STRATA™ VISION3d

The All-In-One
3-D Solution

FIRST PRINTING
June 1997

ACKNOWLEDGMENTS:

Strata Incorporated acknowledges with gratitude the contributions of its development team who produced the Strata Vision3d product. Special thanks to our elite group of beta testers for their work.

- **Software written** by Gary Bringhurst, Jim Boyd, Brad Campbell, John Q. Adams and James Wadley
- **Manual written** by Ron Woodland, and Ken Perkins — with assistance from others on the development team.
  Ammended by Tammy Brinkerhoff
- **Interface design** by Kyle Maxwell and Ken Bringhurst
- **Package design** by Chad Steglich and Rob Kirby

Strata Incorporated
2 W. St. George Blvd. • Ancestor Square, Suite 2100
St. George, Utah 84770 • (801) 628-5218

Technical support (voice): (801) 628-9751
Facsimile transmissions: (801) 628-9756

STRATA

TABLE OF CONTENTS

Chapter 2: Managing Documents

CHAPTER 3: MODELING OBJECTS

CHAPTER 4: USING SHAPES

Chapter 5: Using Textures

Chapter 6: Managing Lighting

Chapter 7: Scripting Animations

CHAPTER 8: RENDERING IMAGES

APPENDICES

INTRODUCTION

Welcome to the fascinating possibilities of 3-D modeling…and thank you for choosing Strata Vision3d™. It will be a powerful tool in your exploration of this challenging new medium of design.

The primary purpose of this Users Manual is to explain how to use Vision3d. It contains the detailed information you need to use this software to your fullest ability. The manual's format facilitates its use as a reference volume, but you may find that at least one sequential, front-to-back reading will provide a foundation upon which to better understand the features you will want to investigate in depth.

It is important to understand at the outset that Vision3d is not like other programs. While it follows the published Apple Interface Guidelines religiously — that is, its interface is very intutive, its controls and tools behave similar to those you have used in other Macintosh software, and the software keeps you informed every step of the way — working in three dimensions (four, when scripting an animation) is not at all like your other favorite 2-D programs. This is not like any software you've used before. You WILL need to read this manual (and perhaps a few other books, as well).

Yes, Vision3d is a complex program, but it is easy to use. You will be rewarded with every new level of skill you acquire. You should be able to create simple models and render impressively-realistic-looking images within minutes of the first time you launch it the first time. But there is much more power below the surface. The point is that, with 3-D, there are many details to consider as you create objects and assemble scenes that usually do not require attention in the 2-D environment. There are new "drawing" techniques to learn and there are many new imaging options to explore. The additional work is definitely worth the effort, though, because the results can often be better and more realistic than anything you could do in conventional painting or drawing software, no matter how good of an artist you are.

The real payoff comes, however, once the 3-D scene is assembled. Because your work is fully three-dimensional inits construction, you can look at it from any direction. Multiple images from different viewing positions are easily acquired and the time required to render additional perspectives is much less than drawing multiple images in any 2-D software

Three-dimensional modeling places significant performance requirements on your machine's hardware. It will be asked to calculate and display the data of a 3-D model in a coherent and editable form. This computational load increases exponentially as you progress to more complex and ambitious projects. In extreme cases, you may see slower screen redraws and longer pauses while executing some commands. In general, Vision3d performs best with faster CPUs and more RAM.

A secondary purpose of this manual is to provide guidelines and tips for efficient modeling techniques. In any project, there are usually several ways to accomplish the same result. Understanding the options and using the most efficient methods available can often make the difference between successful completion of a project and exceeding the design capacities of the program, the operating system, and even the hardware.

Failure to prepare leads to frustration and lost productivity. But it can all be avoided if you READ THE MANUAL!

While Vision3d is capable of working with very large models, you must also be realistic in your expectations. Usually, any limits in Vision3d are imposed by the operating system and/or hardware. Some. like maximum image size, will not change until new system resources become available. Other limits, like the amount of RAM and the rendering speeds, vary depending on the hardware you are using. As you come to fully understand the compromises within which you must work (that is, as you evolve from novice to expert to “power user”), your approach to modeling projects will become more efficient and your successes more sure.

Chapter 1: General Operation

This chapter discusses the general operation of Vision3d. It is presented as an overview of Vision3d's tools and how they interact with the user. Items presented in this chapter include pre-project planning and project sequence, the interface, windows and views, and an explanation of commands that apply to the software's general operation.

Typical project sequence

Typically, each Vision3d project includes these five phases:

1. **Open a model**
2. **Draw one or more objects**
3. **Apply texture(s)**
4. **Position lighting**
5. **Render an image**

Certainly, there are more phases that can be added to this project sequence. For example, if the final goal of your project is producing an animation, then a motion scripting phase would be inserted between the Position lighting and Render an image phases. However, nearly all Vision3d projects, at minimum, will contain the above five phases in sequence.

Experimenting with tools and techniques is always fun, and can be a valuable method for learning how to use Vision3d. But remember, it is wise to do some planning before tackling a complex project. A little time spent planning ahead can save you the headaches often caused by plunging into a project with no forethought. It's recommended that you try as much as possible to think through your entire project before starting. It may be helpful to make an outline or perhaps a story board of the project, using the sequence listed above as a starting point, before beginning construction. You can do this planning mentally if you choose not to do it on your Mac or on paper.

There are no right or wrong ways to use Vision3d. But some techniques definitely are more efficient than others, because they provide the desired effects with less time and effort from you. Efficiently modeled projects also tend to render faster and look better than those that are “band-aided” together or built without regard to the number of objects being generated and the memory overhead for huge mega-models.

Additionally, *Chapter 5: Using Textures* provides further insight into using surface maps for creating realistic surface detail rather than always relying on explicit geometry.

Usually, a little planning and a touch of creativity are all that’s necessary on your part to keep from “painting yourself into a corner.” That is, to avoid a situation where the only way to complete the project you’ve spent days working on is to buy a more powerful Mac and more RAM!

About the interface

The operating metaphor for the Macintosh Interface is the desktop. The graphics on the monitor screen are designed to simulate the top of an office desk upon which you may keep business, artist’s and other tools, as well as the appropriate documents or files. Files may be stored in folders with other related files; files may be moved in and out of folders or from one folder to another; and files or complete folders can be thrown away in the trash…but can be retrieved before emptying it.

The mouse cursor is an extension of your hand as it moves over the desktop manipulating icons that represent different tools. All these actions and events are similar to the way actual files and papers would be handled on a real desk.

The Vision3d interface is designed to be an extension of the Apple Macintosh desktop:

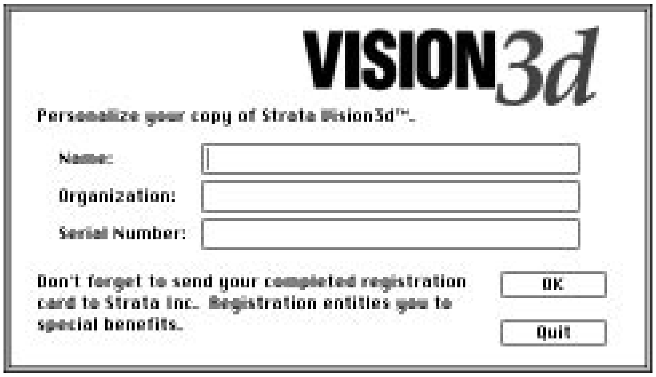
- **Its interface and operation are consistent with the desktop metaphor.** Vision3d follows the Apple Interface guidelines.
- **It is simple to learn and to operate.** You can obtain amazing results very quickly, yet it is powerful and responsive to your advanced techniques.
- **You control it through menus and commands, mouse-guided tools, and the keyboard.** As you become familiar with the interface, the operations become second-nature.
- **Your actions have direct results, with immediate feedback.** If the execution of a command will not be instantaneous, some feedback is always provided to communicate internal processing and activity.

The consistency of this metaphor results in a seamless integration of the computer hardware, operating system, and Vision3d software that communicates what it is doing at all times, is intuitive to use, and is familiar in its operation. But most important, it delivers the expected results.

The first time you launch Vision3d

The first time Vision3d is run, you will be asked to personalize your copy of the program. You will only need to enter the personalization information the first time, then that information will appear in the start up screen in subsequent sessions. Assuming the Strata Vision3d Preferences file is properly installed in the System Folder, this is all you do to start the first time:

1. **Double-click on the Vision3d application icon.** A dialog box appears asking you to personalize your working copy



Personalize your copy of Strata Vision3d™.

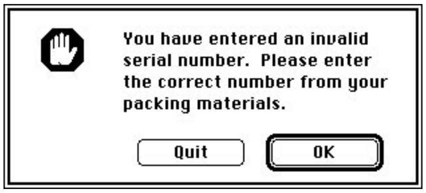
Don't forget to send your completed registration card to Strata Inc. Registration entitles you to special benefits.

2. **Type your name, organization (if any), and serial number.** The organization is optional, of course. But the serial number is not. Please refer to the registration card for your serial number. The registration card has two parts — one side is kept for reference, the other side is returned to Strata Incorporated. Please take the time to fill it out and mail it back. Strata will even pay for the postage if mailed within the United States. Note that this card or a facsimile must be received by Strata Incorporated before you can become a registered owner and eligible for technical support, to receive the quarterly newsletter, as well as other services and special offers.

   Of course, you have the option to quit before entering this information. If you do, this same dialog will appear again the next time you launch. You cannot go beyond this point to begin using the software until the information requested is entered.

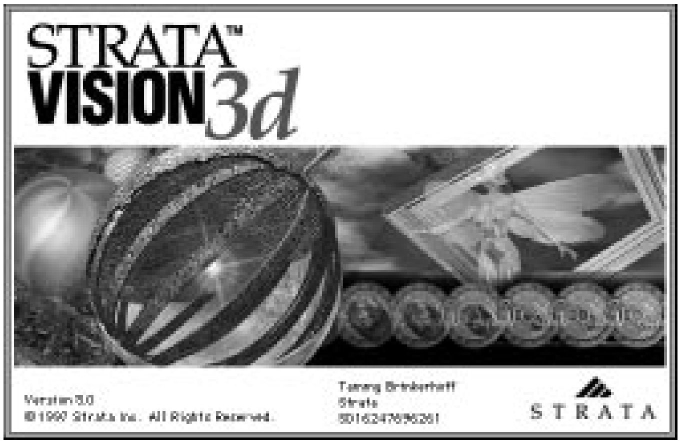
3. **Click on the OK button to have Vision3d accept the information entered.** Be sure it is correct…you will not be able to change it later. The software is imprinted with this information.

If the serial number is not a valid number, the alert shown in the following illustration appears. You then have the option to re-enter the serial number or quit.



You have entered an invalid serial number. Please enter the correct number from your packing materials.

4. **The start up screen is displayed.** This screen appears each time Vision3d is launched. It contains Vision3d's copyright notice and version number. The personalization you enter as owner of this software is also displayed on this screen.



Version 3.0
©1997 Strata Inc. All Rights Reserved.

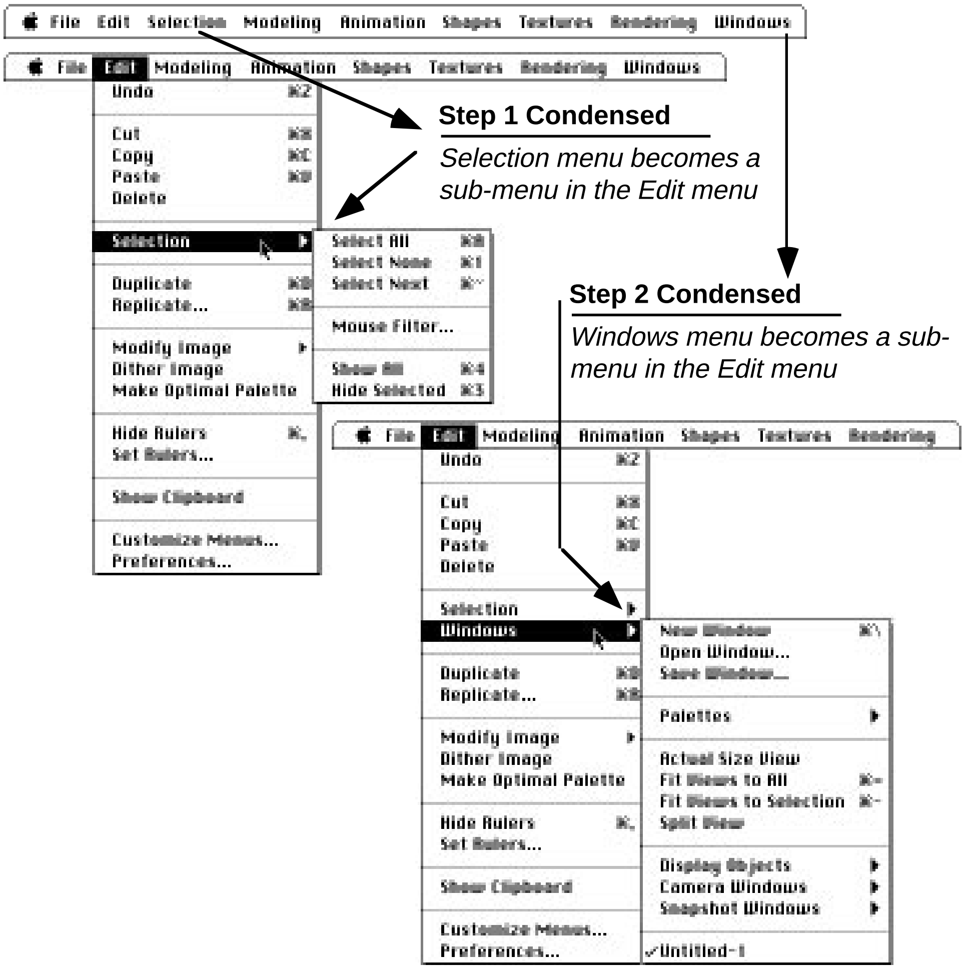
Tammy Brinkerhoff
Strata
SD16247696261

When the menu bar and palettes for Vision3d appear, you are ready to create a new model or open an existing one to begin a session. The start up screen remains displayed until the mouse button is clicked, a menu is pulled down, or a menu command is executed using its keyboard equivalent.

The menu bar

When using Vision3d, you can choose commands from up to ten menus: Apple menu, File menu, Edit menu, Selection menu, Modeling menu, Animation menu, Shapes menu, Textures menu, Rendering menu, and Windows menu. However, some hardware configurations may cause Vision3d to display a menu bar with as few as eight menus.

At start up time, Vision3d determines the size of the screen where its menu bar is to be displayed and adjusts the menu configuration accordingly. This is done to fit the menu bar to the space available — larger monitors can accommodate longer menu bars. All configurations provide full access to the program's features. The difference is that in the condensed menu configurations, some of the menus are converted to sub-menus within other menus.



Step 1 Condensed

Selection menu becomes a sub-menu in the Edit menu

Step 2 Condensed

Windows menu becomes a sub-menu in the Edit menu

The full-width menu is displayed only on monitors with a horizontal resolution greater than 640 pixels. Typically, this would apply to monitors in the 16 to 21 inch range. Step 1 condensing occurs on monitors 640 pixels wide. Apple's 14 inch RGB and 12 inch Gray Scale monitors meet this specification, as do many third-party products. This is the most popular size for Macintosh-compatible monitors. Step 2 condensing occurs on monitors with less than 640 pixels horizontal resolution. These include Apple's 12" color monitor for the Macintosh LC, the built-in screen of the Color Classic, and the built-in 9 inch B&W screen on the compact Macs. The SE/30 is the only Macintosh using a screen smaller than 640 pixels wide that is capable of running Vision3d without additional hardware.

Pulling down a menu displays the list of commands in that menu. Not all commands within a menu are active, only those relevant to the current situation. You can choose a command if it is in solid letters. Dimmed commands are unavailable.

A keyboard shortcut is displayed to the right of most commands. You may change the existing keystroke equivalent or add new keystroke equivalents for any command in the menus using the "Change Command Keys" command in the Edit menu. For more explanation on this command, refer to that command later in this chapter.

When more than one model is open, the commands and tools apply to the currently-active model only. This is true in every situation except the Windows menu. ALL open modeling windows from ALL models are listed at the bottom of the menu regardless of the active model. Selecting a window from that list makes the model contained in that window the active model.

Selecting a command from one of the pull-down menus does one of three things:

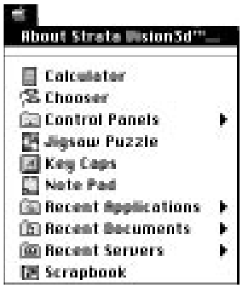
- **Performs an action directly.** Some actions are instantaneous, some require more time to complete. Upon completion of the command, the program waits for the next command.

  Many actions in Vision3d are interruptible, i.e. — you do not have to wait for the action to be completed before executing another command. Screen redraws in modeling windows, drawing multiple selection handles on an ungrouped or faceted object, and rendering an image are examples of actions that can take time to complete, but which will proceed in the background when you pull down a menu to execute another command. Background processes continue even when opening modal dialog boxes.
- **Displays a sub-menu.** In the sub-menu, options are provided to choose an action or a parameter that affects future actions.
- **Opens a dialog box.** The dialog box contains editable fields and initiates immediate or future actions based on the setting in those fields.

Menu-by-menu

A listing of each menu follows. The purpose is to understand the organization of the various commands. Individual commands are NOT discussed here. They are grouped in the chapters later in this manual relating to the specific feature, task, or operation they perform. Please refer to those chapters for a discussion of the individual commands.

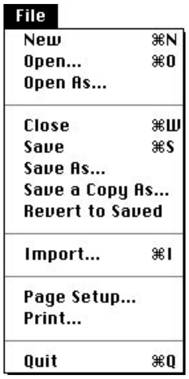
Apple menu



The Apple menu contains commands for Macintosh desk accessories, along with a specific Vision3d entry: the "About Vision3d" command. For a description of how Macintosh desk accessories are inserted into this menu and/or accessed, refer to your Macintosh owner's guide.

The About Vision3d command opens a dialog box that provides important information about the status of the application as it is running. This includes the allocation of memory inside the Vision3d partition and the number of objects in the current model. This dialog is non-modal, but remains in front of the active model window at all times. You may find it helpful to keep this dialog open to monitor the internal operation of Vision3d, especially during renderings when memory is at a premium.

File menu

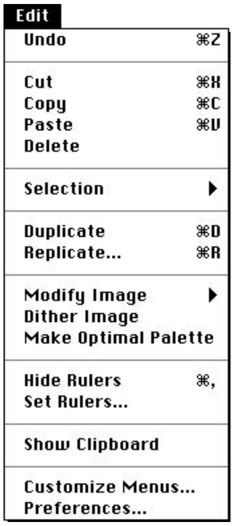


The File menu includes commands that relate to handling models and other documents. File is the term most commonly used when referring to electronic documents.

The File menu's commands are grouped as follows:

- **The first group** creates or opens documents. These may be Vision3d models, files containing 3-D data from other applications, or pixel-based images to be viewed. The documents to be opened must be in formats recognized by the application.
- **The second group** allows models and images to be closed, saved and copied. A command is also provided to discard the last set of changes to a document by reverting to a previously-saved version.
- **The third group** allows importation of Vision3d models and other files for the purpose of merging the data with the current model.
- **The fourth group** sets up printing options to control the way Vision3d prints and sends images to the printer or other output devices designated in the Chooser dialog box.
- **The last command** exits the program.

Edit menu

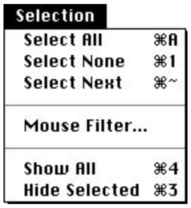


The Edit menu includes commands for editing objects, light sources, and textures. Commands are also provided to use the rulers and change the default specifications for the program.

The Edit menu's commands are grouped as follows:

- **The first command** allows the undoing of certain actions.
- **The second group** provides the standard clipboard editing commands found in all Macintosh programs. These commands can be used on objects, light sources and surface maps in textures, depending on the focus of the program when they are used.
- **The third partition** appears only if the menu bar is condensed for 14" or smaller monitors. On larger monitors it appears as a separate Selection menu.
- **The fourth group** provides enhancements to commands in the second group for copying and/or pasting multiple objects with automated position and scale parameters.
- **The fifth group** allows you to set the ruler and grid units and show or hide rulers in the active window.
- **The sixth partition** displays a window showing the contents of the Clipboard.
- **The last group** allows customizing of menu commands and provides access to the preferences dialog, which controls the general operation of the program.

Selection menu

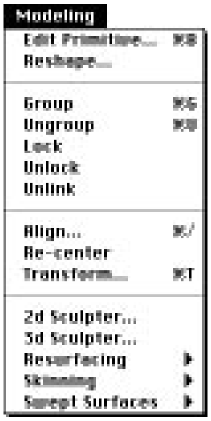


The commands in the Selection menu are designed to facilitate the handling of objects in complex models. The Selection menu allows you to specify how objects are selected in all circumstances. These commands also allow you to identify objects matching specified search parameters. Note that on smaller monitors (with 640 or less horizontal pixels), this entire menu becomes a command with a sub-menu in the Edit menu.

