto complete objects or groups of faces called ***texture groups***. Texture groups are defined using the **Texture Groups** pop up menu and buttons at the lower right portion of the **Texture Map Controls** dialog. When the Texture Map tool is first applied to an object, the texture is, by default, assigned to the complete object. This is indicated by the word **Object** that appears in the **Current Group** pop up menu, which contains no other entry. This pop up menu expands as new texture groups are defined.

Clicking on **New...** defines a new texture group and adds it to the **Current Group** menu. It invokes the **New Texture Group Name** dialog where a name can be entered. As soon as a new texture group is named, the image of the object displayed in the preview window is dimmed to indicate that no face is currently contained by the group. Next, you need to select the faces that will be in this group. One or more faces can be selected in the usual manner. The faces belonging to a group are shown in the preview window in black when the **Wireframe** display option is used.

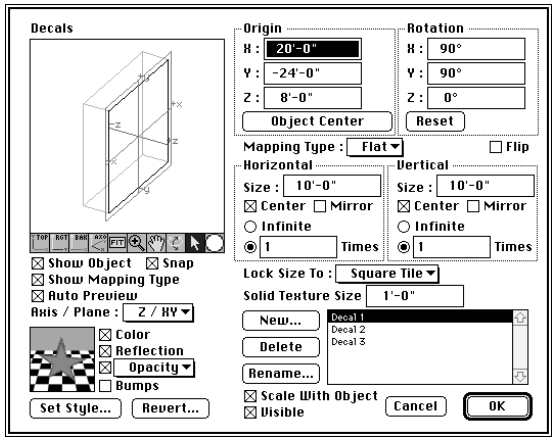
When a face is picked to make it a member of the active texture group, the texture coordinate system can also be placed on it. This is done by pressing the ***option*** key on the Macintosh or the ***ctrl+shft*** keys on Windows when selecting the face. The origin of the texture coordinate system is always placed at the center of the face, the X and Y axes are placed on the surface of the face, and the Z axis is always perpendicular to the face. The orientation of the X and Y axes depends on how the face is picked. This method of automatically mapping the texture coordinate system onto a face when it is selected offers convenience, particularly when a texture needs to be mapped onto a surface which is not parallel to one of the Cartesian planes.

Decal tool

Additional surface styles can be assigned to objects as ***decals.*** That is, after a basic surface style has been assigned and mapped to an object using the Color and Texture Map tools, the Decal tool can be used to assign additional patterns. This feature is useful for rendering stickers or labels, and for arranging a variety of different textures on a single surface. The Decal tool allows you to assign up to 32 surface styles, which may even overlap.

When using the postpick method, with the Decal tool active, click on the object to which you wish to attach a decal. This invokes the **Decals** dialog, which allows you to place, size, and orient decals on an object. To use the prepick method, with the Pick tool active, and topological level set to Object, preselect any number of objects. Then, with the Decal tool active, click anywhere in the graphics window.

The **Decals** dialog is very similar to the **Texture Map Controls** dialog. It contains parameters for defining the decal origin, rotation, size, and mapping type, a preview window, and commands for manipulating the display of the preview and for mapping a decal on the object. It also contains a group of options which are unique to this dialog. These are the options that allow you to create and name new decals, and to associate a particular surface style with them.



The three groups of options at the upper right portion of the **Decals** dialog, **Origin, Rotation,** and **Mapping Type**, are identical to those in the **Texture Map Controls** dialog. The options for sizing the decals, in the middle right portion of the dialog, are very similar to those in the **Texture Map Controls** dialog, but contain additional parameters for determining how many times the decals will be repeated.

When **Center** is on, the decal is centered about the origin. When **Mirror** is selected, each alternating repetition of a decal is mirrored. When **Infinite** is selected, the decal is repeated as many times as necessary to cover the surface, in the horizontal and vertical directions, respectively. When **Times** is selected, the decal is repeated as many times as indicated in the text field next to the radio button.

Creating decals

Decals are created through button commands at the lower portion of the **Decals** dialog. They are listed in a palette-like window, referred to as the ***decal list.*** After new decals are created and a name is assigned to them, they are added to the list. When the decals list is empty, all the fields in the **Decals** dialog appear inactive. This is the case, for example, when the Decal tool is applied to an object for the first time.

One of the decals in the list is the ***active decal*** and its parameters are displayed in the dialog. Another decal can be made active by clicking on its name in the list. There can only be one active decal at any given time. The surface style with which the active decal is associated is displayed in an icon at the lower left end of the **Decals** dialog. This image changes each time the active decal changes.

An object can have up to 32 decals, which may or may not overlap. When they do, the order in which they are displayed at rendering time is determined by the order they appear in the decal list. The decal at the end of the list is applied first, and the decal at the top of the list is applied last and covers all the decals below it. The order of the decals is initially determined by the order in which they are created. It can be changed by clicking on the name of a decal and dragging it to a new position.

The **New...** button allows you to create a new decal and add it to the decal list. It invokes the **New Decal Name** dialog, where a default name for the new decal appears and a different name can be entered in the usual manner. The **Delete** button deletes the active decal. The **Rename...** button allows you to assign a new name to the active decal by invoking the **New Decal Name** dialog. The **Revert...** button cancels all the changes made since the dialog was opened.

When new decals are created, they are associated with the currently active surface style by default. The **Set Decal Style...** button can be used to assign another surface style through a dialog that displays the surface style palette.

The **Color**, **Reflection**, **Transparency**, and **Bumps** options can be used to vary the effects of the decal shaders when they are rendered. That is, when they are on, the settings of the respective component of a shader are applied. By default, **Color** and **Reflection** are on, and **Transparency** and **Bumps** are off.

When **Visible** is selected, the active decal is made visible and is displayed when a rendering is produced. When off, the respective decal is skipped.

Determining the faces to be affected by a decal

When a new decal is created, it initially affects all the faces of the object to which the Decal tool was applied. However, any number of faces can be exempted from it. You turn a face off relative to the active decal by selecting it in the preview area.

Faces exempted from a decal are drawn in the project's ghost color when the decal is active. Picking an exempted face again turns it back on and the face is drawn in black. Faces can be picked as in the **Texture Map Controls** dialog. All faces of an object can be turned on relative to the active decal, regardless of their current status, by pressing the ***command+a*** keys on the Macintosh or the ***ctrl+a*** keys on Windows.

Example 1: Creating a bottle with labels

This example outlines how the shown bottle was rendered by applying three decals.

The bottle itself was rendered using a **Glass** reflection shader, to generate accurate transparencies and reflections.

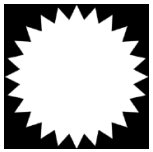
(a) Chardonnay label: This decal was applied using one tile (**Times** = 1 for both **Horizontal** and **Vertical**), centered in both directions, and cylindrical mapping. The decal surface style was set to use the **Color**, **Reflection**, and **Transparency**, which caused the decal to override the color, reflection, and transmission of the bottle glass.

(b) Circular label: The surface style defined for this decal, for its **Color Texture Map** used the color channel of the PICT image. For its **Transparency Map**, it used the alpha channel of the same PICT image. In the **Decals** dialog, the same settings as the previous label were used. In the decals list, this label was listed above the Chardonnay, which caused it to be rendered on top.

(c) Neck label: For this label, all the parameters in the **Decals** dialog were set as for the other labels, except that **Infinite** was selected for the **Horizontal** direction. This caused it to be repeated as many times as necessary to cover the perimeter of the neck of the bottle.