The Selection menu's commands are grouped as follows:

- **The first group** enables selection of all objects, de-selection of all objects, and selection of objects within or behind other objects.
- **The second group** allows you to specify parameters, which are used whenever a selection attempt is made via clicking in the model with the mouse.
- **The third group** provides for hiding selected objects and un-hiding all hidden objects.

Modeling menu

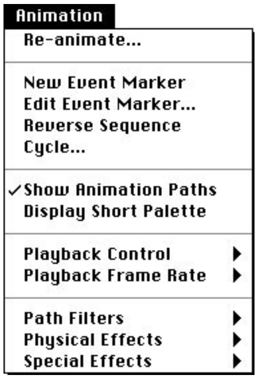


The Modeling menu contains commands for working with primitives, light sources, polygonal shapes, and individual polygons — in other words, objects. The bottom of the Modeling menu is where any installed modeling extensions appear.

The Modeling menu's commands are grouped as follows:

- **The first group** provides fine tuning an object's surface or converting an object to another type for more extensive modifications.
- **The second group** provides the ability to group/ungroup multiple objects and lock/unlock an object or objects; preventing items from being moved or edited accidentally. A command is also provided to break parent/child links established through the Link tool.
- **The third group** allows you to accurately position and align selected objects in 3-D space through special dialogs.
- **The fourth group** accesses the currently-installed modeling extensions. Commands for each modeler appear in this part of the modeling menu when the modeling extensions are installed in the Extensions folder.

Animation menu

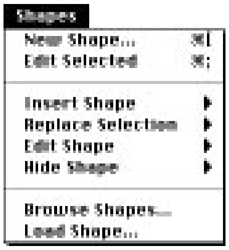


The Animation menu contains commands that provide additional control to the powerful animation features in Vision3d. They work in conjunction with the Animation palette. Each object has its own independently-definable internal action "script." The scripts may be modified in certain ways by the commands in this menu.

The Animation menu's commands are grouped as follows:

- **The first command** allows you to recalculate a particular animation effect that was generated with one of the animation extensions.
- **The second group** manages event markers. You can add a marker to an object's path or modify existing ones for other effects. Commands in this group become available only when an object is selected and its event markers are visible in the Animation palette.
- **The third group** controls the display of animation information, both on the animation palette and in the modeling window.
- **The fourth group** sets parameters when playing animation files in an animation window. Commands in this group become available when an animation window has been opened and is the active window.
- **The fifth group** contains commands to use any currently-installed animation extension. Animation extensions create special effects for chosen objects during the scripting of an animation.

Shapes menu

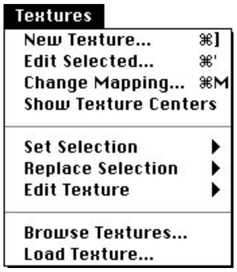


The Shape menu contains commands for creating shapes that can then be used to construct a model, for inserting and replacing shape instantiations in the model, and for navigating through the multiple levels of a hierarchical model constructed using shapes.

The Shapes menu's commands are grouped as follows:

- **The first group** enables the creation of new shapes and opening shape workspaces for inserted shape instances when selected in the model space.
- **The second group** provides controls for inserting shapes into the model space, exchanging selected shape instantiations with instances of other shapes, and for editing or hiding instantiations.
- **The third group** allows access to existing shapes for editing, re-naming, deleting, saving, etc., as well as loading other shapes into the current model from the shape folder on the hard-disk drive.

Textures menu

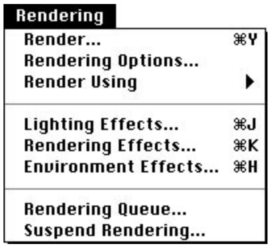


Commands in the Textures menu are the primary means of defining and applying surfaces to objects. Textures include color, grain, transparency and a host of other properties that determine how a surface reflects or transmits light. Texture descriptions can simulate virtually any surface, ranging from paper and plastic to metals and marble.

The Textures menu's commands are grouped as follows:

- **The first group** provides commands to create new textures, edit existing textures, and control the way a texture is mapped on a selected object or group of objects.
- **The second group** allows you to apply textures to objects and switch textures on selected objects.
- **The third group** manages existing textures in the current model, as well as allows loading other textures from the Textures folder on the hard-disk drive or any other volume.

Rendering menu

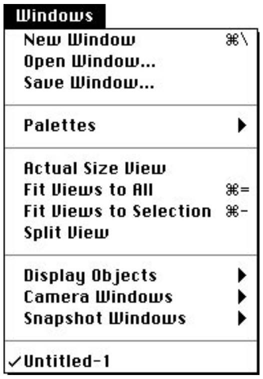


The Rendering menu includes commands for specifying the various options pertaining to rendering images and for managing the suspending and restarting of renderings.

The Rendering menu's commands are grouped as follows:

- **The first group** allows you tochoose options for and to initiate renderings of both single images and animation sequences. Provision is also made in this group to switch between the internal renderers and available external rendering methods.
- **The second group** provides the means to set or disable various rendering, lighting and environment effects, which determine how images appear when rendered.
- **The third group** provides for suspending renderings in progress and for adding or removing suspended renderings from queued rendering sessions.

Windows menu



The Windows menu includes commands to control which windows are displayed on the screen and to indicate which palettes are open or closed. There are also commands to adjust the proportions and size of modeling windows and to set the level of detail at which objects are displayed.

Groups of related commands in the menu are partitioned by dotted lines:

- **The first group** provides commands for opening new windows for the current model, opening previously-saved windows and saving custom window configurations.
- **The second group** allows displaying or hiding of the various palettes. Palettes provide access to tools, lighting, textures, shapes, modeling, and animation controls.
- **The third group** provides quick adjustment of the view scale for objects displayed in windows. There is also a command in this group to split windows into multiple panes
- **The fourth group** provides a command to choose how accurately to display objects, to open or close camera windows, and monitor the status of in-progress rendering windows.
- **The fifth group** lists all open modeling windows of all open models and indicates which one is the active window.

Palettes and buttons

In addition to the menus and commands that make up the Menu bar, Vision3d uses a Button bar and other special purpose palettes on which various tools and controls are grouped according to their purposes. There are five such palettes:

- **Button bar**
- **Tool palette**
- **Lighting palette**
- **Textures palette**
- **Shapes palette**

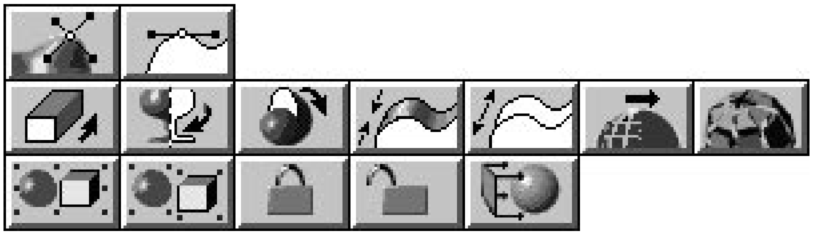
Of course, the Menu bar is the primary control mechanism for Vision3d. These palettes are considered supplemental interface elements for facilitating the application's ease of use.

A listing of each palette follows. The purpose is to understand the organization of the various palettes. Individual controls or buttons on the palettes are NOT discussed here. Each palette is presented in a chapter later in this manual relating to the specific feature, task, or operation they perform. Please refer to those chapters for a discussion of the individual palettes.

Button bar

The Button bar resides on the screen directly below the Menu bar. It is not movable. It provides a graphical shortcut to commands found in the Modeling and Animation menus. Use the buttons on the Button bar for one-step, on-screen access to the modeling operations you use most often.

As with menu commands, buttons are active only when they are relevant. A button is grayed out or dimmed if its operation does not apply to the current situation.

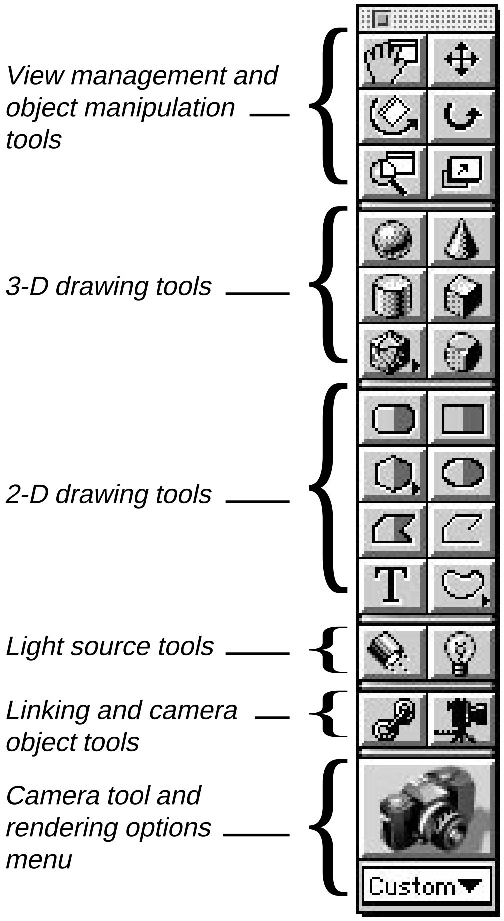


The number of buttons on the Button bar depends on the modeling and animation extensions currently installed in the Vision3d Extensions folder. If more buttons are present than will fit in the width of your screen, the horizontal arrow buttons on the left of the Button bar cycle to allow you to view and use more than one row of buttons.

The Button bar is discussed in *Chapter 3: Modeling Objects*.

Tool palette

The Tool palette differs from the Button bar. While both have functions relating to modeling processes, the Tool palette specifically contains all of Vision3d's drawing and object manipulation tools.


View management and object manipulation tools
3-D drawing tools
2-D drawing tools
Light source tools
Linking and camera object tools
Camera tool and rendering options menu
Custom▼

Tools on the Tool palette are grouped according to their functions:

- **The first group** are for view management and object manipulation.
- **The second group** are the 3-D drawing tools.
- **The third group** are the 2-D drawing tools.

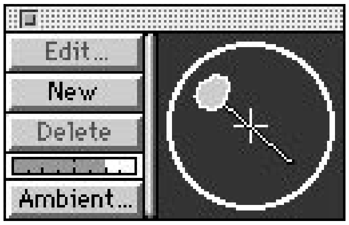
- **The fourth group** are tools to insert local light sources.
- **The fifth group** are tools to create parent/child links and insert camera objects.
- **The sixth group** is one large button representing the Camera tool. Do not confuse this tool with the Camera Object tool above it.

The last item on the Tool palette is not a tool. It's a special pull-down menu listing the pre-defined rendering options for the camera tool.

The Tool palette is discussed in detail in *Chapter 3: Modeling Objects.*

Lighting Palette

The Lighting palette contains the buttons and icons needed to add, delete and position Directional light sources. Directional lights are like sunlight. They have parallel rays of light that illuminate an entire model from an infinite distance.

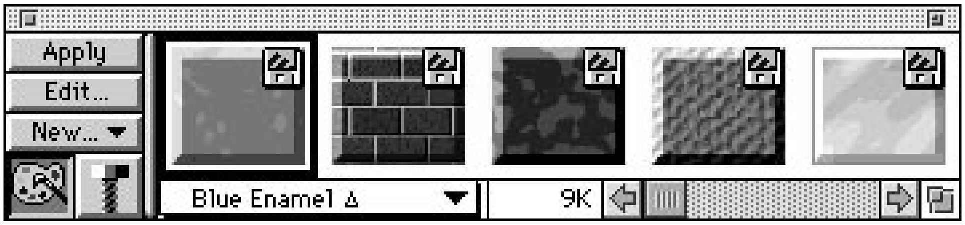


Buttons are also provided on this palette for adjusting lighting intensity and applying lighting gels or color to selected lights. Additionally, another button allows you to adjust the intensity and color of the ambient or fill illumination in a model.

The Lighting palette is discussed in *Chapter 6: Managing Lighting*.

Textures palette

The Textures palette provides the controls to manage textures. Each texture found in the Textures folder at the time Vision3d is launched is represented on the Textures palette by a color preview.

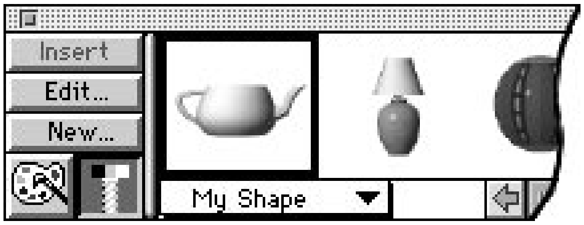


Textures that are created or loaded into the current model are also given a space. The Textures palette is often an easier, more graphical way to access textures than using commands in the Textures menu. From this palette, you can create new textures, apply textures to selected objects, and modify or alter existing textures.

The Textures palette is discussed in *Chapter 5: Using Textures*.

Shapes palette

The Shapes palette provides the controls to manage shapes. It shares screen space with the Textures palette and is accessed by clicking on the Shape button in the bottom left corner of the palette.



Texture and Shape buttons toggle palette function.

Each shape found in the Shapes folder at the time Vision3d is launched is represented on the palette by a color preview. Shapes present in the current model are also given a space. The Shapes palette is often an easier, more graphical way to access shape workspaces than using commands in the Shapes menu.

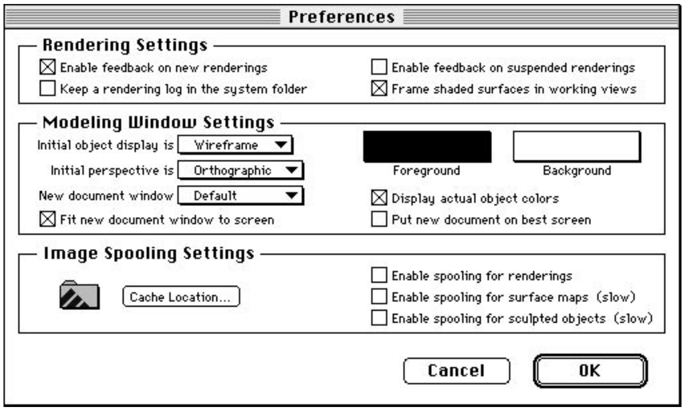
From this palette, you can create new shapes in the current model, insert shapes into the model, and modify or alter existing shapes.

The Shapes palette is discussed in *Chapter 4: Using Shapes*.

Setting preferences

Choosing the Preferences command in the Edit menu opens the preferences dialog box. This dialog contains fields for specifying the general operation of the software. The preferences settings are global in nature. That is, they affect not only the current model, but any other models that you create or load afterward.

NOTE: *Models are saved with their current configuration. That means a model remembers what windows were open, the display method, perspective setting, etc. When you load an existing model, its specific configuration — if different than the preferences settings you have chosen — will temporarily override the current preferences for those items.*



Preferences

Rendering Settings

☒ Enable feedback on new renderings
☐ Keep a rendering log in the system folder
☐ Enable feedback on suspended renderings
☒ Frame shaded surfaces in working views

Modeling Window Settings

Initial object display is Wireframe
Initial perspective is Orthographic
New document window Default
☒ Fit new document window to screen
☒ Display actual object colors
☐ Put new document on best screen

Image Spooling Settings

☐ Enable spooling for renderings
☐ Enable spooling for surface maps (slow)
☐ Enable spooling for sculpted objects (slow)

A preferences file is provided on the installation disks. This file contains some preset rendering configurations and settings for controlling the Raytracing and Raydiosity™ renderers. It should be placed in your System Folder. If no preferences file can be found when Vision3d is launched, the software will alert you to this deficiency then quit. THE STRATA VISION3D PREFERENCES FILE MUST BE IN THE SYSTEM FOLDER BEFORE THE APPLICATION WILL RUN.

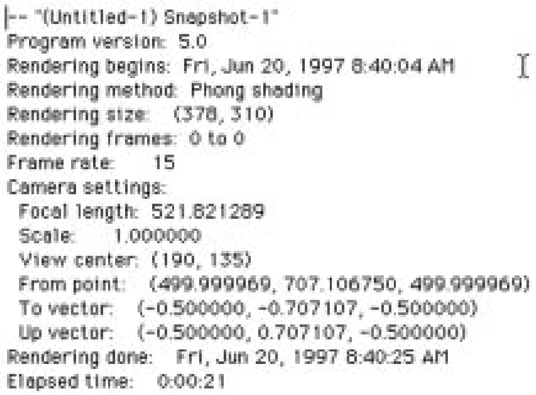
Some preferences affect the appearance of the model; others tell Vision3d where its support files are located. An explanation of the fields in the Preferences dialog box follows.

Rendering Settings

- **Screen feedback on new renderings.** If the feedback is enabled for new renderings, the rendering window opens with screen feedback turned on. You may watch an image take shape. The rendering window will show the progress of a rendering in real time, as it "develops."

  If the screen feedback is disabled, images still render internally, but no information is sent to the screen. By turning off the visual feedback, a rendering can often proceed faster.
- **Screen feedback on suspended renderings.** This field controls whether screen feedback is on for suspended renderings when they are restarted. Its behavior is identical to the previous description of screen feedback for new renderings.
- **Keep a rendering log in System Folder.** This field turns on a rendering log that is stored in the same location of the System Folder as the Preferences file.

The rendering log contains a variety of information, including start time, finish time, and time to render.

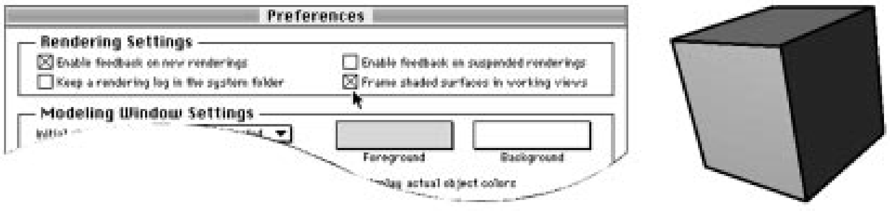


```
-- "(Untitled-1) Snapshot-1"
Program version: 5.0
Rendering begins: Fri, Jun 20, 1997 8:40:04 AM
Rendering method: Phong shading
Rendering size:   (378, 310)
Rendering frames: 0 to 0
Frame rate:     15
Camera settings:
  Focal length: 521.821289
  Scale:      1.000000
  View center: (190, 135)
  From point:   (499.999969, 707.106750, 499.999969)
  To vector:   (-0.500000, -0.707107, -0.500000)
  Up vector:   (-0.500000, 0.707107, -0.500000)
Rendering done:   Fri, Jun 20, 1997 8:40:25 AM
Elapsed time:   0:00:21
```

The rendering log is a simple text file that logs start and end times for rendering projects. It also documents the speed of the rendering (in pixels per second), as well as the conditions under which a rendering in progress has been terminated, if applicable.

- **Frame shaded surfaces in working views.** A check box is provided here to choose whether you want frames on shaded objects in the modeling windows.

Enabling the "Frame shaded surfaces in working views" option in the preferences dialog produces quickshaded objects with a black frame around each polygon when shaded objects are used in the modeling windows.

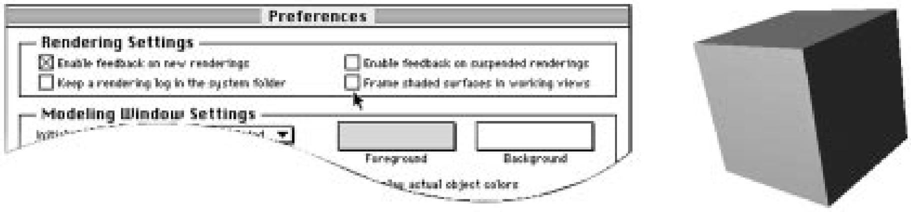


Preferences

Rendering Settings

☒ Enable feedback on new renderings
□ Keep a rendering log in the system folder
□ Enable feedback on suspended renderings
☒ Frame shaded surfaces in working views

Modeling Window Settings



Preferences

Rendering Settings

☒ Enable feedback on new renderings
□ Keep a rendering log in the system folder
□ Enable feedback on suspended renderings
□ Frame shaded surfaces in working views

Modeling Window Settings

Disabling the "Frame shaded surfaces..." options results in no frame on shaded objects in the modeling windows. Depending on the lighting and surfaces, objects without lines around the polygons can sometimes be hard.

On occasion, you may wish to view the scene you're assembling as shaded objects, with and without frames around the objects. Some objects appear better with frames than others, but you have the option of choosing how they will appear. This may be preferable to taking flat shaded preview snapshots, since the objects can continue to be modeled and adjusted.

This preference option does not affect the way objects are handled in the model, only how they are represented in the modeling windows.

Modeling Window Settings

- **Default Display method:** The display method used when a new model window opens is set with a pop-up menu for the "Initial object display" field. There are three choices in this menu:

  1) Wireframe
  2) Hidden line
  3) Shaded

  Of course, these choices correspond to the three display-method buttons at the top of each modeling window. These renderers are available for the modeling windows' views. The differences should be obvious just by looking at a model using the different display methods.

  A discussion of these display choices is presented along with the other renderers later in the manual.

- **Default Lens perspective:** The default perspective used when opening new modeling windows is set by the "Initial Perspective" field. The pop-up menu allows you to choose from three settings:

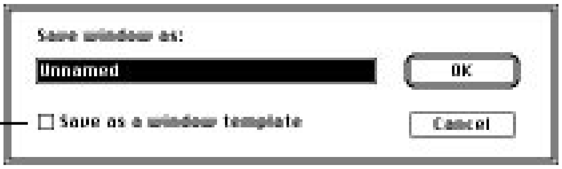
  1) Orthographic
  2) Normal
  3) Wide Angle

  These likewise correspond to the three-position perspective control found at the top of each modeling window. The three choices indicate how the views in the new window present objects in the model. They indicate how much perspective is in the display, similar to the focal length setting of a camera lens.

  Orthographic displays present the parallel lines of objects in the model as parallel lines in the views. There is no vanishing point, no convergence of lines with distance. Orthographic displays preserve the absolute dimensions of objects, so they can be sized accurately relative to each other or to the rulers. In orthographic displays, rulers show the scale of the object in units chosen in the Set Rulers command.

On the other hand, a perspective display (the Normal and Wide Angle settings), produce more realistic views but do not maintain actual or relative dimensions. Distant objects are displayed smaller than those closer to the viewing plane, even when they are the same size.

- **Default window arrangement:** The configuration for opening new modeling windows is set through this pop-up menu. The default is a single pane displaying a Front view of the model. If other configurations have been saved as general window templates using the Save Window command, you may select them here as the configuration for new windows.



Save window as: Unnamed

☐ Save as a window template

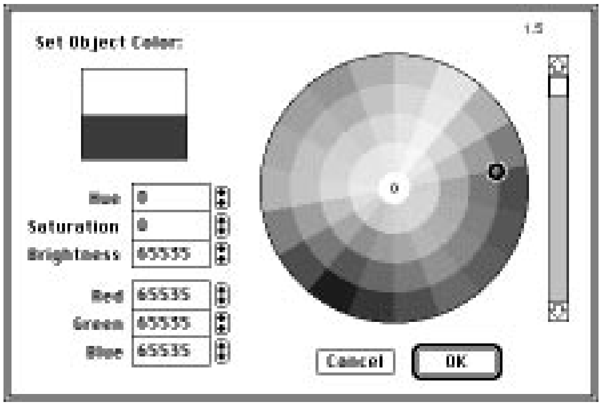
Click on this checkbox if you want the window configuration available to all models

You may define and save other configurations that will then appear in this pop-up menu through the Save Window command in the Windows menu. You may save a window configuration that you would like to use any time a model is open. Once the window is set up, including multiple view panes, their sizes and view position, Select the Save Window command from the Windows menu. The dialog allows you to name the window arrangement and choose whether the saved window is to be part of this document only or a template available for use in other documents. Windows saved as global definitions are available in all models, including this pop-up menu. Windows that are not global are saved with the current model only.

- **Foreground/Background colors:** The “Model Windows Colors” fields determine the colors used to display the model in the modeling windows. Obviously, you should use colors that are complementary, yet with enough contrast to see the information.

To change either foreground or background color:

1. **Move the cursor to the box representing the color you wish to change.**
2. **Click once in the box.** The standard Apple Color Picker dialog box appears. The operating system opens a dialog box representing the colors available for the monitor on which it is being displayed. There is even a B&W representation of the color picker.



Set Object Color:

Hue 0
Saturation 0
Brightness 65535
Red 65535
Green 65535
Blue 65535

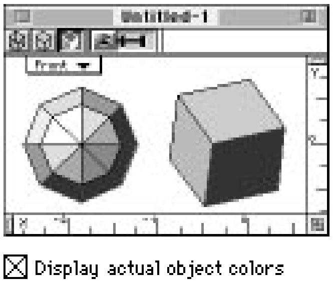
3. **Select the color for that field.** You may either click the mouse cursor on the color you want, or enter the values for that color in the hue, saturation and brightness or red, green and blue fields. The numeric fields allow you match exact colors.

NOTE: *If multiple monitors are installed on your Macintosh computer, Vision3d will ALWAYS open the color picker on the monitor with the most color depth available.*

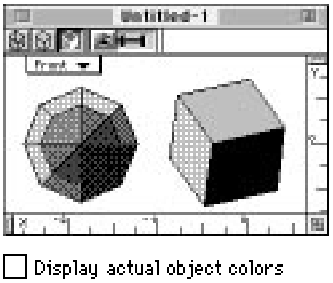
Changes in the foreground or background colors (or both) apply immediately to the current model when the OK button is clicked, as well as for all future models to be open. The foreground and background colors are not saved with the model.

- **Display actual object color:** The colors used to display objects in modeling windows may be the simple foreground color or the actual colors of the objects, as determined by the texture currently applied to them. Using actual color is enabled with the "Display Actual Object Colors" check box.

Note difference when "Display actual object colors" is selected.



☒ Display actual object colors



□ Display actual object colors

If this item is disabled, the modeling windows display objects in the designated foreground color ONLY. Objects displayed with the foreground color only will appear as if your monitor's screen is in B&W/monochrome mode. This is especially apparent when a window is using the Quick Shade display method.

If actual object color is enabled, the applied texture is used for the object's representative color, even in wireframe display. If the texture uses a color map or pattern, either of surface or solid type, a single color is chosen for that object in the modeling windows that is an average of the map's colors. No attempt is made to track any color variations across the map.

Remember, objects with no texture have a default opaque matte white surface. Thus, these objects appear with a white representation in the modeling windows. If the background color is also white, or close to white, the objects may become invisible or difficult to see. To avoid potential confusion in this area, you should try to pick a background color that is unlikely to be matched by object colors.

- **Fit new document window to screen:** When opening a new model, you can choose whether the window that opens is sized to fit the full size of the monitor or simply opens at a smaller, default size.

  If this preference option is enabled, the window extents will be set as large as possible without overlapping any open palettes. That means, if the Button bar is visible at the top of the screen plus the Lighting and Textures/Shapes palettes are showing at the bottom of the screen (where they are designed to fit), the new model window will be set up to fit just under the Button bar and just above the palettes at the bottom. If the Animation palette is also showing and positioned just above the Textures/Shapes palette, the window will be automatically shortened so that it will not overlap the Animation palette.

  If disabled, the window opens at a standard size and does not adjust for any palettes that may be showing. This may be desirable when the window is opening on a large monitor, but you do not want it to fill the entire screen.

- **Put new document on best screen:** In hardware configurations where more than one monitor is present, you may want the new model's window to open on the monitor with the highest color depth.

  If enabled, the new window will open on the best color monitor. If there is no difference in color depth between the monitors, i.e. — they are all set at 8-bit/256 colors, the window will open on the monitor where the Menu bar appears.

  If this option is disabled, the new window also opens on the monitor where the Menu bar appears.

Image Spooling Settings

Image spooling is the process of setting up a buffer or cache on the hard-disk drive. A cache is a temporary memory or hard-disk storage allocation used by Vision3d to facilitate the rendering process. The use of caches usually involves a trade off between speed and memory, but not always. Several rendering processes require caches, especially when using Raytracing, to make more room available when handling images that cannot fit completely in RAM.

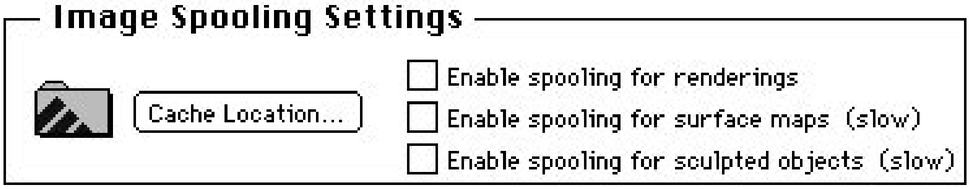


Image Spooling Settings

□ Enable spooling for renderings
□ Enable spooling for surface maps (slow)
□ Enable spooling for sculpted objects (slow)

You are given the option of using or disabling three different caches depending on your rendering priorities. The proper configuration of Vision3d's cache options can mean the difference between a rendering being possible or not.

Vision3d requires more memory for rendering than for modeling. The higher the rendering quality, the more memory needed. This is an unavoidable reality, stemming from the render's method of "seeing" the model. Vision3d and System 7 cooperate in the managing of memory requirements in order to successfully render an image.

When setting up for rendering, the general rule is to try rendering with all spooling options turned off. If you have sufficient RAM to render the entire image in memory, that is the preferred configuration.

If Vision3d indicates there is not enough memory, you can enable the "spooling for renderings" option first and leave the other two spooling caches off. If the rendering is still not possible, you may have to turn on the second spooling cache as well (for surface maps) and try again. If this still does not free up enough memory, then you must choose a lesser-quality renderer, a smaller rendered image, reduce the number of objects and the size of texture maps in your model, or allocate more RAM to Vision3d to be able to render. Any combination of these changes can free up sufficient memory to allow the rendering to occur.

The third cache option (for sculpted objects) applies only for a special class of object called a Sculpt object and does not generally enter into this memory management equation.

An explanation of the settings in this portion of the Preferences dialog box follow on the next page.

- **Setting cache location:** The Cache Location button allows you to designate the volume on which Vision3d's "virtual image" file is to be created. Clicking this button opens the standard directory dialog containing directory navigation controls. You may put the spooling file on any mounted volume to which you have write access, including removable media like SyQuest or Magneto-Optical drives. Note that faster hard-disk drives provide better performance.

  When you have navigated to the desired location, so that the contents of the volume or folder within that volume are visible in the directory dialog, you can click either on the "Select Current folder" button at the top of the dialog or on the "Here" button in the bottom right corner. Clicking either button completes the location selection process and closes the dialog.
- **Spooling for Renderings:** This check box enables or disables caching of the rendered image on the designated volume. Segments of the image are then spooled into RAM for processing. The actual rendering process occurs in memory, but through the use of image caching , renderings too large for the available memory can still be created. Disabling this option forces Vision3d to keep the entire image in memory, which may render faster but can severely limit the size of rendered images.
- **Spooling surface maps:** This check box enables or disables caching of surface maps. Each map used in a texture requires a buffer where the map can be "laid out" for the renderers to look at while calculating how that map should appear on the applied object(s). If enough maps are used, or the maps used are quite large, and all of them are in memory at the same time, there may not be enough memory left to render the image. By spooling surface maps, i.e. — making them available to the rendering process in manageable segments only as needed, Vision3d can handle more (and larger) surface maps without running out of memory.
- **Spooling Sculpted objects:** This check box enables or disables caching of the sculpted object's gray-scale image used to make a 3-D representation. Caching a sculpt object allows Vision3d to deal with the geometric result in smaller chunks during the rendering of that object.

  The sculpted-object class converts gray-scale pixel-type images to elevation data and represents it in the model as a 3-D object. Modeling windows display the sculpt object in polygonal form strictly for manipulation in the model space. During the rendering process, however, handling of the sculpt object's grayscale representation is very similar to surface maps. The gray-scale image must be referenced much as a surface map would be, except that the image defines a geometric shape instead of surface patterns.

In reality, rendering is a matter of managing the available limited resources of RAM and hard disk space, as well as CPU performance. If after juggling Vision3d's memory partition and these cache settings, and the rendering just will not start, you are faced with the need to reduce the model or increase the amount of physical RAM in your machine.

Notice that nowhere in this discussion has the use of System 7 virtual memory been suggested. While using System 7 virtual memory is an option, you should know that feature in the operating system was never intended to replace physical RAM in which you could launch applications with larger partitions. It was meant only to allow multiple applications to be launched and swapped more quickly into RAM when each was made the active application. Strata Inc. makes no warranty of performance in a virtual memory environment.

Saving preference settings

Changes in the Preferences dialog box take effect immediately in the current model upon clicking on the OK button. However, the changes apply only to the current session until you quit Vision3d. At quitting, Vision3d updates its preference file stored in the System folder so that the same configuration can be used the next time Vision3d is launched.

If something happens while in Vision3d that exits the application without quitting — a system or application crash, power failure, etc. — the preferences changes are not saved to the preferences file. When Vision3d is restarted the next time, it will open the same as before the Preferences changes were made.

Navigating in 3-D space

The requirements for creating and editing 3-D models is far more complex than for working with images in 2-D applications. Two-dimensional images in a paint program usually require only one window; the shapes and colors of the drawing are displayed in the window directly; there are no issues of lighting, shading or view position or perspective, unless artificially introduced by the user as part of the 2-D image.

In Vision3d, three-dimensional objects must be projected onto the flat viewing surface of the monitor. The program is capable of showing every object in a model as it appears from any viewing position. And during the rendering process, lighting direction and intensity must be taken into account to accurately describe the appearance of the surfaces on those objects.

Think of a view plane as a gateway into Vision3d space. In addition to viewing the model space, you perform most of your scene layout in these views. Through the views, you can draw and manipulate objects, assign textures, position lights, and take snapshots with the camera tool as desired to obtain finished, rendered 2-D images.

In order to present the 3-D data of a model, Vision3d is given the ability to open multiple views which allow you to look at models from any direction. Each view within a window defines a plane upon which the 3-D objects are projected. By specifying the point in space from which to view the objects, and by establishing how much of the scene to include in the display (angle of view), an unlimited number of viewing and editing positions can be defined.

Working in scale

The scale or unit of measure in which you wish to model is set through the Set Rulers command in the Edit menu. Once set, the designated units are global in scope and used for all models. You may change the units as often as needed. Changes will be immediately shown in the current model. The size of the objects do not change; just the units used to display their measurements.

Strata Units

In reality, the size of objects in a model are defined using an internal unit, called Strata Units. When you choose a unit of scale — inches, for example — the Strata Units for that object are translated into the chosen display units. One Strata Unit is 1/72 of an inch.

Using rulers

Selecting the Show Rulers check item (in the Edit menu) toggles rulers on or off in the views in the active window. However, only views meeting specific conditions will have usable rulers; rulers will not be present in other views not meeting the criteria.

Only the Top, Bottom, Front, Back, Right, and Left home positions qualify for this requirement. Note that the Isometric view position is the only view in the “standard view set” not included in this group. Thus, only these views will have usable rulers…and only when there is no perspective present in the view.

Unlike paint or drawing programs, where the image’s display requirements are much simpler and usually limited to one window, Vision3d can display its 3-D objects from any direction and with varying perspective. But rulers are a 2-D mechanism. As such, they are only valid in views with no perspective and which are aligned to axes of the coordinate system. If handled correctly, the views would also be parallel to the faces of the objects. That is because only non-perspective views aligned to one of the absolute axes can display objects in a way that can be represented on the rulers. Accurate dimensions on the rulers are possible only when objects can be drawn in true scale and their sizes relative to each other can be visually determined.

Orthographic while modeling

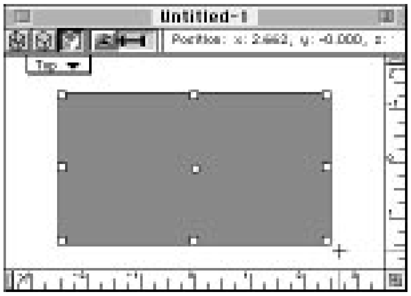
The discussion about rulers in the previous topic leads to one of the most important rules for navigating in 3-D space. That rule, printed in all caps and bold letters for emphasis, is:

"ALWAYS MODEL WITH ORTHOGRAPHIC DISPLAY, THEN SWITCH TO PERSPECTIVE FOR RENDERING."

An examination of this statement should make its usefulness obvious. As explained earlier in this chapter, displaying objects orthographically presents them in their true size relative to other objects. It also removes the problem of parallax in the view due to perspective.

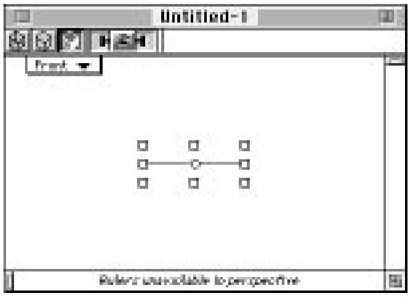
A quick manual alignment example demonstrates this concept:

1. **Upon opening a new model, draw a filled rectangle from the Top view.** The size isn't important for this purpose. Just draw it to fit comfortably in the view.



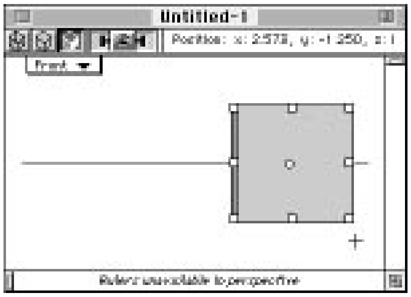
Position: x: 2.662, y: -0.000, z:

2. **Switching to the Front view,** set the 3-position lens control to its center position of Normal.



Rulers unavailable in perspective

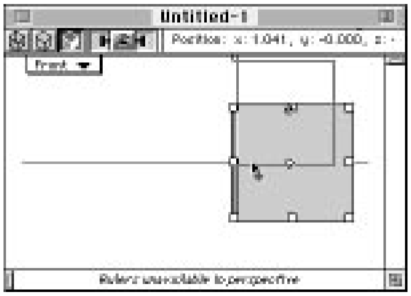
3. **Choose the "Fit Views to All" command in the Windows menu.** As the lens setting is moved from Orthographic to Normal, the rectangle appears to become smaller. It is just an optical effect, and using this command compensates by moving the view plane closer to the object.
4. **Draw a Cube in the Front view.** Again, the size isn't critical, just a comfortable size that fits in the view. It is also not critical where you draw the cube.



Position: x: 2.573, y: -1.250, z: I

Rulers unavailable to perspective

5. **Using the Object Move tool** (the tool at the top right of the Tool palette), now click and drag the cube object so that it is positioned where you think it is resting on the surface of the rectangle.



Position: x: 1.041, y: -0.000, z: -

Rulers unavailable to perspective

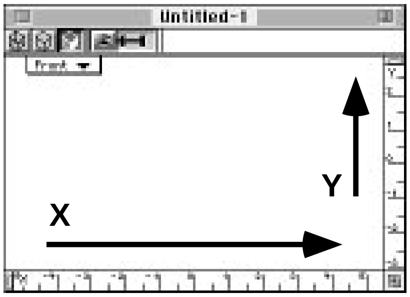
How close were you able to line-up these two objects while the view was in perspective? To check, switch the lens control back to Orthographic and choose the "Fit Views to All" command again (to compensate for the telephoto effect of the lens change). When the view redraws, you will probably find the object is not where you thought it would be.

Moral of this story (and to summarize): The ONLY way to be sure of object size and position is to model in orthographic display.

The coordinate system(s)

Vision3d uses a three-axis coordinate system, patterned after the standard Cartesian space model, where the three axes are always at right angles to each other. This model is sometimes also referred to as Euclidean geometry model. By convention, the axes are titled X, Y, and Z for horizontal, vertical, and depth respectively. When looking at the Front view, positive-X is to the right, positive-Y is up, and positive-Z is toward the view plane. To help you remember, Vision3d uses the “left-hand rule” for its coordinate mathematics: With the thumb pointing up (positive Y), the index and middle fingers point at positive X and positive Z respectively. This is also reflected in the numbers showing on the rulers.

Note that the numbers on the rulers are positive to the right (x) and positive in the up (y) directions.



Of course, this naming convention makes a basic assumption about which way is up in the 3-D space. As you will find when using the full-featured window and view system in Vision3d, views can be moved and/or rotated into virtually any orientation. So how do you keep track of the “up” direction?

Relative and absolute references

Actually, there are three coordinate systems maintained in Vision3d:

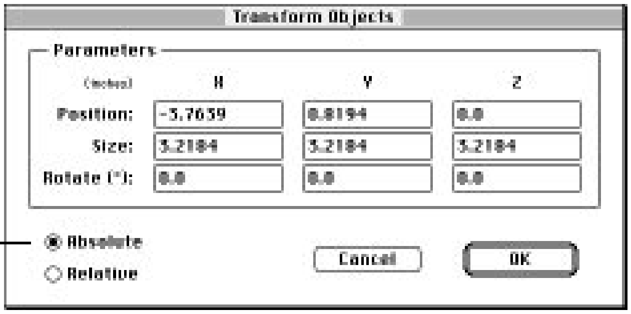
1. **Model coordinates** (absolute)
2. **View coordinates** (relative)
3. **Object coordinates** (relative)

As noted, the model space has absolute coordinates against which all objects and the other relative coordinate structures are referenced. This means a given object or group of objects can be referenced either by its absolute model coordinates or relative to the active view. It also means when an object is rotated so that it is no longer aligned to the absolute coordinates — the positive Y axis is no longer UP — it has its own coordinates to maintain a reference for things attached to it, like textures.