a

b



c

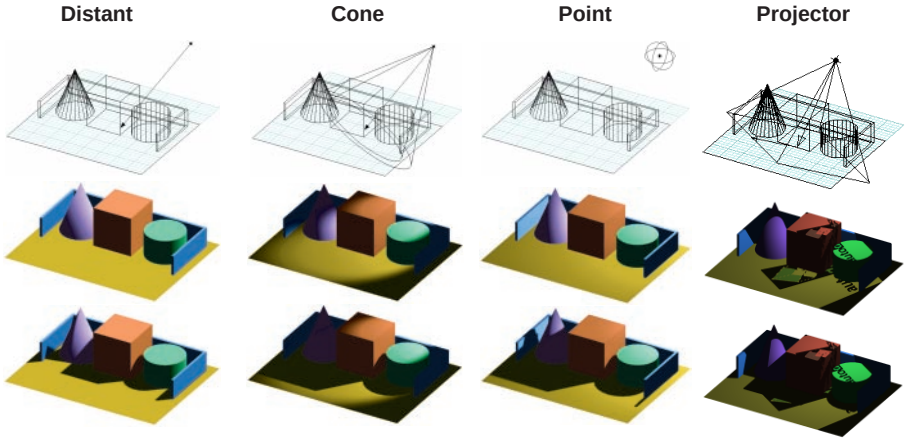
Example 2: Surfaces with partial reflections

Imagine a marble which has silver streaks. While most of the marble has a matte appearance, the silver streaks are reflective. Note that when assigning a surface style to an object with the Color tool, a surface such as marble can be made either reflective or matte, but not both at the same time. However, this effect can be achieved with the Decal tool.

- Two surface styles (Stone and Mask) are defined using **Marble, Textured**.
- Stone: **Marble, Textured** is used as **Color** shader, and **Matte** for **Reflection**.
- Mask: **Plain Color** is used for **Color** shader, **Mirror** for **Reflection**, and **Marble, Textured** for **Transparency**. The parameters for the **Transparency** and **Color** shaders are set to the same values.
- Surface style Stone is assigned to the sphere using the Color tool.
- Surface style Mask is assigned to the sphere as a decal with only the **Reflection** component of the surface style turned on.

Lights

In **form•Z RenderZone** you can define multiple light sources which can be one of four available types: ***distant***, ***point, cone,*** and ***projector*** lights. One of the distant lights is designated as the ***sun***, and is the light used by the lower levels of rendering which work with only one light (**Quick Paint*** and **Surface Render***). Note that these lights are in addition to the ***ambient*** light that illuminates all the surfaces equally. A light's color and intensity is set in the **Lights** dialog. The four types of lights are displayed as shown below. They appear when the wire frame display method is used and they are marked visible in the Lights palette. The type of light they emit and the type of shadows they cast are also illustrated below.



Distant (or direct) light emits parallel rays from an infinitely distant light source, such as the sun. The intensity of this light remains constant throughout a scene.

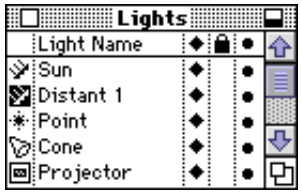
Point lights emit rays from a given point outward in all directions. The intensity of these lights may or may not decrease as the distance from the source increases. A candle would be an example of a point light.

Cone lights emit rays from a given point in the direction defined by a conic shape that is associated with them. The intensity of these lights may or may not decrease as the distance from the source increases. The intensity of these lights also decreases at the perimeter of the cone. This area of decreased intensity is defined by a second cone inside the outer cone. The headlight of a car is an example of a cone light.

Projector lights emit rays from a given point in the direction defined by a pyramid that is associated with them. The intensity of these lights may or may not decrease as the distance from the source increases. The rays emitted by such lights are also filtered through an image map, projecting that image onto a scene. An example of a projector light would be a slide projector.

At start-up, **form•Z RenderZone** automatically creates one distant light, representing the sun. In addition, a scene always contains ambient light. The distant, point, and cone lights and their names are displayed in the Lights palette. New lights can be defined and existing lights can be deleted or edited through the **Lights** dialog. The parameters and the positions of lights can be set in the **Light Parameters** dialog. The positions of lights can also be manipulated using the geometric transformation tools of **form•Z**. Lights may or may not cast shadows at different levels of quality.

The Lights palette and dialog

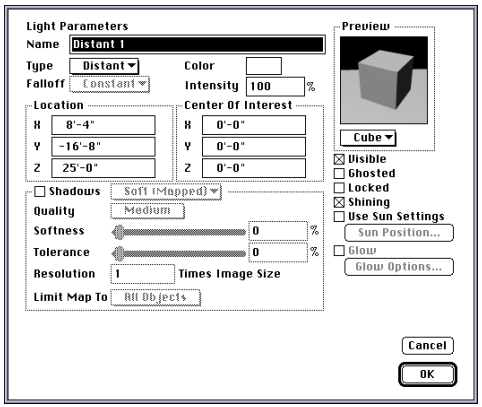


All currently defined lights are displayed in the **Lights** palette as in the regular version of **form•Z**, except that **form•Z RenderZone** has additional types of lights:

= distant light, = sun,

= cone light, = projector light, and = point light.

New lights can be created and defined, and existing lights can be deleted, copied, and edited through the **Lights** dialog, as in the regular version of **form•Z**.

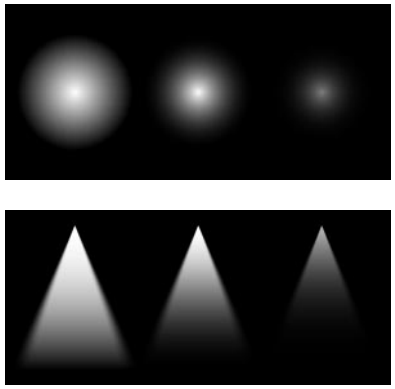


The **Light Parameters** dialog is invoked when you execute the **New...** or **Edit...** commands in the **Lights** dialog, and when you double click on the name of a light in the Lights palette. This dialog is very similar to its counterpart in the regular version of **form•Z,** however, it has a few additional features.

If the current light is a ***point, cone, or projector*** light, the **Falloff** menu allows you to select how the intensity of the light decreases as the distance of the light source increases.

There are three options: **Constant** results in a constant intensity, regardless of the distance of a surface from the light. **Linear** decreases the intensity of a light linearly with the distance. **Square** decreases the intensity exponentially. The value in the **Intensity** field is expressed as a percentage.

The **Glow** option is available only for cone and point lights. When checked, the light displays a glowing volume when rendered in the Preview Z-Buffer and above rendering modes. For point lights, this volume is a sphere. Cone lights show a conical volume. The volume of the point light has a maximum brightness at the center of the sphere, and diminishes toward its perimeter. The volume of the cone light has maximum brightness at the origin of the light. The brightness diminishes along the direction of the cone light, and disappears at the center of interest of the light. The brightness also diminishes in the area between the inner and outer cones.



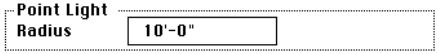
Intensity:	100%	100%	50%
Falloff:	**Linear**	**Square**	**Cubic**

The parameters for the light glow are set in the **Glow Options** dialog, which is invoked by clicking on the **Glow Options...** button. There are two variations of this dialog: one for the point and one for the cone light.

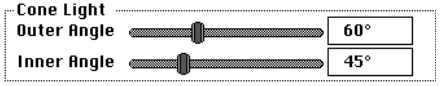
Positioning lights

The physical position of a ***light source*** is defined by its coordinates in 3D space. The distant, cone, and projector lights also have a direction, which is defined by the position of a ***center of interest*** taken together with the position of the light source. The positions of lights can be set through both numeric and graphic input. These operations work as in the regular version of the program, except that for point and cone lights, some additional parameters must be set.

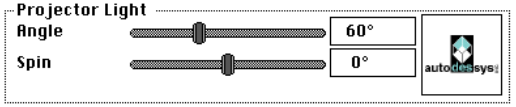
Numeric input: The location and direction of a light source are set through values entered in two groups of numeric fields: The **Location X**, **Y**, and **Z** fields determine the position of the light source in 3D space. The **Center Of Interest X**, **Y**, and **Z** fields determine the position of the center of interest of the light source. These parameters apply to all lights except point. Additional type specific parameters are set at the lower portion of the **Light Parameters** dialog.



The **Point Light Radius** parameter, that applies to point lights only, determines a spheric surface, centered at the origin of the light, on which the light's intensity takes the value specified in the **Intensity** field. This is also the spherical surface shown when point lights are displayed in wire frame.