Object coordinates are used by textures. They are referenced to set and maintain the position of a texture on an object, even when that object is moved or rotated.

Transform dialog box

In addition to using the rulers as a reference for drawing objects to desired scale, a Transform command is provided in the Modeling menu. Choosing this command opens the Transform dialog box.



(inches)	X	Y	Z
Position:	-3.7639	0.8194	0.0
Size:	3.2184	3.2184	3.2184
Rotate (°):	0.0	0.0	0.0

● Absolute
○ Relative

Click on the Absolute radio button to see the selected object actual position, scale, and rotation in the model space

The dialog box presents the current position, scale and rotation of the selected object or group. At the bottom of the dialog, two radio buttons allow you to switch between absolute and relative coordinates. The relative setting is based on the currently-active view.

Entering numbers directly into one or more of the fields in this dialog will effect a change in the object upon clicking the OK button.

NOTE: *While you can often use the rulers of a view (or multiple views) to "draw" an object to a desired size, if the dimensions of the object are especially critical you may want to double-check that object with this Transform dialog. As with any application where the scale of its window(s) can be magnified or reduced, the accuracy of the rulers declines as the scale is reduced and improves when magnified.*

This command is active when only one object or group is selected. If no selection is present in the model, or more than one object is selected, this command is not available.

Extensions explained

Installing extensions is a simple process of placing them in the Extensions folder within the application folder. Extensions are software add-ons developed either by Strata Incorporated or third-party vendors to extend the capability and feature set of the standard Vision3d program. These separate extensions to Vision3d are typically distributed on floppy disks and represented as separate files on the desktop. Instructions for using each extension are typically shipped in the package with the extension.

Extensions are placed in the Extensions folder that must be located within the Vision3d application folder on the same level as the application itself. Its name and location are not optional. Sub-folders may be used within this folder, if desired, to organize the various extensions located there, but are not necessary. Vision3d looks in the Extensions folder for the extensions it can use each time it is launched. The extensions are then available from within Vision3d for reading or saving specific file formats, for creating objects, or to create special animation effects during a modeling/rendering session.

Depending on the type of extension, Vision3d adds that extension's features to the appropriate area of the program:

- **Import/Export.** Support for importing and exporting additional file formats will appear as options in the Open, Open As, and Save As dialogs.
- **Filters.** Post-rendering filters appear in the Modify Image sub-menu in the Edit menu. They will follow the list of built-in filters always found in that sub-menu.
- **Modelers.** Modeling extensions, which provide additional modeling tools for creating unique object types are appended to the bottom of the Modeling menu.
- **Rendering Export.** Rendering extensions that support external rendering applications appear in the Render Using command sub-menu in the Render menu, as well as in the "Render Using" pull-down menu in the Render command's dialog box,.

Some extensions are included free of charge with the product. Other extensions are sold as an after-market enhancement. They may be purchased either from Strata Inc. or third-party developers who support the Vision3d product. For detailed instructions on a specific extension's operation, refer to the appropriate documentation provided with that extension software.

For information concerning the currently available extensions, please call the Strata Incorporated general telephone number listed at the beginning of the manual.

Windows and Views

Windows are used to access documents. An open Vision3d window displays an open document. Documents may be either 3-D models, renderings in progress, still images from any source, or animations to be played. Each type of document is supported in a different type of window, with controls to handle the information or data present in that specific document. Any combination and number of these windows may be open at one time on the desktop, depending on the available memory.

Technically, dialog boxes and alerts are also windows. This chapter deals only with document windows. If dialog boxes and alerts are discussed, they are presented in context with the operation of the document windows.

Windows are elements of the desktop interface. Their presence on the screen and their movement, sizing, opening and closing, etc. are handled by the Macintosh Operating System. The contents of the windows, however, are controlled by Vision3d. Vision3d supports multiple windows on the screen; each window showing views of a model, multiple models, other documents, or auxiliary information used in processing the documents.

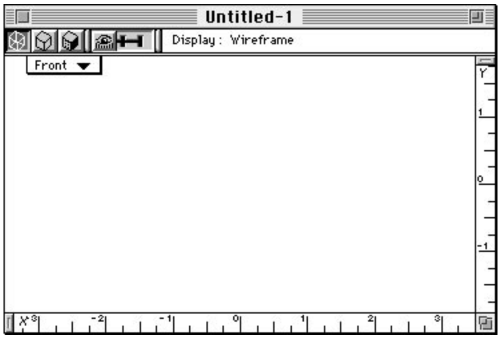
This section discusses the operations within Vision3d windows.

Quick Tutorial

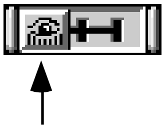
This Quick Tutorial is meant to provide a basic understanding of window operation. Follow it to go through a step-by-step exercise of the various controls found on modeling windows

To open a new model:

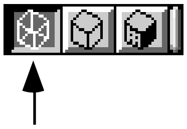
1. **Pull down the File menu from the menu bar.**
2. **Choose the "New" command.** Vision3d responds by opening a window. The default configuration for a new model is a single pane containing a Front view of the model space.



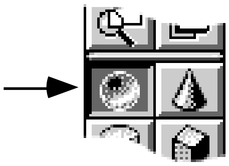
NOTE: *The specific window configuration for a new model is determined by the "New Document Window" Preferences setting. Please refer to the discussion about the Preferences dialog box earlier in this chapter.*



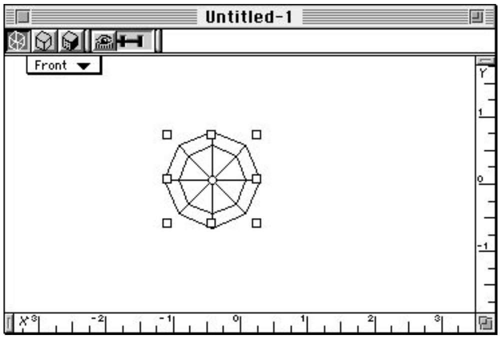
3. **Set the three-position lens control to Orthographic.** This allows you to see the absolute sizes of objects relative to each other. That's important for accuracy when aligning or scaling objects. Orthographic ignores size as a factor in the view.



4. **Set the display method to wireframe.** Like window configuration, the lens control position and the display method for a new model can be set in Preferences.



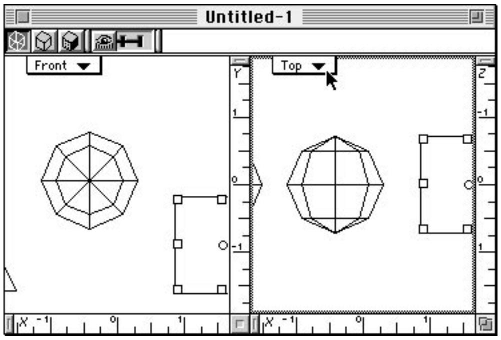
5. **Select the sphere primitive tool from the Tool palette.**
6. **Draw a sphere in the model window.** You can do this two ways: 1) Click once in the window to place a sphere at the default size or 2) click once in the window and drag until the marquee is the approximate size you want the sphere to be. Release the mouse button to place the sphere in the window.



7. **Place several more primitives of your choosing in the window.**

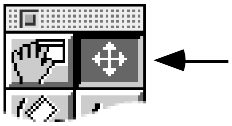
8. **From the Windows menu, choose the "Split View" command.** Vision3d responds by splitting the window vertically into two identical panes.

9. **In the right pane, click on the view title.** A pull-down menu appears listing the standard view positions.
10. **Choose the Top view from the view title menu.** Vision3d responds by redrawing the view in the right pane from the top orientation. Contrast this view with the original Front view in the left pane.



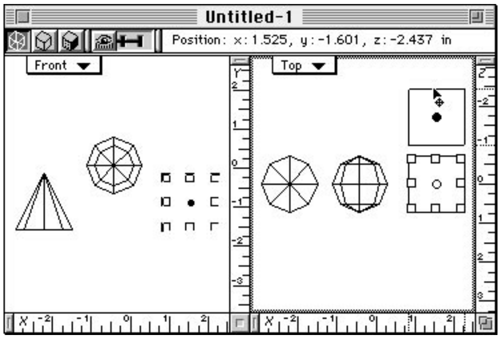
If all the objects are not fully visible in the views after splitting and changing view orientations, you can use the Fit Views to All command in the Windows menu. This adjusts the scale of the views to show all the objects in the model.

11. **Select the Object Move tool from the Tool palette, if not already selected.** This is the tool at the top right of the palette.

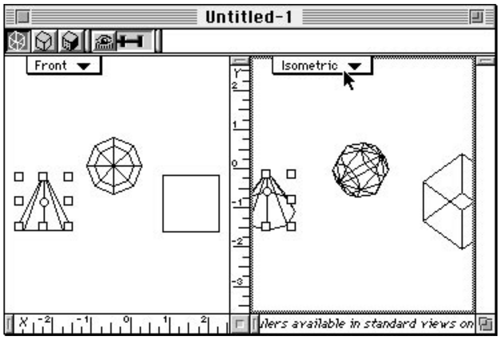


12. **In the Top view, click anywhere on one of the objects and drag it upwards on the screen.** Releasing the mouse button causes the object to redraw in its new position. Moving an object upward on

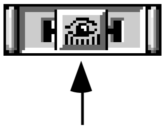
the screen in the Top view is the same as moving the object further away in the Front view.



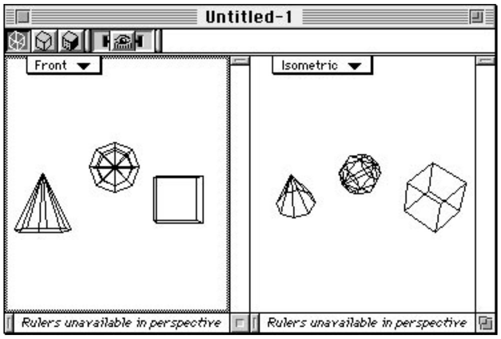
13. **Still in the Top view, reposition another object toward the bottom of the screen.** Naturally, this is the same as moving the object closer in the Front view. Notice in the Front view how the size of the repositioned objects do not change, despite the fact that they have moved closer or further away. This is because Orthographic ignores closeness to the view plane as a factor in the view.
14. **Using the View title menu, change the Top view to Isometric.** As before, the view is redrawn from the new orientation. Contrast again with the Front view. Notice how it's sometimes difficult to visually determine which way a wireframe object is facing, especially without perspective.



15. **Switch the lens control to Normal.** Notice how the objects get smaller.

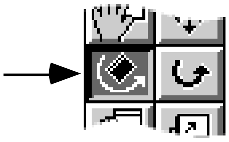


16. **From the Windows menu, choose the "Fit Views to All" command.** Notice how the objects are now bigger. They also now have vanishing point perspective. Notice also how closer objects are bigger than those further away.

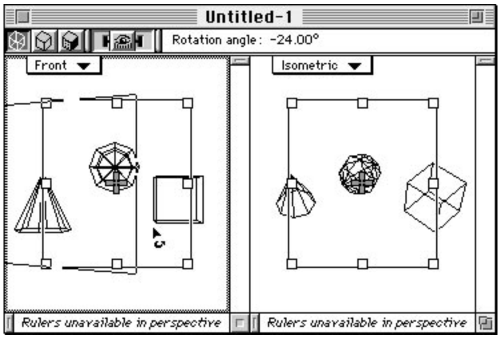

Untitled-1
Front
Isometric
Rulers unavailable in perspective
Rulers unavailable in perspective

17. **Cycle through the display methods by clicking in turn on the hidden line, quick shaded and wireframe settings.** Notice how hidden line helps you understand which way objects are facing. Notice how the quick shaded display shows lighting direction. *NOTICE HOW WIREFRAME REDRAWS QUICKEST!*

18. **Select the View Rotate tool on the Tool palette.** Vision3d places an overlay in front of the windows.



19. **In the Front view, click and drag the right center handle of the overlay 24 degrees to the left.** Watch the feedback area at the top of the window as you drag the overlay handle to see when the view is rotated 24 degrees to the left of true Front.


Untitled-1
Front
Isometric
Rulers unavailable in perspective
Rulers unavailable in perspective

Rotation angle: -24.00°

When you release the mouse button, Vision3d redraws the view from the new orientation. Notice Vision3d changed the View Title to Custom. If you rotate the view too far, choose the “Undo View Rotation” command from the Edit menu to return to the true Front view, then repeat the above steps to rotate the view to the left. Alternately, you can simply choose the Front view again from the view title.

Viewports into 3-D space

Only one window can be active at one time. In the case of Vision3d’s modeling window, which supports multiple views as split panes within the window, only one view is active within the active window at one time.

From time-to-time, you may wish to work in different views. By using views that look at a model from different directions, you often can more easily access objects that are hidden or obscured behind other objects.

To make another view active:

- **Click on the view.** If the desired view is within a window already open and visible on the desktop, simply click within its borders to make it the active view.
- **Use window-handling commands in the Windows menu.** If the window is hidden behind another window or not currently open, you may bring it forward by selecting the window from the list of open windows or opening one using the New Window command. The selected window becomes the active window.

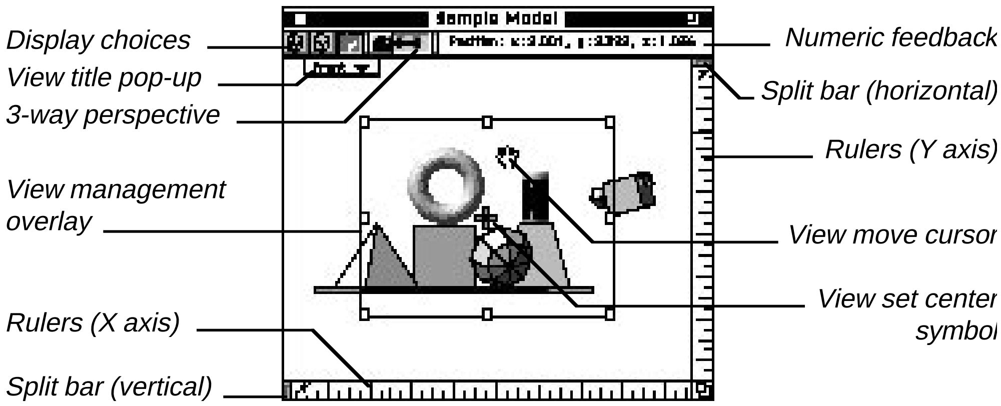
Either way, the chosen view is brought forward and its controls enabled. Controls are not visible on inactive views or windows.

Anatomy of modeling views

Vision3d windows have the standard controls common to most Macintosh applications. These controls are for manipulating the windows in the desktop environment, i.e. opening, closing, moving, sizing, etc.

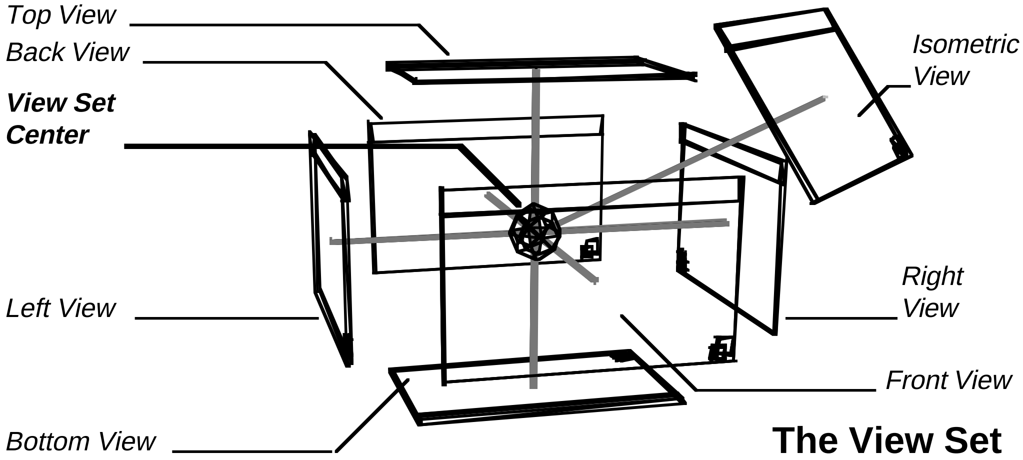
You should also become familiar with the Vision3d view controls. The controls associated with handling views in Vision3d include: display method, view angle, view title pop-up, feedback area, and view management tools’ overlays.

While most of these controls are integrated into the window frame, the view management overlays contain a special set of controls available only when one of the view management tools is chosen. Depending on which view management tool is selected, the overlays allows you to move the entire view set on any axis, move individual view planes in or out relative to the view set center, or rotate the views on any axis around the view set center.



The View Set

Each window has a complete set of views available to it. The Vision3d view set consists of seven home views: Front, Back, Top, Bottom, Left, Right and Isometric. More than one view may be present in a window by splitting the window into panes. The specific view shown in each pane is chosen through its title pop-up menu. Of course, more than one pane can use the same home position. You may also set custom view positions by rotating the view from one of the home positions.



The View Set

Each window has its own view set associated with it. Each pane within a window is assigned a “home” position in the view set in this manner and linked together through an imaginary point common to all views in the window. This common point is called the View Set Center.

View Set center

Views are always tied to their respective view set center, no matter where they are positioned in 3-D space. This minimizes the possibility of becoming lost. Each view remains aligned on the central point, providing a consistent orientation during rotation, moving in and out, or when displacing the entire view set to another location in the model space.

This center point is represented by a "plus"-shaped icon in the center of the active view whenever a view manipulation tool is selected from the Tool palette.

A view set may be displaced on any axis to allow concentration on a specific area of a model. To move a view set to another position in 3-D space, simply move or drag the view set center icon in the active view to the area of the model you are interested in. Dragging a view set center in one view moves all the views connected to that view set. When a view set is moved, each view in the set redraws to show its new respective location.

Multiple view sets

Opening a new window automatically installs a new view set center, independent of other existing view sets in the model. As view panes are added to a window, by splitting an open window, they become part of that window's view set, tied to that window's view set center.

A view set, with its attendant views, can be positioned independently anywhere in the model space. This allows you to open separate modeling windows that look at different areas of the same model.

Managing a view set

Selecting one of the view management tools automatically displays a set of view control handles as an overlay in the active view. Each view management tool has an overlay with specialized handles to accomplish the functions of the selected tool. These handles provide interactive adjustment for positioning and/or rotating views as needed around or in a model.

The view management overlays include one or more of the following components:

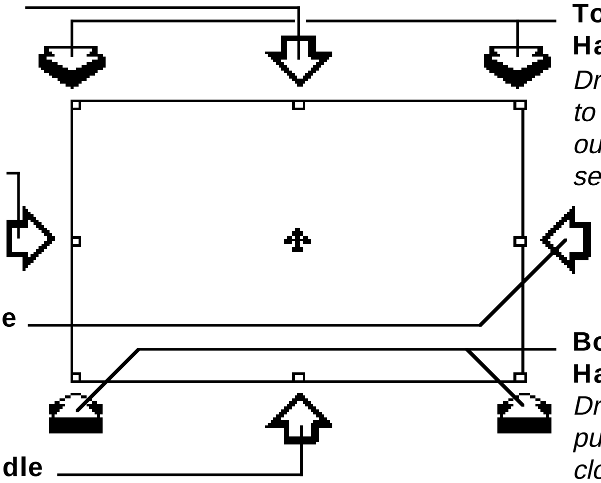
- **View set center icon** (indicates the actual location of the view set in relation to objects in the model).
- **Horizontal/vertical move handles.**
- **In/out move handles.**
- **Horizontal/vertical rotation handles.**
- **Clockwise/counter-clockwise rotation handles.**

While these overlays are just one part of the total view management system, they are considered the primary method for handling views and view sets.

View management overlays are modal in function. This means you can't edit the model while they are visible. Selecting another tool from the tool palette removes the overlay from the active view.

Moving the entire view set

With the view rotation tool selected, the view set center is represented by a "plus"-shaped symbol in the center of the active view. Whenever the view set center is moved, the view set moves with it.



Top Center Handle
Drag cursor down to move view set down vertically.

Left Center Handle
Drag cursor right to move view set right horizontally.

Right Center Handle
Drag cursor left to move view set left horizontally.

Bottom Center Handle
Drag cursor up to move view set up vertically.

Top Corner Handles
Drag cursor down to pull view plane outward from view set center.

Bottom Corner Handles
Drag cursor up to push view plane closer to view set center.

To move a view set center:

1. **Position the cursor on the symbol in one of the open views associated with the view set,**
2. **Press the mouse button and drag horizontally or vertically as desired within the view.**
3. **Release the mouse button.**

All views connected to the active view set re-draw to show their position at the new area of the model.