The volume of a **Cone Light** is defined by two parameters: the **Outer Angle** and **Inner Angle** determine the angle for the outer and inner cones of a cone light, and range from 0° to 180°. The two are interdependent. The inner angle cannot be greater than the outer angle, and it cannot be less than half of the outer angle.



The **Projector Light Angle** defines the angle of the projector pyramid. It may vary from 0° to 180°.

The **Projector Light Spin** determines a rotation about the axis of the light, defined by its origin and center of interest. The spin also affects the position of the texture map that is always associated with a projector light. With a spin angle of 0 degrees the vertical direction of the texture points upward towards the world Z axis. In wireframe display, the up direction of the projector light is indicated by a small arrow from the center of interest along the vertical direction of the texture map.

To the right of the **Projector Light** box, a texture map is displayed. Clicking on it invokes the **Projector Map Options** dialog. This is a standard texture map dialog that can be used to load a new texture map. Projector maps can be of type PICT on the Macintosh, BMP or Metafile on Windows, and Targa or TIFF on both operating systems.

Graphic input: Any of the four types of lights can be repositioned using one of the geometric transformation tools, namely Move, Rotate, Independent and Uniform Scale, and Mirror, available in the modeling tool palette.

You can move the light source, the center of interest (when it exists), or the complete light. A point light can be transformed in only one way, which is by clicking the mouse on its source point. When a distant, cone, or projector light is transformed, its complete graphic representation is transformed.

Casting shadows

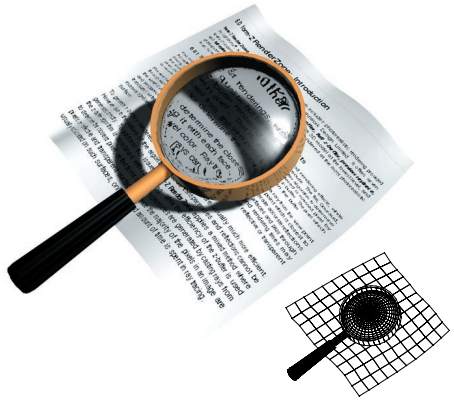
For objects to cast shadows, their **Object Casts Shadows** attribute should be on. Note that this is set on by default, when new objects are created. It can be turned on/off with the Set Rendering Attributes tool (12th row). Four of the rendering modes available in **form•Z RenderZone** can cast shadows by turning on the appropriate options, as follows:

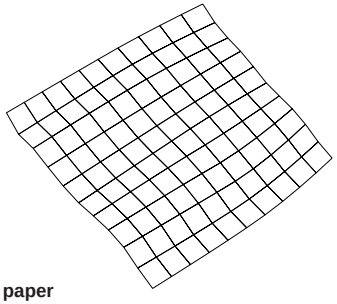
• **Quick Paint***, **Surface Render***: Select the **Render With Shadows** options in the **Quick Paint Options** and **Surface Rendering Options** dialogs, respectively. Recall that these rendering modes work with a single light, which is the ***sun*** light. This is also the only light that casts shadows in these modes.

• **Shaded Render***: Turn on the shining attribute and the **Shadows** options of at least one light, and turn on the **Render With Shadows** option in the **Shaded Rendering Options** dialog. **Shaded Render*** works with one or more lights, all of which can be of only one type, distant. When more than one light is used, any number of them can cast shadows.

• **RenderZone***: As for **Shaded Render***, the shining and **Shadows** attributes of at least one light needs to be turned on. When it is, the option to select **Hard** (raytraced) or **Soft** (bit mapped) shadows is also available. The **Shadows** option in the **RenderZone Options** dialog is also turned on. When it is, the option to select **Opaque** or **Transparent** shadows is also available. This option only affects the shadows of objects to which transparent shaders have been applied and hard (raytraced) shadows are cast. Examples of transparent shadows are shown below.