A view set center can be effectively moved on any axis by using views perpendicular to each other. The view set moves all of its views in spatial unison whenever it is shifted to another area of the model.

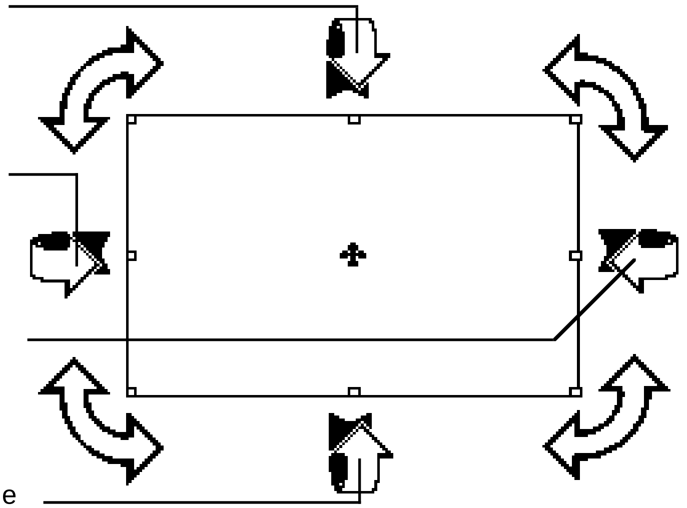
Modifier keys that apply to moving views:

A. **Shift key** — Constrains view set movement in the horizontal or vertical direction (in the active view) while dragging with the "grabber" function of the tool. Using the overlay handles automatically self-constrains the movement direction by the handles' functions.

B. **Option key** — During a move on the in/out (Z) axis, the tool moves only the active view closer to or farther from the view set center. Without the Option key, all views of the view set move in unison.

Rotating views around the model

Just as objects within the model can be rotated to present a different side in a view, views may be rotated to look at the model from different sides, positions, and angles. Rotating views from their designated home positions to custom positions (and back) does not affect viewing plane distance from the view set center. As a view is rotated around its view set center, it remains a constant distance from the center point.



Top Center Handle
Drag cursor down to rotate view plane down vertically.

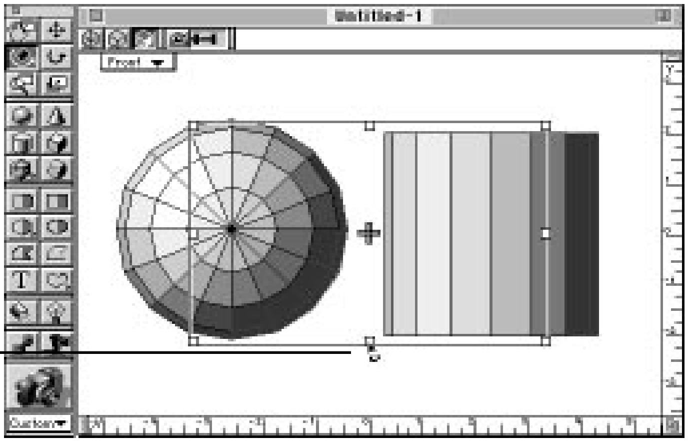
Left Center Handle
Drag cursor right to rotate view plane right horizontally.

Right Center Handle
Drag cursor left to rotate view plane left horizontally.

Bottom Center Handle
Drag cursor up to rotate view plane up vertically.

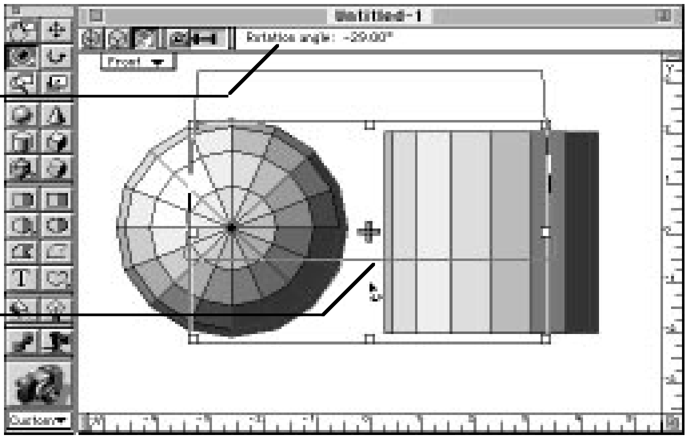
ALL Corner Handles
Drag cursor clockwise or counter-clockwise to rotate view plane around in/out axis.

When the View Rotate tool is chosen, eight view rotation handles become visible on an overlay in the active view. Dragging one of these handles rotates the view in the direction of the cursor drag. Each view can be rotated independently, but remains linked to its view set center at all times.



View rotate tool selected

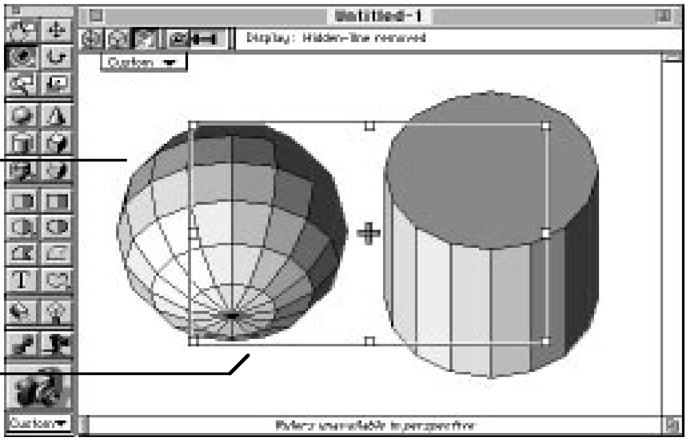
View rotate tool's cursor engages rotate up handle on view management overlay



Rotation angle: -29.00°

As cursor is dragged up, the feedback area tracks the angle of rotation

…and a view rotation marquee appears to indicate the relative position of view



Display: Hidden-line removed

Rulers unavailable in perspective

When mouse button is released, view re-draws to show new position

View management overlay remains in place until another tool is chosen

To rotate a view:

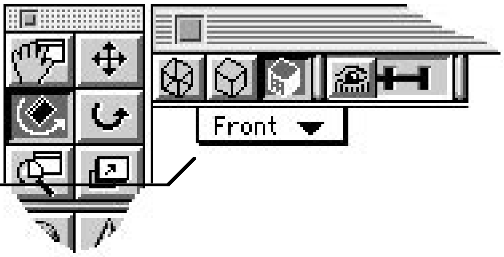
1. **Position the cursor on one of the view handles.** Use the left edge handle to rotate right; right edge handle to rotate left; bottom to rotate up; and top to rotate down.

   Rotation around the in/out axis in either a clockwise or counterclockwise direction can be accomplished by using one of the four corner handles. These rotate a view in a circular motion while maintaining the same position and plane angle relative to the model.

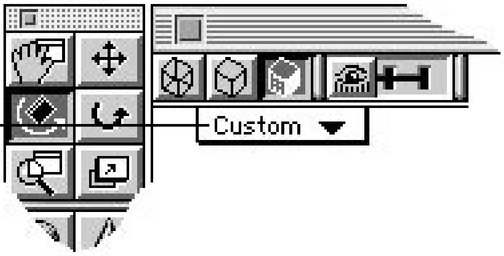
2. **Hold the mouse button down as you drag the handle.** A marquee appears and moves with the cursor to provide visual feedback. It previews the rotation operation. Numeric information is also presented in the feedback area at the top of the active window.

Releasing the mouse button executes the view rotation and the view redraws to show its new position. The title for the rotated view changes to a "Custom" designation; indicating that it is not one of the seven home positions.

As explained, a view's position is identified by its title appearing in the top left corner of the view frame. Clicking on the view title causes a pop-up menu to appear, allowing you to select a home position of that view. You may choose from seven home positions provided in a view set.



When a view is rotated, its view title changes from the starting home position



…to the "Custom" designation

If a view has been rotated using the rotation handles, the title of that view is automatically changed to "custom," since it is no longer at a designated home position. To return a rotated view to a home position again, simply select one from the pop-up menu. The view will redraw showing the designated view position.

For example, you may want to rotate the Front view to obtain a better angle on an object for a particular operation or rendering. When you are finished and want to return the view back to true front, just choose the “Front” title again. Upon release of the mouse button, the view redraws to show the model from the chosen direction.

Modifier keys that apply to rotating views:

A. Shift key — Constrains the view rotation angle to 45° increments.

B. Command key — Allows view rotation in less than 1° increments.

Moving view planes in and out

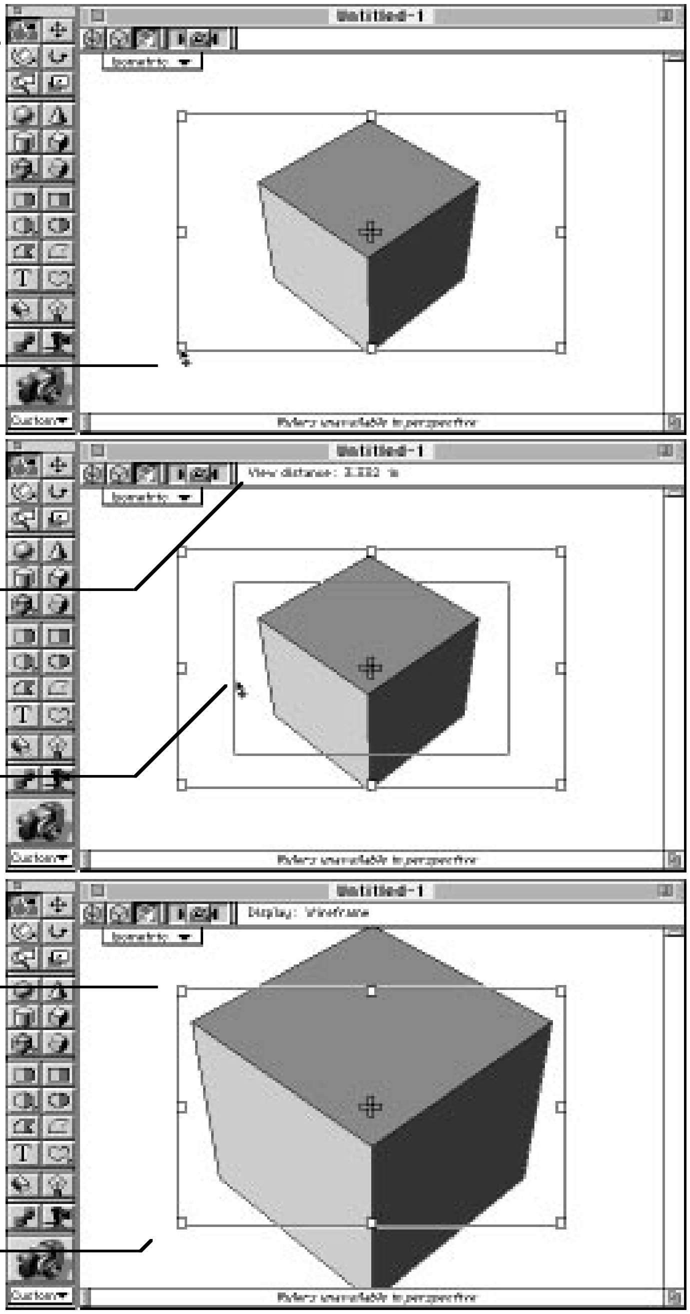
The distance from the view set center to a view plane is controlled by handles on the view move tool overlay.

In/out handles are located at the corners of the overlay. They normally move all views connected to a view set in or out simultaneously, as you drag the tool in the active view. To move only the active view and leave the other views unchanged, hold down the Option key while dragging the cursor.

Moving views closer to or farther away from objects in a model does not affect the objects, it just changes the distance of the views from their view set center and, thus, the way they display objects showing in them.

To move the view set in or out:

1. **Position the cursor on one of the in/out handles on the tool palette.** The in/out handles are located on the corners of the overlay marquee. The top handles pull the view away from the view set center. The bottom handles push the view toward the view set center. The cursor shape indicates engagement when properly positioned over a handle.
2. **Push the mouse button and drag the cursor.** Drag the cursor down from a top corner handle to move the view set outward; drag the cursor up from a bottom corner handle to move the view set inward. Visual feedback is provided in the form of a marquee in the active view as it is moved in or out. Additionally, numeric feedback at the top of the active view indicates the views’ distance from the view set center.
3. **Release the mouse button to complete the view movement.** The view is moved a distance corresponding to the distance of the cursor drag. When the view move operation is executed (via the release of the mouse button), the view redraws to show its new position.



View move tool selected

View move tool's cursor engages move in handle on view management overlay

As cursor is dragged up, the feedback area tracks view's distance from its view set center

…and a view move marquee appears to indicate the relative position of view

When mouse button is released, view redraws to show closer position

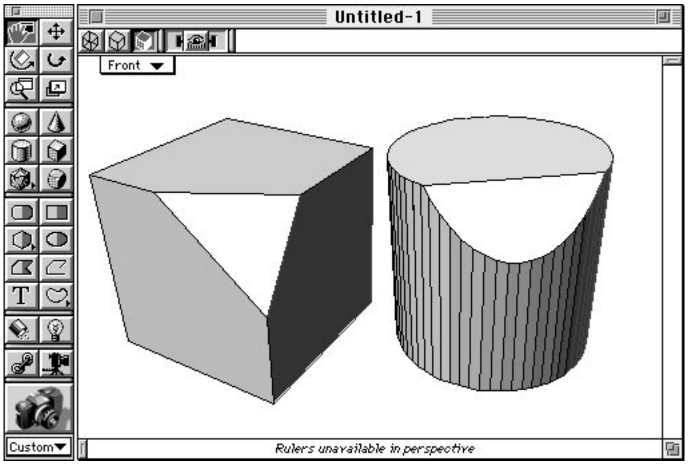
View management overlay remains in place until another tool is chosen

4. **Repeat as needed to position the view exactly.** The distance moved in a single drag of the cursor varies with the distance of the view from the view set center. The farther away it is, the farther it will move in either direction. The increment becomes smaller when the view is closer to the center point. This approach helps minimize over-corrections when attempting to precisely position a view.

At times, views will not show any apparent movement, even after moving the view plane several times for what you feel is an extended distance. The lack of visible movement is usually caused by one or more of the following conditions:

- **No perspective in the view.** Orthographic or parallel projection views do not have any depth cues and will not usually show view distance changes, except in the most extreme positions, i.e. very far away or very close.
- **The view plane is very close to the view set center.** In this case, the unit of movement is small and will take longer to move outward large distances.
- **The view plane is very far away.** In this case, the increment is greater and the views moves farther with each cursor drag, but the movement is not obvious because of the extreme distance.

Often it is more efficient to first use the “Fit Views to All” or “Fit Views to Selection” commands in the Windows menu. These commands automatically position views a comfortable distance from the designated object(s). Then, once within range, you can use the move handles to fine tune the view set’s position.



It is possible to move a view too close to objects in a model, causing the view plane to "clip" some polygons making up the object(s). While you can still render a clipped object, its surface disappears at the clipping plane. If the object is made of single-sided polygons, its interior will also disappear. In most cases when you experience clipped objects, you will want to back up slightly to make sure the entire object is visible on the front side of the viewing plane.

Sometimes an object will seem to disappear when views are moved in or out. Don't be concerned, the object is still there. It is just outside the field of the view. Sometimes it will not be visible because the view has been moved inward too far. That is, the object is actually behind the view. If desired, move outward until the object is visible again. Switching to another view that shows the model from a different direction may also allow you to see the object again.

Modifier keys that apply to moving views:

A. **Shift key —** Constrains view set movement in the horizontal or vertical direction (in the active view) while dragging with the "grabber" function of the tool. Using the overlay handles automatically self-constrains the movement direction by the handles' functions.

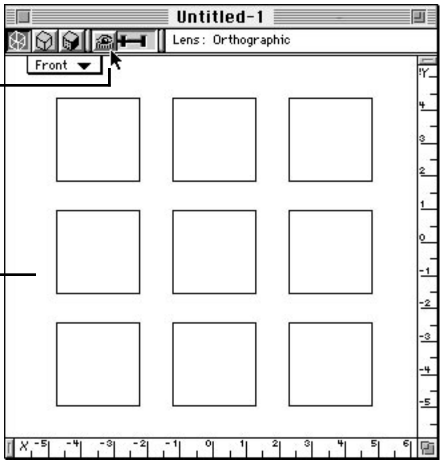
B. **Option key —** During a move on the in/out axis, moves only the active view closer to or farther from the view set center. Without the Option key, all views of the view set move in unison.

Setting perspective

Views without perspective display the model in an orthographic projection. As mentioned earlier in this chapter, you should use orthographic display during the modeling phase of your projects.

With orthographic projections, parallel lines of objects in the model appear parallel in the view and objects always appear their true size relative to one another. There is no convergence of lines or diminishing of objects with distance even though they may be great distances apart. Engineering or architectural drawings commonly employ these types of images. Orthographic projections are often used to produce front, side and top views of objects in CAD-type programs as well.

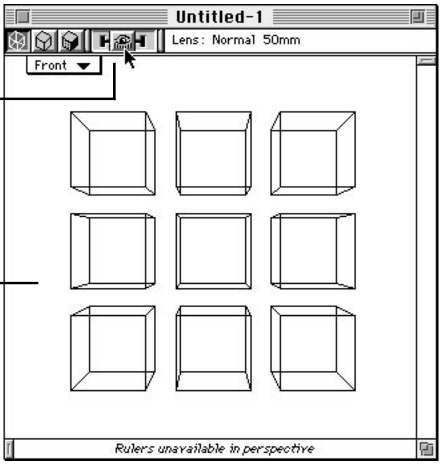
Perspective adds the element of depth to the views. With perspective, objects closer to the view plane appear larger than those farther away. Views with perspective appear more natural, since this is the way your eye or a camera forms images in the real world.



Lens: Orthographic

Perspective control set to Orthographic

View displays objects in model with parallel projection

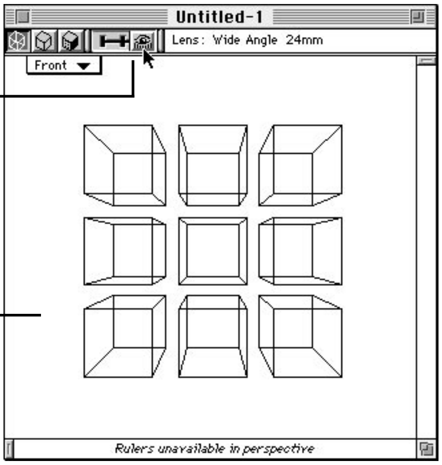


Lens: Normal 50mm

Rulers unavailable in perspective

Perspective control set to Normal

View displays objects in model with a perspective projection



Lens: Wide Angle 24mm

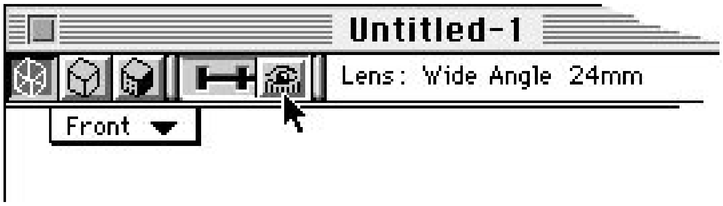
Rulers unavailable in perspective

Perspective control set to Wide Angle

View displays objects in model with increased perspective in projection

Perspective control in model windows

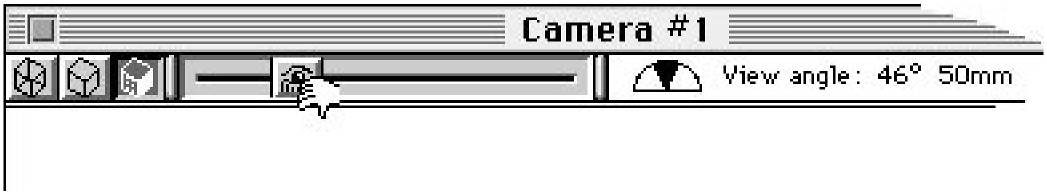
Vision3d's view angle control operates much like the focal length adjustment on a camera's telephoto lens, but with a much greater range. A three-position view angle control is located at the top of each window. The three settings are: Orthographic (no perspective), Normal, and Wide angle.



Modeling views are primarily for editing the model, and so have limited perspective adjustment. Perspective in the modeling views is useful primarily for understanding spatial relationships within a model and providing help in setting up scenes.

Perspective control in camera windows

Camera windows come from camera objects inserted in a model. They are explained later in this chapter, but are mentioned here to compare their perspective control with that of the model window's. While snapshots can be taken in modeling views, the camera windows are actually intended (and preferred) for that purpose. For that reason, the windows from camera objects offer more control over perspective.