Soft shadows are generated by producing a shadow map for each light that casts shadows, which can be very memory intensive, if too many lights cast shadows. How many is too many depends on how much memory you have available on your machine. One way to economize is to turn off the shadow casting attribute of large objects (such as ground) that do not really cast shadows. Hard raytraced shadows, while they typically require more processing time, they work with less memory.

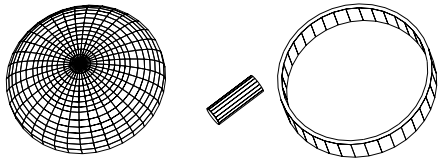




Example 3: Magnifying glass

Magnifying lens

Color: Plain
R: 100%, G: 100%, B: 100%
Reflection: Glass, Accurate
Ambient Reflection: 0 %
Diffuse Reflection: 0 %
Specular Reflection: 90 %
Transmission: 90 %
Reflectivity: 19 %
Roughness: 2
Refraction: 1.591
Transparency: None
Bump: None



Magnifying lens frame

Color: Plain
R: 95%, G: 70%, B: 39%
Reflection: Metal, Accurate
Ambient Reflection: 20 %
Diffuse Reflection: 70 %
Specular Reflection: 30 %
Reflectivity: 35 %
Roughness: 10
Refraction: R: 0.186, G: 0.59, B: 1.518
Absorption: R: 3.756, G: 2.297, B: 1.897
Transparency: None
Bump: None

Lens handle

Color: Plain
R: 0%, G: 0%, B: 0%
Reflection: Plastic
Ambient Reflection: 100 %
Diffuse Reflection: 75 %
Specular Reflection: 50 %
Roughness: 4
Specular Color: R: 100%, G: 100%, B:100%
Transparency: None
Bump: None

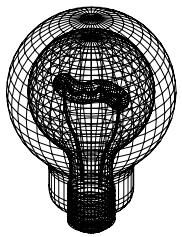
Sheet of paper

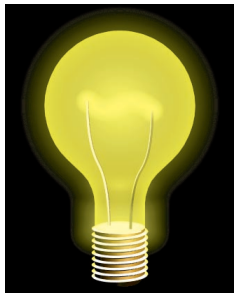
Color: Image Map of scanned text
Horizontal Reps: Infinite
Vertical Reps: Infinite
Scale: 100%
Point Sampling: On
Cropping: None
Reflection: Matte
Ambient Reflection: 100 %
Diffuse Reflection: 100 %
Transparency: None
Bump: None

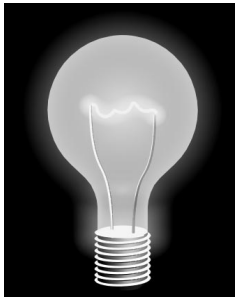
RenderZone rendering options

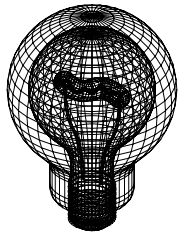
Rendering Mode: Full Raytrace
Shadows: On
Bump Mapping: Off
Transparencies: On
Reflections: On
Environment: None
Depth Effects: Off
Smooth Shading: On
Objects Without Smoothing Attributes Only
Edges With Angle Greater Than: 140°
Render C-Meshes Always Smooth
Background: White
Decompose: Off

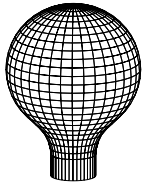
Example 4: Bulb











Example 4: Bulb

(continued)

Light bulb glass

Color: Plain
R: 100%, G: 100%, B: 0%
Reflection: Constant
Transparency: Center-Edge
Center Transparency: 75 %
Edge Transparency: 47 %
Bump: None

Metal socket

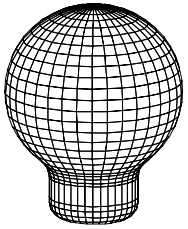
Color: Plain
R: 81%, G: 58%, B: 2%
Reflection: Accurate Metal
Ambient Reflection: 10 %
Diffuse Reflection: 80 %
Specular Reflection: 80 %
Reflectivity: 31 %
Roughness: 10
Refraction: R: 0.5, G: 0.5, B: 0.5
Absorption: R: 8.39, G:8.39, B: 8.39
Transparency: None
Bump: None