The perspective for a camera window is continuously variable. The slider control is located horizontally at the top of the window frame. At the full right of its travel, the camera window view is at maximum telephoto, which also translates into essentially no perspective. The full left of its travel is ultra-wide angle. The view angle in that case is so extreme in perspective that objects are often distorted beyond recognition. The objects, of course, do not change…just the way they are being viewed.

In 35mm camera format terms, Vision3d's lens range is comparable to a 10mm to 50,000+ mm zoom lens…with macro focusing! This may seem a bit excessive, but the point is that seemingly small adjustments of the view angle control can sometimes create radical changes in the focal length for the window.

Consider an actual camera lens. Reducing the focal length on a telephoto lens widens the angle of view. Predictably, the scene appears farther away as the field of view widens. Conversely, increasing the focal length brings the image closer as the lens' telephoto power zoom in on its subject. If the scene is too far away or too close, you can simply compensate by physically moving the camera closer or farther from the subject. With the camera in your hands, you can walk around taking pictures of objects from any angle, at various distances, and with varying orientations. Whatever appears in the camera's view finder is focused onto the flat film surface and captured with a click of the shutter.

Applying the lens metaphor to the operation of Vision3d's perspective control: you can adjust view planes in 3-D space through various mechanisms, including moving, rotating, magnifying and focal length. Looking at a model projected on a view plane is similar to looking through the view finder of a Single Lens Reflex camera, i.e. what you see in the view finder is what you get when you take a snapshot.

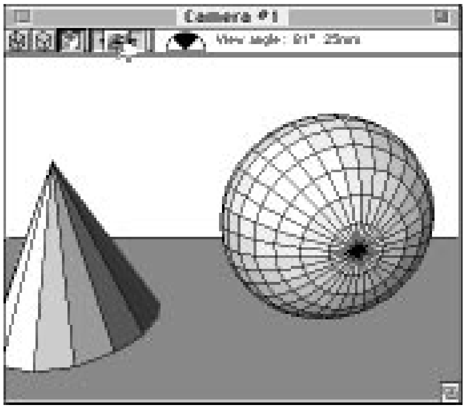
The perspective appearance of objects in a camera window depends on three factors:

- **The view angle** (focal length)
- **The magnification** (zoom factor)
- **The distance of the view** (from the objects)

Just as in the real world, the distance between the camera (view) and subject (objects) affects how much perspective actually shows in the image. If you are very far away, there is little apparent perspective. If you are close, the perspective becomes more extreme. Obviously, the most useful situation is a balance of these three factors to obtain something approaching a "normal" representation.

When a camera window is first opened in Vision3d, the view angle control is set at orthographic by default. To increase the view angle (reducing the focal length and introducing more perspective into the view):

1. **Position the cursor over the slide control on the active camera window.** Remember, only the active window has active controls. If the desired camera window is not the active window, simply click the cursor once anywhere in its boundaries. It will become the active window and you can then adjust its controls. You can also make a camera window the active window by selecting it from the "Camera Windows" sub-menu, in the Windows menu.

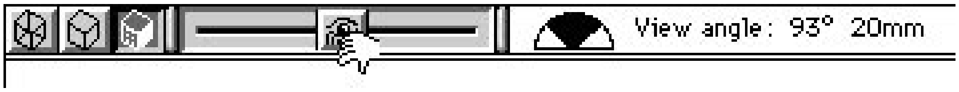


Camera #1

View angle: 81° 25mm

If there is no camera window open for a camera object inserted in the model, use the Camera Windows command to select the desired camera window from the sub-menu. This causes the window to open and become the active window. You may also double-click on the camera object to open its window.

2. **Press the mouse button and drag the control to the right.** The feedback area displays the view angle for the window as you drag the cursor. Depending on the length of the slider control, you can adjust the view angle in increments as small as one degree. Narrower windows have a shorter control extent, so the increments may be coarser.



View angle: 93° 20mm

3. **Release the mouse button to execute the perspective change.** The control shows its new setting and the view redraws to show the model with the adjusted perspective.

To obtain more perspective in a view, adjust the control further to the right. As the view angle is increased (focal length decreased), you will notice that objects in the model appear smaller. They seem farther away from the view, even though neither the objects nor the view has actually moved.

You can compensate for the wider view angle in two ways:

- **Move the view plane(s) closer to the objects.** This is done with the view move tool.
- **Magnify the image of the objects in the view.** This is done with the magnify tool.

Often a combination of the two options is used to obtain the desired view of the model.

Conversely, moving the sliding control to the right reduces the perspective and increases the telephoto factor until, at the end of its travel, the camera window shows an orthographic projection of the objects. If the objects appear too close when the control is set at the desired view angle, it may be because the camera object itself is positioned too close to the objects.

If you wish to move the camera closer or farther from the objects to compensate for the view angle/perspective, you can easily re-position the camera presenting that view of the model. This can be done two ways:

- **By using the Object Move tool and dragging the camera object away from the scene in one of the modeling windows.** The view presented in the camera window for that object will redraw showing the new position of the moved camera object.
- **By using the View Move tool in the camera window.** The overlay that appears in the camera window allows you to move the view plane just like you can in the modeling windows. But this time, the camera object responds by moving its actual location in 3-D space in concert with the view move tool's operations.

Magnifying objects in views

Views may have a scale factor applied to the images they display. This is often convenient when you wish to "zoom in" on a particular area of the model for more detail. You should not confuse the magnify tool's ability to make objects appear larger or smaller in a view with the view move tool's control operation of moving the view plane closer or farther from the objects.

Nor should you confuse the larger or smaller size of the objects in the view with changing their actual size. Scale in a view is different from the actual scale of objects. The magnify process does not alter the model in any way, nor does it change the distance between the view planes and the model. It only magnifies the image that is displayed in a view.

The magnify tool is located on the tool palette. It works in both the modeling windows and the camera windows. When this tool is selected, the cursor changes to a magnifying glass shape whenever it is within the borders of the active view.

When to use the View Magnify tool

As the view angle is adjusted for a particular window, the objects may appear larger or smaller than desired. As discussed, one of the ways you can compensate is by moving the views in or out. Sometimes it may be more desirable to keep the views positioned where they are and simply adjust the scale factor of a particular view.

There is no limit to the magnification or reduction available. However, if you try to magnify or reduce beyond the mathematical precision of the software, an alert is displayed announcing that you cannot scale the view any further that direction. It should not be necessary to ever zoom in or out that far, but if that occurs, you should reduce the magnification ratio until the views work appropriately again, then move the view plane closer to or farther from the object(s).

NOTE: *In some extreme magnification cases, a view may display inaccurate representations of the objects even before the limit is reached, because the math precision is compromised in some way. Be assured that the data in the model is still intact, and will render correctly once the view is adjusted back within acceptable magnification limits.*

How the View Magnify tool works

The magnify tool works three ways:

- **Single click** (fixed zoom factor)
- **Dragging** (variable zoom factor, with view set center shift)
- **Shift-Dragging** (variable zoom factor, with stationary view set center)

There are situations when one method is more useful than the other. It is convenient to have multiple magnification methods available.

Clicking once in the active view magnifies or reduces the view by a factor of two. When magnifying, the image will zoom in on the point where the tool was clicked in the view. This also changes the location of the corresponding view set center to align with that coordinate in the model space. When reducing, the view set center does not change position.

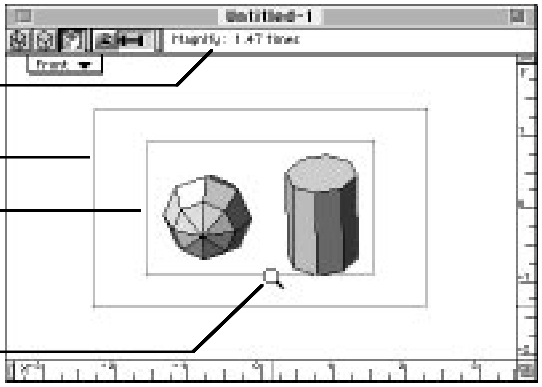
A plus or minus symbol is visible in the center of the cursor to indicate whether the single click/zoom will enlarge or reduce. Without the Option key, the tool will magnify an image, signified by a plus symbol in the cursor shape. Holding down the Option key changes the symbol to a minus. With the minus symbol showing, a single click of the tool reduces the view's image.

Dragging the tool in the active view performs a variable enlargement or reduction on the view. The view set center will likewise shift to the point where the magnify tool was clicked in the active view, unless constrained by the shift key.

When the Shift key is held down, the view set center remains fixed, but the enlarging or reducing is still variable. With this method, if the object(s) in which you are interested are slightly off center, you may have re-position the view set center to keep them in the view as you zoom in.

To use the magnification tool by the constrained variable drag method:

1. **Hold down the Shift key.** This locks the view set center at its current position.
2. **Position the cursor anywhere in the active view.** The direction you drag determines whether the view is to be enlarged or reduced.
3. **Hold the mouse button down and drag the cursor.** Move the cursor toward the center to enlarge; away to reduce. As the cursor is moved, a marquee shrinks or expands to provide visual feedback.



Feedback area tracks magnification ratio

Starting marquee

Drag marquee

With Shift key held down, the cursor doesn't have a "+" or "-" symbol in icon

4. **Release the mouse button.** The screen redraws to show the magnified view.

With the Shift drag method, if the cursor is placed at or near the edge of the view and moved toward its center, the marquee rectangle indicates the area of the view that will be enlarged. In other situations, the degree of magnification corresponds to the difference in size between the beginning and ending marquee rectangles.

Modifier keys that apply to magnifying views:

A. **Option key** — Using the single-click method of magnification, holding the Option key down reduces the image.
B. **Shift key** — Constrains the zoom operation by locking the view set in position. The magnification or reduction is aligned on the view

set center. Without the Shift key, the view set center moves to the point in the view where the mouse button was clicked to the zoom in or out.

Window Types

There are five types of document windows in Vision3d:

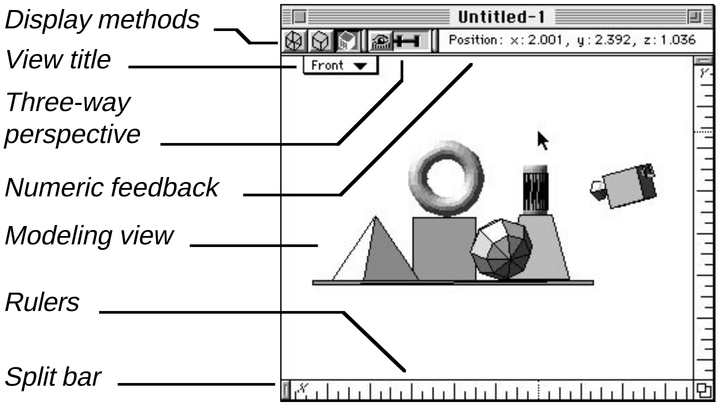
- **Modeling**
- **Camera**
- **Rendering** (also called a snapshot window)
- **Image**
- **Animation**

As discussed at the introduction of this section, the window frame for each of these window types contains the standard Macintosh interface elements. Those standard elements with which you should already be familiar include the title bar, size box, close box, zoom box, and scroll bars.

The structural components inside the windows are presented here. Mention is also made when the frame of the window types vary from the standard elements common to all windows on the Macintosh desktop.

Modeling window and its controls

Modeling windows are typically the most-often used windows in Vision3d. A modeling window contains one or more views into a 3-D model's space.



More than one model may be open at one time with its windows looking into its modeling space. At least one window must be open on the desktop for a model to remain open in memory. Closing the last window of a model closes the model as well.

A model space is an unlimited area in which you draw objects and assemble scenes. More than one window may be opened through which you may view and work in a model. Multiple modeling windows are usually oriented to view the model from different directions, although more than one view can use the same position in space.

The size of a modeling window is completely adjustable. Larger modeling windows may be split to contain multiple views. Each window has a view set center that ties all the views in that window together. Moving the view set center moves all the views in unison, but individual views may be moved around the view set center, as well as closer to or farther from it. A view's magnification scale may also be adjusted.

View move, rotate and scale operations are performed with the three view management tools located on the Tool palette.

Opening and closing windows and views

Choosing the New command from the File menu opens a modeling window for the new model. Choosing the Open command from the File menu opens those modeling windows for the Vision3d model file that were open at the time it was saved.

Additional modeling windows for a model that is already open may be opened through the New Window or Open Window commands in the Windows menu. A window may be closed by simply clicking on its close box in the upper left corner of the window frame or by using the Close command in the File menu.

The New Window command opens a completely new window whose configuration is exactly like the currently active window. The Open Window command allows previously-saved windows to be reopened. Once open, the window may be altered as needed for size, number of views, view positions, etc.

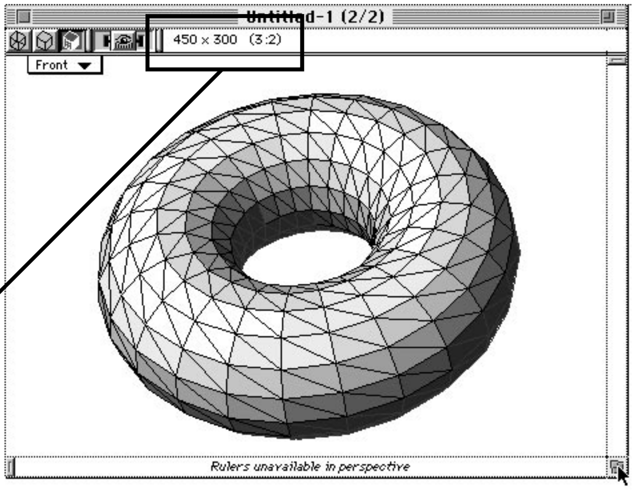
Windows may be opened or closed one at a time, as needed for the editing task at hand. You may also close a model and all its windows with one click by holding down the Option key while clicking on a window's close box or using the Close window command.

Sizing windows

A model window may be sized in two ways. Clicking on the Zoom box in the upper right corner expands the window frame to fill the available space in the screen on which it resides. Clicking on the zoom box again returns it back to the original size. Clicking and dragging on the Resize box at the lower right corner of the window frame allows you to manually set the size and proportions of the window.

In Vision3d, as you drag the window frame of either a model window or a camera window, a numeric readout appears in the feedback area at the top of that window. This readout gives you the size in pixels of the active view within the window frame.

Sizing Model and Camera windows



450 x 300 (3:2)

Rulers unavailable in perspective

When dragging the Resize box on a model or camera window, the size of the view inside the window is displayed in the feedback area at the top of the window

The numbers are only visible during the resizing operation

Additionally, if the height and width of the view happen to be sized so that a whole number ratio exists between the two numbers, that is displayed as well. The numbers are visible only while the resize operation is taking place. When the mouse button is released, the numbers disappear.

View controls

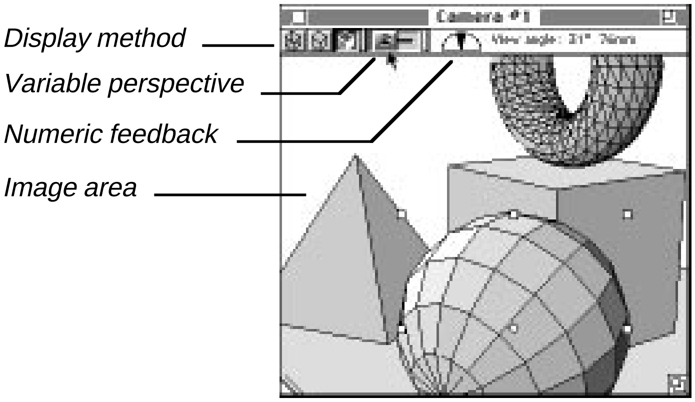
Modeling windows include some specialized controls to adjust the way objects are displayed and what type of information is presented. All of the view controls can be given a default setting for new models and/or new windows through the Preferences and Set Rulers commands, both in the Edit menu. The options for the controls are:

- **Display Method:** Choose from wireframe, hidden line-removed wireframe, or quick shaded. The display method applies to all views within the window.
- **Lens Setting:** Choose from orthographic, normal perspective, and wide-angle perspective. The lens setting applies to all views within the window.
- **Feedback Area:** No choice is available. Each tool provides relevant information during editing operations in this area of the window. Use this information to more accurately control the tool operations.

- **Rulers:** Choose the unit of measure and the major and minor graduations that appear on the rulers.
- **Split bars:** Add or delete panes in the window. To add panes, drag a split bar to the size of the pane desired or double-click on a split bar to split an existing pane in half. To delete a pane, drag its split bar to the window's edge or hold down the Option key while single-clicking on the split bar.
- **View title:** Use the pull-down menu in the view(s) to choose the view direction you wish displayed. You can choose from seven "home" positions. If a view has been customized by rotating the view away from its home position, you will see the word "custom" appear in the view title. To return to a home position, simply click on the view title and select a home position again.
- **Move and Rotate overlay:** The overlay appears when one of the view management tools is selected on the Tool palette. Using the View Move tool, you can use the overlay to move the view set in 3-D space. Using the View Rotate tool, you can use the overlay to rotate views around the view set center.

Camera window and its controls

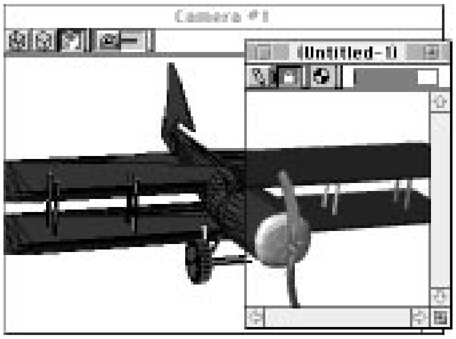
Camera windows are similar in function to modeling windows, but are linked to camera objects inserted in the model. Camera windows cannot be split. They contain only one view per window. Camera windows are independent of any other sets of views found in modeling windows, so they have more flexibility in positioning and orientation.



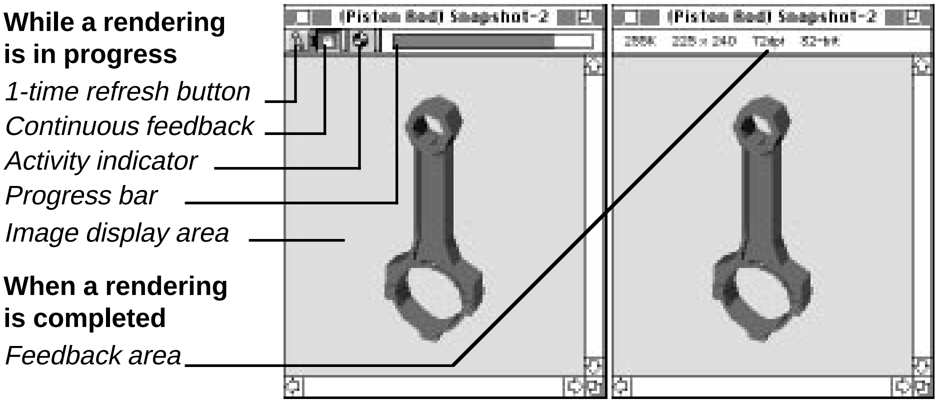
Camera windows have a continuously-variable focal length control in their window controls instead of the three-position one found on the modeling windows. Each camera window is attached to a specific camera object inserted in the model. The proximity of the camera object to other objects in the model determines what objects are seen in the window and their apparent perspective.

Rendering window and its controls

Images can be rendered of the assembled model using the camera tool from the Tool palette. Vision3d's image rendering capability includes many options and involves extensive calculations. Simply, renderings can be thought of as taking photographs of a model. A snapshot is made from the active view. Camera windows are usually preferred for setting up renderings because of their positioning flexibility, variable perspective control, and range of adjustments via the Adjust Object command.



When a rendering is initiated, Vision3d opens a separate rendering window over the view selected. Think of these rendering windows as frames of film in the process of developing. You may watch the image develop, if you wish, by turning on the preview option. Turning off the preview helps Vision3d render quicker.



A rendering window contains some specific controls that are unique to this type of window. These controls allow you to observe the rendering process as it occurs. Rendering windows serve as on-screen monitors during an image's creation. The main area of the window usually corresponds to the size of the rendered image. If the image size being rendered is larger than the screen, the rendering window controls will also include scroll bars and a re-size button to allow repositioning any part of the currently rendering image within the window.

As with any desktop window, a rendering window can be moved around the screen or to the background, even while the rendering continues. To terminate the rendering, simply click on the window's close box.

Refresh button

The rendering window display may be updated as often as desired by clicking the refresh button. The refresh button is at the far left of the feedback area, at the top of the window frame. Each time the button is pushed, the image is re-drawn once to the rendering window in its current state. You may choose whether the colors of the image will be dithered or not by setting the Dither check item on or off in the Edit menu. It toggles on and off each time the command is selected.