Bulb filament

Color: Plain
R: 100%, G: 100%, B: 0%
Reflection: Constant
Transparency: None
Bump: None

Filament glow

Color: Plain
R: 100%, G: 98%, B: 38%
Reflection: Constant
Transparency: Neon
Intensity: 69 %
Falloff: Linear
Bump: None



Bulb Glow

Color: Plain
R: 100%, G: 98%, B: 38%
Reflection: Constant
Transparency: Neon
Intensity: 60 %
Falloff: Linear
Bump: None

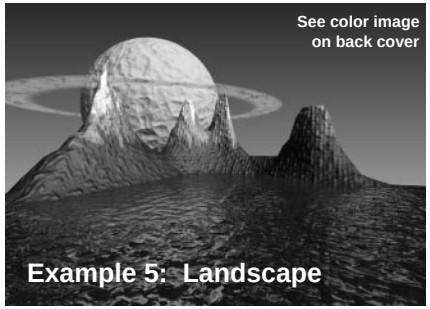
Filament holders

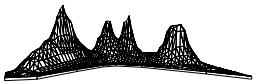
Color: Plain
R: 48%, G: 49%, B: 48%
Reflection: Metal, Simple
Ambient Reflection: 100 %
Diffuse Reflection: 100 %
Roughness: 25
Transparency: None
Bump: None

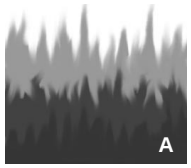
RenderZone rendering options

Rendering Mode: Full Z-Buffer

Shadows: Off
Bump Mapping: Off
Transparencies: On
Reflections: On
Environment: None
Depth Effects: Off
Smooth Shading: On
Objects Without Smoothing Attributes Only
Edges With Angle Greater Than: 140°
Render C-Meshes Always Smooth
Background: Project Color(black)
Decompose: Off







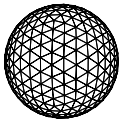
Mountains

Color: Image Map A
Reflection: Plastic
Ambient Reflection: 100 %
Diffuse Reflection: 75 %
Specular Reflection: 30 %
Roughness: 31
Specular Color: R:100%, G:100%, B:100%
Transparency: None
Bump: Casting
Scale: 100 %
Casting Amplitude: 10 %
Dented Amplitude: 30 %
Dented Scale: 100 %
Dented Threshold: 40 %
Detail: 30 %
Texture Map Controls:
Texture Group: Object
Origin: Object Center
Rotation: None
Mapping Type: Cylindrical
Horizontal Size: 60°
Horizontal Center: On
Horizontal Mirror: Off
Vertical Size: 34'-0"
Vertical Center: On
Vertical Mirror: Off

Water

Color: Plain
R: 30%, G: 40%, B: 69%
Reflection: Mirror
Ambient Reflection: 10 %
Diffuse Reflection: 28 %
Specular Reflection: 90 %
Reflection: 56 %
Roughness: 6.25
Transparency: None
Bump: Rough, Solid
Scale: 100 %
Amplitude: 4 %
Detail: 30 %
Sharpness: 0 %
Solid Texture Size: 16'-0"

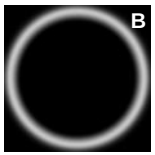
Planet



Color: Plain
R: 100%, G: 72%, B: 8%
Reflection: Matte
Ambient Reflection: 250 %
Diffuse Reflection: 100 %
Transparency: None
Bump: Casting
Scale: 100 %
Casting Amplitude: -33 %
Dented Amplitude: 0 %
Dented Scale: 0 %
Dented Threshold: 0 %
Detail: 20 %
Solid Texture Size: 28'-0"

Planet's ring

Color: Marble, Simple
Scale: 100 %
Detail: 30 %
Color: R: 100%, G: 35%, B: 12%
Reflection: Constant
Transparency: Transparency Map B
Bump: None
Texture Map Controls:
Texture Group: Object
Origin: Object Center
Rotation: None
Mapping Type: Flat
Horizontal Size: 141'
Horizontal Center: On
Horizontal Mirror: Off
Vertical Size: 141'
Vertical Center: On
Vertical Mirror: Off
Solid Textures Size: 14'-0"



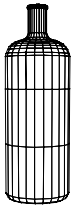
RenderZone rendering options
Rendering Mode: Full Raytrace
Shadows: Off
Bump Mapping: On
Transparencies: On
Reflections: On
Environment: Cubic Maps
Depth Effects: Off
Smooth Shading: On
Objects Without Smoothing Attributes Only
Edges With Angle Greater Than: 140°
Render C-Meshes Always Smooth
Background: Horizon
Decompose: Off
Environment Maps:
Image C on each side of environment cube
Horizon:
Top Sky Color: R: 26%, G: 6%, B: 39%
Bottom Sky C.: R: 100%, G: 47%, B: 7%
Top Earth Color: R: 2%, G: 20%, B: 13%
Bottom Earth C.: R: 44%, G: 55%, B: 44%
Horizon: 65 %

C

Example 6: Bottles

Green (tall) bottle

Color: Plain
R: 62%, G: 89%, B: 70%
Reflection: Glass, Accurate
Ambient Reflection: 0 %
Diffuse Reflection: 0 %
Specular Reflection: 90 %
Transmission: 88 %
Reflectivity: 80 %
Roughness: 2
Refraction: 1.591
Transparency: None
Bump: None

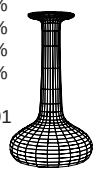


Ash tray

Color: Marble, Simple
Scale: 5000 %
Detail: 30 %
Color: R: 59%, G: 44%, B: 60%
Reflection: Glass, Accurate
Ambient Reflection: 28%
Diffuse Reflection: 11 %
Specular Reflection: 90 %
Transmission: 90 %
Reflectivity: 90 %
Roughness: 3
Refraction: 1.591
Transparency: None
Bump: None
Solid Texture Size: 2"

Red vase

Color: Plain
R: 96%, G: 37%, B:42%
Reflection: Glass, Accurate
Ambient Reflection: 0 %
Diffuse Reflection: 10 %
Specular Reflection: 90 %
Transmission: 90 %
Reflectivity: 57 %
Roughness: 2
Refraction: 1.591
Transparency: None
Bump: None



Wine glass

Color: Plain R: 90%, G: 100%, B: 99%
Reflection: Glass, Accurate
Ambient Reflection: 0 %
Diffuse Reflection: 0 %
Specular Reflection: 90 %
Transmission: 90 %
Reflectivity: 90 %
Roughness: 2
Refraction: 1.591
Transparency: None
Bump: None

Whiskey glass

Same as above

Whiskey bottle

Color: Plain R: 100%, G: 96%, B: 72%
Reflection: Glass, Accurate
Ambient Reflection: 0 %
Diffuse Reflection: 0 %
Specular Reflection: 90 %
Transmission: 100 %
Reflectivity: 50 %
Roughness: 6
Refraction: 1.591
Transparency: None
Bump: Rough, Solid
Scale: 8000 %
Amplitude: 8 %
Detail: 10 %
Sharpness: 0 %
Solid Texture Size: 2"

Table surface

Color: Marble, Simple
Scale: 20000 %
Detail: 50 %
Color: R: 47%, G: 55%, B: 60%
Reflection: Mirror
Ambient Reflection: 100 %
Diffuse Reflection: 100 %
Specular Reflection: 80 %
Reflectivity: 30 %
Roughness: 6.25
Transparency: None
Bump Shader: None
Solid Texture Size: 2"

RenderZone rendering options

Rendering Mode: Full Raytrace
Shadows: Off
Bump Mapping: On
Transparencies: On
Reflections: On
Environment: Cubic Maps
Depth Effects: Off
Smooth Shading: On
Objects Without Smoothing Attributes Only
Edges With Angle Greater Than: 140°
Render C-Meshes Always Smooth
Background: White
Decompose: Off
Environment Maps: White image map on each side of environment cube.