The effect of this command is noticeable only on monitors set to color depths less than full color (24-bit). If the Dither effect is enabled, images are drawn with the help of Color QuickDraw, which halftones the image using the available system palette

Continuous rendering feedback

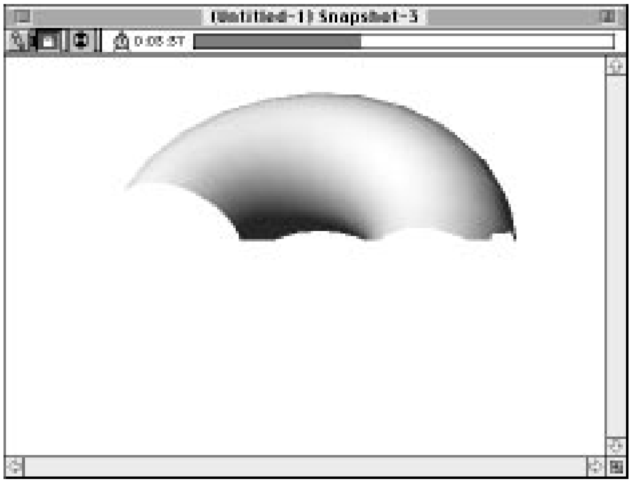
An image may be continuously updated to the window as it is being rendered by clicking on the “lock” button, located next to the refresh button. When engaged, the screen display is updated in real time as the rendering proceeds. While desirable at times, this significantly slows the rendering speed.

Activity indicator

Located to the right of the refresh once and refresh lock controls in the rendering window feedback area is a “beach ball”-style activity indicator. The beach ball rotates as the image renders. Rotation speed can also be an indication of rendering speed.

Estimated time to render

A time estimate is continuously displayed and updated in the feedback area of the window. The time is displayed in minutes and seconds or in hours and minutes, depending on the interval of time.



Obviously, the estimate is just that — an estimate. The only way it could be more accurate is to render the image to decide how long it will take to render, then render it again for the image. The estimate approach is intended to be advisory, and typically is longer than the actual time.

The estimate is updated periodically as the rendering proceeds and becomes more accurate as it gets closer to completion. The estimate calculation attempts to include all factors that impact the rendering time: 1) rendering method, 2) image size and resolution, 3) number of objects and light sources in model, 4) number and size of texture maps in model, 5) CPU speed, 6) whether Vision3d is running in the foreground or background, and 7) if in background, how much foreground activity is occurring.

Progress bar

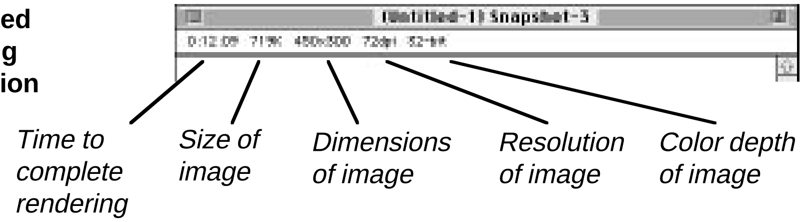
The last control is a progress bar, also located in the feedback area at the top of every rendering window. The bar moves horizontally (to the right) as the imaging proceeds within that window. Vision3d makes a "best estimate" of the amount of work completed at any point during the rendering process, however some rendering processes use recursion and may communicate an inaccurate completion status.

When the bar reaches the right edge of the window and the beach ball stops turning, the rendering is complete. You may also look at the Snapshot Windows sub-menu to see the status of all currently open renderings.

"It's done!"

For your convenience, the chosen system beep sounds when a rendering is completed. Upon completion, the rendering controls are removed from the feedback area and replaced with valuable information about the rendered image's time to render, file size, dimensions, resolution, and color depth.

Completed rendering information



(Untitled-1) Snapshot-3

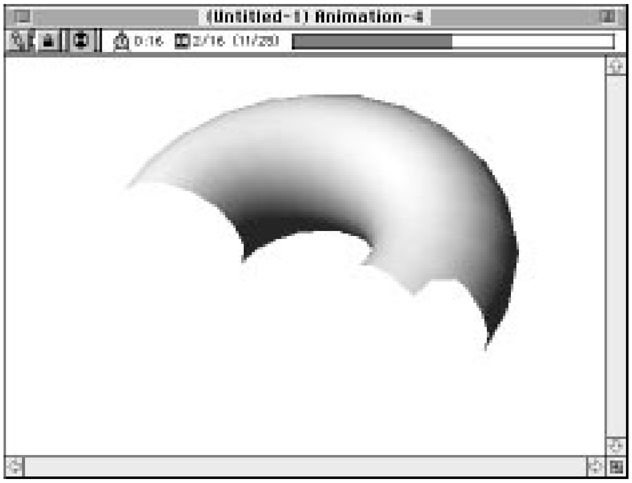
0:12:09 719K 450x300 72dpi 32-bit

If the completed rendering was started and completed in a single session from a "normal" snapshot, the software waits for you to decide what to do with the finished image.

The image will save automatically if the rendering was queued in a batch-type situation (even just a batch of one). This is set up by either choosing the "Render Later" option in the Render dialog or by using the Suspend Rendering command on a rendering that has already started, then restarting the rendering to allow it to finish.

Animations

Rendering animations add one more element to the rendering window controls while it is in progress. In addition to the time estimate (for the total animation), a combination of numbers appear between the time estimate and the progress bar. These numbers indicate the current frame being rendered and the total frames to render.



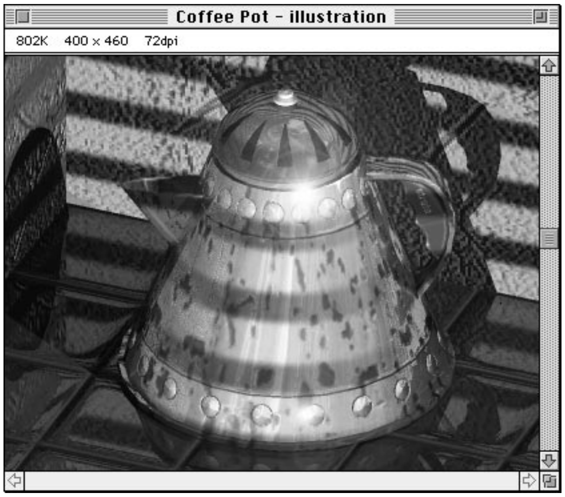
(Untitled-1) Animation-4

0:16 2/16 (11/20)

Additionally, if the animation being rendered is actually a sub-set of the total frames scripted in an animation sequence, two set of numbers are displayed. The first set is the current frame and total frames to be rendered during the current project; the second set (in parenthesis) show where the frames being rendered fit in the larger animation sequence.

Image (viewing) window

Image windows are used to display pixel or draw-type images on the screen. The data contained in an image window may be from any source, Vision3d renderings or otherwise, and in any format supported by an import/export extension. PICT is the native Macintosh format, and thus is directly supported without the need for extensions.



Upon opening an image window, the feedback area above the image contains relevant information about its size in memory, dimension in pixels, dots-per-inch resolution, and color depth. Draw-type PICT images do not display the resolution, since draw elements in the image are resolution independent.

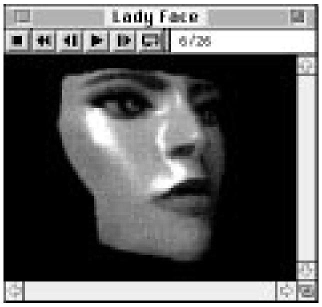
You may save an image from one of these windows in any format supported by the installed extensions. Additionally, you may print the image to any Chooser-selectable output device. As many image windows may be open at one time as desired, depending on the available memory.

Animation (playback) window and its controls

Animation windows are available for the purpose of playing animation sequences that have been saved to multiple-image files. The animation sequence may be from any source, Vision3d or otherwise, and in any format supported by an import/export extension.

Both QuickTime and PICS formats are supported by Vision3d when a set of frames is rendered as part of an animation sequence. The software automatically assembles and saves the file as the frames are rendered. The total number of frames to be rendered and the starting frame is set through the Render command in the Rendering menu.

The PICS format is commonly used for animation data exchange between applications. The QuickTime format is available only on Macintoshes running System 6.0.7 or later; System 7.0 or later is recommended. A QuickTime extension is also required in the System Folder (not in Vision3d's Extensions folder) to support this format.



The method used to play the animation depends on the format of the file being displayed. In the case of a PICS files, the individual images making up an animation sequence are normally spooled one image at a time from hard disk.

However, Vision3d has a provision to load the frames of a PICS animation sequence into memory so they may be played in the animation window more smoothly. This option is enabled with the Playback from Memory command in the Animation menu. Its operation is discussed in the Animation chapter under that command's reference.

QuickTime movie files are intended to play from the hard disk without the need to load them all into memory. The maximum frame rate depends on the speed of data transfer from the hard disk, as well as the decompression rate. The QuickTime format can contain audio information, which Vision3d also supports.

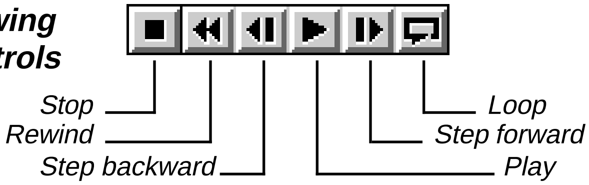
NOTE: *Vision3d does not provide animation "editing" tools, beyond applying post-rendering filters that may be installed as extensions. Vision3d is intended as a 3-D modeling and rendering solution. It provides high-quality "raw footage" usable in other applications whose rea-*

son for being is animation editing. The extent of QuickTime support provided in Vision3d is discussed in the Animation chapter.

Playback controls

As with other windows in Vision3d, the animation window contains a set of controls that allows it to perform a specific function. Six buttons are located at the top of the animation window. These buttons are very similar in function to the controls on a home VCR machine.

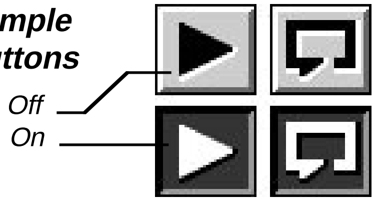
Viewing controls



If the "Loop" button is on, the animation sequence repeats continuously until you click on any of the other buttons. The sequence in which the animation sequence is looped depends on the status of the Reverse on Loop command in the Animation menu.

You should refer to the explanation of the Reverse on Loop command's operation under the Animation chapter, found later in this manual.

Sample Buttons



If the Loop button is off, the animation is played through once and stops. To view it again, click the Rewind button first, then the Play button.

Full-screen playback

Whenever an animation playback window is the active window, two commands become active in the Animation menu: Full-screen Playback and Background color. Do not confuse this Background Color command with the background color field in the Preferences dialog box. The Preferences setting applies to the background color of the modeling windows. This color command sets the color behind an animation being played in full-screen mode.

The Full-screen Playback command toggles on or off each time it is selected. When enabled, the animation frame contained in the playback window is centered on the screen and the rest of the monitor is temporarily filled with the chosen background color.

During full-screen playback, the Menu bar is temporarily hidden. It is still active, however, and its menus can still be accessed as usual by clicking in that area of the screen.

Controlling animation playback through the keyboard

In addition to the playback buttons provided on the animation window, Vision3d allows you to control animations through the cursor and Return keys. These keys operate as follows:

Return and up-arrow cursor	Start or stop Playback (each time pressed)
Down-arrow cursor	Rewind to frame 1 (beginning)
Left-arrow cursor	Move backward 1 frame
Right-arrow cursor	Move forward 1 frame

Additionally, you can click directly on the view with the mouse cursor to alternately start and stop playback.

These keyboard and mouse click equivalents can be a convenient alternative when the button controls on the playback window are hidden during full-screen playback. They save you from having to show the window frame each time you wish to start, stop, or rewind an animation.

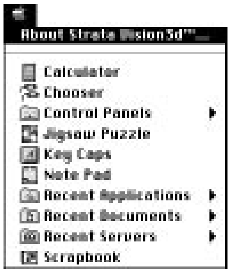
Commands used for General Operation

The following commands are considered pertinent to the general operation of the software. A detailed explanation is included.

Keyboard shortcuts are shown on those commands that have been given one in the standard configuration. You may change or add keystroke equivalents to any command through the Customize Menus command. This command is explained later in this section of the chapter.

About Vision3d command (Apple menu)

Opens an information window for Vision3d. The window identifies the version of Vision3d in use and the copyright notice for Strata Inc. More important, it displays the amount of memory available to Vision3d and how much is being used for objects and images. In System 6 MultiFinder or System 7, this window monitors only the amount available within the allocated partition, not all memory available through the System.



The "About Vision3d" window also displays the number of objects currently in the model space. This can be important information in determining the amount of RAM that will be required to be able to successfully render images of your model.

If this dialog is too big to keep open on your monitor, but you want to continue observing the space available in the image and object buffers...

...Click on the zoom box in the move bar

It will reduce the dialog from this size

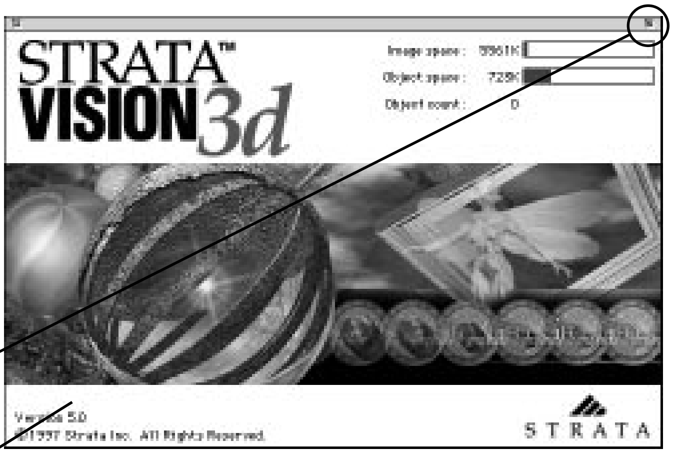


Image space: 5561K
Object space: 723K
Object count: 0

Version 5.0
©1997 Strata Inc. All Rights Reserved.

... to this smaller size

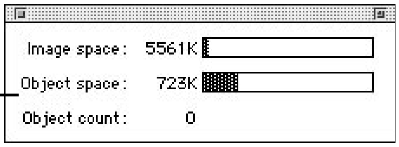


Image space: 5561K
Object space: 723K
Object count: 0

You can keep this dialog open on your screen to monitor Vision3d's operation without taking up a lot of space

Note that instances of shapes only count as one object even though the shape itself may be of complex construction containing many objects or other shapes. Some renderers — specifically the Raytracer and Raydiosity algorithms — require that these shapes be broken down into their individual components during rendering. Thus, depending on the model structure, the object count given in this window may not represent the actual load imposed on the renderer.

If you wish to keep this information visible while working in the application, but do not want the entire "About Vision3d" dialog box taking up valuable monitor real estate, click on the zoom box to reduce the window so it shows just the essential information. It may be dragged to some out of the way spot, where you can still see it without it getting in the way.

This window is not modal, i.e. — it may be open while other operations are performed in the active window. It can be used to monitor the amount of free memory from moment to moment. Note that this window is designed to stay in front of all model windows.

Close command (File menu, ⌘W)

The Close command closes the active (front-most) window. Choosing Close has the same effect as clicking the close box in the upper left corner of the active window.

Non-active windows are not affected by this command, except to bring another model window or rendering window forward upon closing the active window.

Closing model windows

If the active window is the only window open for a model, the model is also closed with this command. Holding the Option key down while using the Close command (or while clicking the close box of the active window) closes all windows associated with the active document and, by extension, closes the document.

When closing a model that has never been saved, Vision3d asks if the model should be saved. Clicking “yes” displays the Save As dialog box, with controls to name the model and select the disk volume or folder in which to save it. Clicking “No” discards the model. Clicking “cancel” returns to the model without closing it.

When closing a previously-saved model with changes, Vision3d asks if the changes should be saved. Clicking “yes” saves the changes made since the model was last saved; replacing the old version with the new one. Clicking “No” discards the changes made, leaving the previously-saved version intact. Clicking “cancel” returns to the model without closing.

Closing rendering windows

If the active window is a rendering window, and the rendering has been completed, only that window is affected. A rendering window has no ties to a model once the rendering is done. However, closing a model with a rendering still in progress terminates the rendering as well.

When closing an unfinished rendering, if the rendering has been in progress for longer than four minutes, Vision3d asks if you wish to suspend the rendering. If you choose "Yes," the rendering will be suspended, then the window is closed. If you choose "No," the window is closed without suspending the rendering and saving the work performed to that point. If you choose "Cancel," the program continues with the rendering.

You may save an unfinished rendering two ways:

- **As an image only** (in its current state of completion)
- **As a suspended rendering** (that can be restarted at another time for completion)

To save as an image, use the Save command in the File menu with the rendering window as the active window. A Save As dialog appears where you are asked to name the image, designate its format, and indicate where it is to be saved. The Save As dialog automatically provides a default name based on the name of the model where the snapshot was taken. The image format defaults to PICT for the save format. You may choose other formats from the pull-down menu in the dialog. The formats available depend on what import/export extensions are installed in the Extensions folder.

To suspend a rendering, use the Suspend Rendering command in the Rendering menu. A Save As dialog appears where you are asked to name the suspended rendering and indicate where it is to be saved. To restart a rendering that has been suspended, use the Open command in the File menu.

When closing a just-finished rendering that has never been saved, Vision3d asks if the image should be saved. Clicking "Yes" displays the Save As dialog box, with controls to name the rendering and select the disk volume or folder in which to save it. Clicking "No" discards the image. Clicking "Cancel" returns to the image without closing it.

When closing a rendering that has been previously saved (possibly as a partially-completed image), Vision3d asks if the changes should be saved. Clicking "Yes" saves the changes made since the image was last saved — the new version replaces the old one. Clicking "No" discards the changes made, leaving the previously-saved version intact. Clicking "Cancel" returns to the image without closing it.

Modifier keys that apply to the Close command:

A. Option key — Closes ALL modeling windows for the active model. This effectively closes the model. Without the Option key, the Close command closes only the active window; leaving other windows alone, if present.

Copy command (Edit menu, ⌘C)

The Copy command places a copy of selected items on the Clipboard. Copy does not remove anything from the document, but does replace the previous contents of the Clipboard with the copied items.

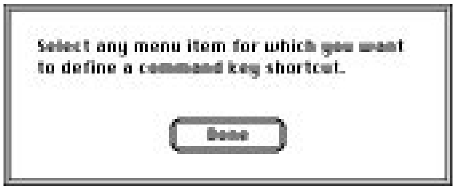
Modifier keys that apply to the Copy command:

A. **Option key —** Copies selected objects from a Vision3d model with the coordinates for those objects so they may be pasted into another model with absolute coordinates (see Paste command). Without the Option key, the objects' positions in 3-D space are not retained and will paste into another model relative to the active view.

Customize menus command (Edit menu)

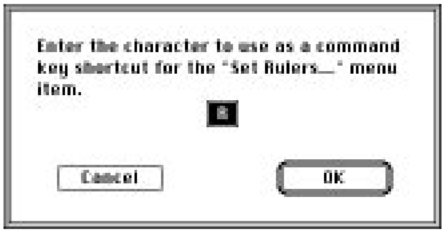
The Customize Menus command allows you to customize the keystroke equivalents used by menu commands. This command is always available.

Vision3d is shipped with a default set of keystrokes attached to its commands. You may use these, alter, or delete them as needed. You may also add keystrokes to any command that does not already have one specified.



Select any menu item for which you want to define a command key shortcut.

When this command is called, a modal dialog opens instructing you to choose a command for which the keystroke command is to be changed. All the menu commands will be active for the purpose of selection. They will not perform their normal function while this dialog is open. This dialog will remain open to allow you to change as many command keystrokes as needed at one time. When all changes have been made, click the "Done" button or push the Return key.



Enter the character to use as a command key shortcut for the "Set Rulers…" menu item.

Each time a command is chosen, a second dialog opens that allows you to enter a single character that is to be used with the Command key for executing that command from the keyboard. The chosen command is displayed in the dialog as a convenience while choosing the keystroke. The command name itself is not editable.

Normally, command keystrokes use a two-character sequence involving just the Command key and another character. On occasion, you may wish to use a three-character combination because of its special meaning to you. While you may enter only one character in the dialog, that character may use the Option key to enter it. Thus, when executing the chosen command, you then must hold down both the Command and Option keys while typing the character. Only two characters are displayed in the menu (the Command key and character combination), but the character displayed is the one obtained through the Option key, if used, to remind you of the special keystroke variation.

NOTE: *The command keystroke assignments are saved as part of the preferences file. Many other user settings are saved there as well. If the preferences file becomes corrupted, or you must replace it for any reason, all of the settings and keystroke modifications that vary from the factory settings will be lost. While this is a rare event, it is suggested you archive a copy of the preferences file — with all the settings just the way you like them — so you can replace it with minimal inconvenience.*

Cut command (Edit menu, ⌘X)

The Cut command removes selected items from the active document and stores them in the clipboard. A cut operation replaces the previous contents of the Clipboard with the new cut items.

The items that are cut may be objects (and light sources) in a model, text in a dialog field, surface maps in a texture, or any other selectable element. Depending on the context, more than one item may usually be selected and cut at a time.

Modifier keys that apply to the Cut command:

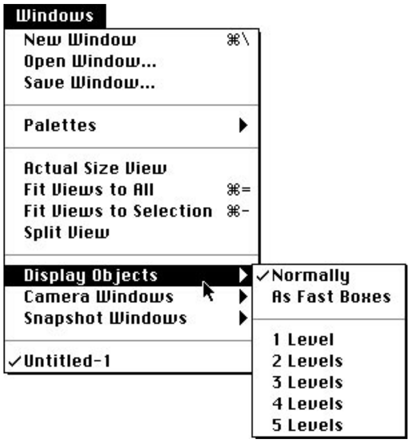
A. **Option key —** Removes the selected object(s) from a Vision3d model and retains the coordinates for those objects. Without the Option key, the object(s) position in 3-D space is not retained when pasted into another model.

Delete command (Edit menu)

The Delete command removes selected items from the active document without saving the items on the Clipboard. The Delete command functions just like the Delete key on the Macintosh keyboard.

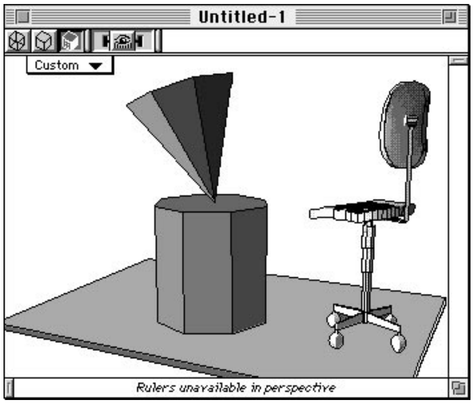
Display objects command (Windows menu)

The Display Objects command allows you to set the level of detail displayed in the modeling windows. Display Objects is available whenever a model is open. Calling the command displays a sub-menu from which the display detail is chosen. The higher the level, the less detail displayed, depending on the hierarchy of the model construction.

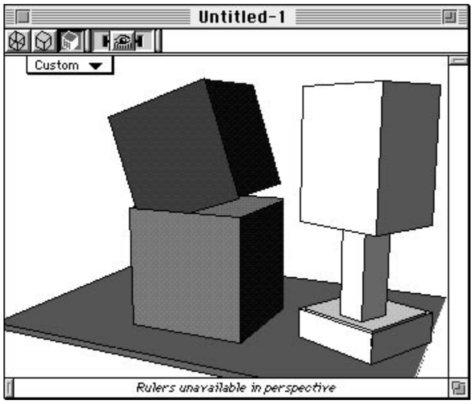


The current display detail is indicated by a check mark next to the menu item. The detail level chosen affects all models globally until changed again with this command. It is retained from session to session.

The "Normally" setting displays all objects in the model in full detail. Use this setting to see exactly how each object appears. All objects in the model are drawn polygon-by-polygon.



Normal display of objects in model



Objects displayed as Fast Boxes

If the model is so complex that the redraw times begin to take an extra long time, you can choose to display the objects as fast boxes. Fast boxes are essentially bounding boxes calculated by the objects' size and proportions. Each object's position and size can thus be indicated without the burden of faithfully redrawing all the polygons that make up each object every time the screen redraws.

You can also choose the level at which fast boxes are employed. Each level setting moves the point at which fast boxes appear further down the hierarchical structure, when viewing at the model level. Choosing "1 Level" displays standard objects drawn at the model level normally, but displays shapes inserted in the model as fast boxes. Choosing "2 Level" displays the first level shapes normally, but any second level shapes used to make the first level shape as fast boxes. The level chosen depends on how interested you are the sub-parts of instantiated shapes.

Renderings are unaffected by the Display Object command. A model using fast boxes for displaying objects in the modeling windows will still render the scene in full detail. There is no need to switch to normal display detail before taking a snapshot.

Fit Views to All command (Windows menu, ⌘=)

The "Fit Views to All" command adjusts all views in the active window to include all objects in the active document. "Fit Views to All" is available when a document is open.

The command adjusts the distance of all views in the active window from their view set center and, if necessary, repositions the view set center for a "best fit" around the model. The magnification settings in the affected views are also reset so that all objects in the model will appear in all the views. This is useful when you have lost your way.

This command provides an automatic way to move views to a larger view of the model or shape. Depending on the location of the active view set center at the time this command is executed, the affected views may be radically relocated to accomplish its function.

Only the views of the active window are affected, unless other windows are also linked to the same view set center. In that case, the views in those windows will also reset to show the entire model.

Modifier keys that apply to Fit Views to All command:

A. **Option key —** Includes the paths of all objects with animation scripting in the view. To fit everything in the view, Vision3d determines the outermost extent of the paths attached to objects or of the objects themselves (whichever is bigger) to set the scale of the view. Without the Option key, this command fits only to the extents of all the objects.

Determining the extents of all the animation paths of all the objects so they can be included in the views takes longer to calculate. Thus, this command operates noticeably slower with the Option key held down.

Note that you will not be able to use this command's keyboard equivalent while also holding down the Option key. You must use the mouse to select the command in the Windows menu in tandem with the option key to obtain this feature.

Fit Views to Selection command (Windows menu, ⌘-)

The "Fit Views to Selection" command adjusts all views in the active window to position around a selected object or set of objects. "Fit Views to Selection" is available when a selection is made, for whatever purpose.

The command adjusts the distance of all views in the active window from their view set center and, if necessary, repositions the view set center for a "best fit" around the selection. The magnification settings in the affected views are also reset so that the object(s) in the selection will appear in the views.

This command provides a way of automatically moving to a tighter view of a specific area within the model or shape. Depending on the location of the view set center at the time this command is executed, the affected views may be radically relocated to accomplish its function. This same shifting of views and viewing scale can be accomplished manually, if desired, with tools provided on the Tool pallet.

Only the views in the active window are affected, unless other windows are also linked to the same view set center. In that case, the views in those windows will also redraw to show the selection.

Modifier keys that apply to Fit Views to Selection command:

A. Option key — Includes the paths of the selected object(s) with animation scripting in the view. The operation of this command with the Option key modifier is essentially the same as with the Fit Views to All command, EXCEPT that the view extents are calculated just for the selected object(s). Without the Option key, this command fits the view(s) only to the selected objects.

Determining the extents of the animation path for the selected object so it can be included in the views takes longer to calculate. This command operates noticeably slower with the Option key held down.

As with the Fit Views to All command, you will not be able to use this command's keyboard equivalent while also holding down the Option key. You must use the mouse to select the command in the Windows menu in tandem with the option key to obtain this feature.

New command (File menu, ⌘N)

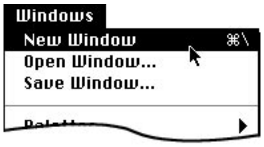
The New command creates a new Vision3d model. This command is always available, since more than one model can be open at one time, depending on the available memory. If there is not enough memory available to open a new model, an alert is displayed to that effect when the command is executed.

Choosing New automatically opens a single window using the window configuration specified in the Preferences dialog box. (The default window contains a single Front view.)

Each document is separate. It has its own model space independent of any other models currently open. **This is important to understand:** The objects you draw and new textures you make in one model are not automatically available in another model. Complete models or selected objects within models can usually be copied, imported, or merged in any combination you wish, but it must be done manually.

New Window command (Windows menu, ⌘/)

The New Window command opens another window with the same configuration as the active window. New Window is available when a document is open.

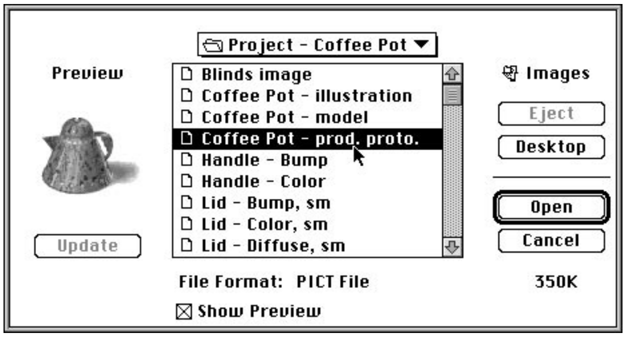


The second window also displays views of the active document. Once open, the second window may be altered as needed, i.e. — view panes within the window may be added, deleted, re-sized, changed in orientation, or changed in display scale.

Each window has an independent view set center. You can move the view set for a window, without affecting any other window of the same document. The view set center placement determines the area of the model space that shows within the views of that window

Open command (File menu, ⌘O)

The Open command allows opening an existing Vision3d model. Open is always available, since more than one model can be open at a time. If not enough memory is available to open an existing model, an alert is displayed when the command is chosen.



File Format: PICT File 350K

☒ Show Preview

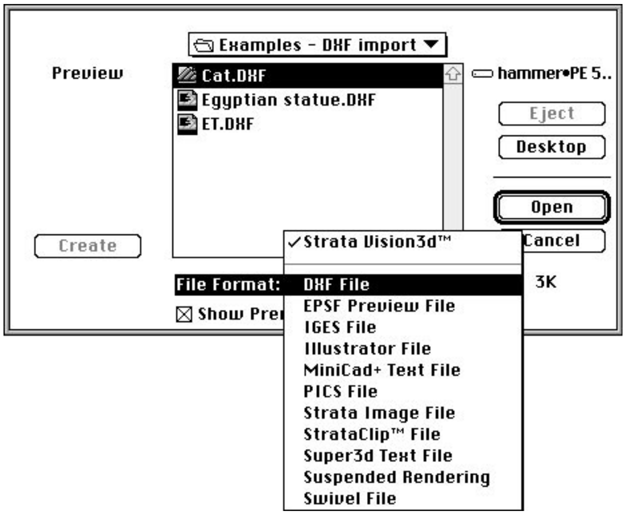
Choosing Open displays the Open directory dialog box. Depending on the import/export extensions currently installed, this dialog will recognize files in other than Vision3d's model format that may be read into Vision3d. Vision3d models, DXF, Swivel 3d, Super 3d, PICT images, etc. may be opened with this command. Of course, StrataVision 3d models are also supported.

However, note that the directory dialog box shows only those files matching the formats supported by the extensions. Formats not supported do not appear in the directory.

Open As command (File menu)

The Open As command allows opening files with non-standard format designations. With this command, all files are displayed in the directory dialog. When a target file is selected, the user must also specify its format from the available options in the "File Type" pop-up menu.

This command is especially useful when attempting to read non-standard file formats. For example, if a DXF document is transferred from an MS-DOS® system to the Macintosh, the file type of the document is sometimes designated as TEXT instead of the correct DXF. In this case, the Open command would not recognize the file accurately, but the Open As command will allow reading the file as a DXF document.

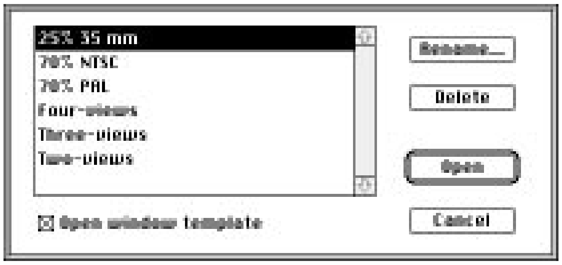


File Format: DXF File

With this command, no attempt is made by Vision3d to "guess" the file type of a selected file; the user is expected to know and manually designate the format. If the file format chosen is not correct, i.e. — the actual data does not conform to the expected format, an alert is displayed and the Open As attempt is terminated.

Open Window command (Windows menu)

The Open Window command opens a previously-saved window configuration. Open Window is available when one or more windows have been saved.

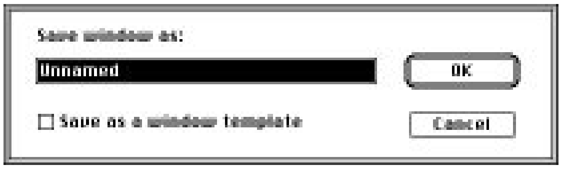


☒ Open window template

Saved window configurations are retained either exclusively with a model from which it was derived or as templates for all models to use.

A previously-saved window is opened in exactly the same location and configuration as it was saved. Once opened, the window may be altered, but must be saved again to retain any changes made to it.

If the window chosen through this command is already open, the System beep sounds and the window is brought forward, making it the active window.



Save window as:

Unnamed

OK

□ Save as a window template

Cancel

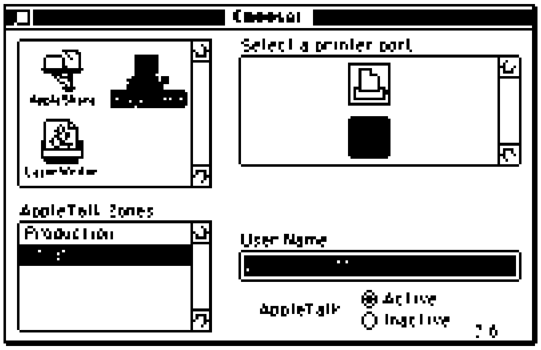
Saved modeling windows may also be renamed by clicking on the Rename button in the Open Window dialog.

Page Setup command (File menu)

The Page Setup command allows specifying printing information such as page size, image orientation and image size on the page. Page Setup is available when a document is active. The active window may contain either a model view or an image. The image does not have to have originated in Vision3d.

Choosing Page setup displays the Page Setup dialog box.

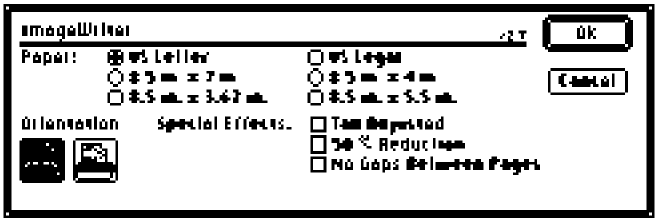
Vision3d uses the Macintosh System's Apple Print Manager to print documents. In this way, Vision3d is able to print to any Chooser-selectable device. The Chooser is a desk accessory provided by the System as a menu item in the Apple menu. It allows selection of a printer from the available print drivers installed in the System Folder.



The format of the Print dialog box displayed depends on the output hardware chosen, i.e. ImageWriter, QuickDraw LaserWriter, PostScript LaserWriter, Linotronic typesetter, Film Recorder, etc.

Page Setup for QuickDraw printers

QuickDraw Page Setup is available when a document is active and a QuickDraw-compatible output device has been selected in the Chooser. These devices include Apple-brand printers like the ImageWriter, the ImageWriter LQ, or the non-PostScript LaserWriter SC. Other companies may produce printers that are usable by Macintosh computers.



ImageWriter v2.7

Paper: ● US Letter ○ US Legal
○ 5 in x 7 in ○ 3 in x 4 in
○ 8.5 in x 3.67 in ○ 8.5 in x 5.5 in

Orientation

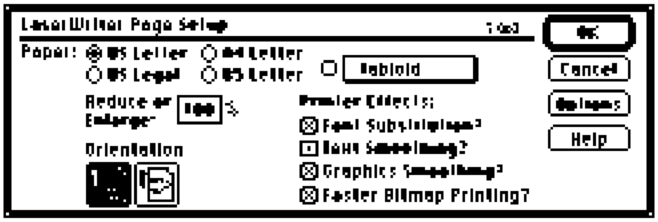
Special Effects: □ Tall Adjusted
□ 50 % Reduction
□ No Gaps Between Pages

The dialog box controls printing only, not the format or other details of a document. The QuickDraw Page Setup dialog provides options for the following settings:

- **Paper.** Specifies the physical size of the paper in the printer. QuickDraw printers cannot print to the edges of the paper, so the print area is always smaller than the physical size. For example, the print area for an ImageWriter on an 8.5″ by 11″ U.S. Letter sheet is a 7.5″ by 10″ rectangle centered on the paper.
- **Orientation.** Specifies the orientation of the printed page. The left icon represents vertical or portrait page orientation. The icon on the right represents horizontal or landscape orientation. Portrait orientation is the default setting whenever a session is begun. If the orientation is changed, it is remembered only during the current session.
- **Special Effects.** Because QuickDraw prepares output optimized to the resolution of the device, the Tall Adjusted and 50% Reduction options are not available. Check No Gaps Between Pages to eliminate the unused area at the top and bottom of each page.

Page Setup for PostScript printers

PostScript Page Setup is available when a document is active and a PostScript device has been selected in the Chooser. These devices include Apple printers such as the LaserWriter NT and NTX. Other companies also sell PostScript-equipped printers for use with Macintosh computers.



LaserWriter Page Setup

Paper: ● US Letter ○ A4 Letter ○ US Legal ○ B5 Letter ○ Tabloid

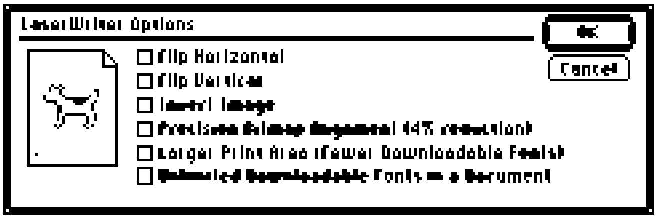
Reduce or Enlarge: 100 %

Orientation

Printer Effects:
☒ Font Substitution?
☐ Text Smoothing?
☒ Graphics Smoothing?
☒ Faster Bitmap Printing?

The dialog box controls printing only, not the format or other details of a document. The PostScript Page Setup dialog provides options for the following settings:

- **Paper.** Specifies the physical size of the paper in the printer. Depending on the printer driver in the System Folder, as many as seven different paper sizes may be available. If the paper size selected is smaller than the actual image being printed, only part of the image will print on a single sheet of paper.
- **Reduce or Enlarge.** Allows scaling of the printed image in a range of 25% to 400%, where 100% is "normal" size.
- **Orientation.** Specifies the orientation of the printed page. The left icon represents vertical or portrait page orientation. The icon on the right represents horizontal or landscape orientation. Portrait orientation is the default setting whenever a session is begun. If the orientation is changed, it is remembered only during the current session.
- **Printer Effects.** Allows certain choices supplied through the Macintosh System software, including font substitution, bit map text smoothing, graphics smoothing, and faster bit map printing.
- **Options.** Clicking on the Options button accesses any custom features available through the specific print driver being used. In the case of Apple's print driver for all LaserWriter-compatible PostScript printers, the Options button displays the LaserWriter Options dialog box.



LaserWriter Options

☐ Flip Horizontal
☐ Flip Vertical
☐ Invert Image
☐ Precision Bitmap Alignment (4% reduction)
☐ Larger Print Area (Fewer Downloadable Fonts)
☐ Unlimited Downloadable Fonts in a Document

In the Options dialog box, additional parameters may be set for the way images are printed. The exact format and fields for this dialog box may vary according to the version of the LaserWriter driver in the System Folder.

- **Help.** To display Apple's on-line information about Page Setup options, click the Help button.

Paste command (Edit menu, ⌘V)

The Paste command places items contained on the clipboard in the active document.

The items being pasted from the Clipboard must usually be in the same context as the place they are pasted, i.e. a 3-D object cannot be pasted into a text field in a dialog box.

However, black-colored text, PICT, and EPS images may be pasted into 3-D model space, which generates a polygonal extraction of the 2-D shapes. This is an extension of the functionality provided by the Import command.

Objects copied from one Vision3d model may be pasted into a second model with all data intact, i.e. — the applied texture (if any), coordinates (with Option key modifier), as well as the shapes used to construct them (if any).

Paste and the other editing commands may also be used to insert and remove surface maps in the texture editing dialogs.

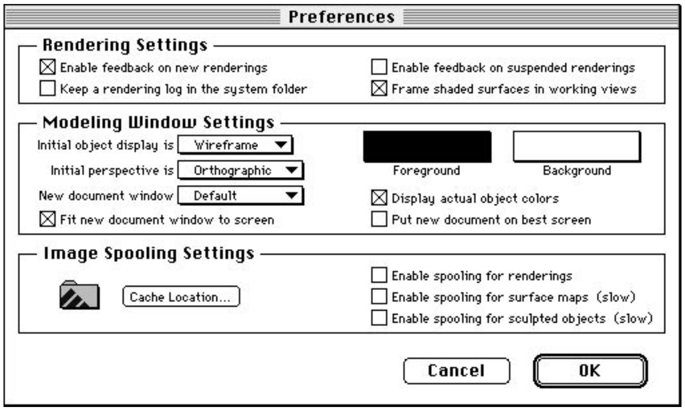
Modifier keys that apply to the Paste command:

A. **Option key —** Pastes the contents of the clipboard into the same coordinates from which it was copied. This behavior works ONLY if the item on the clipboard is a Vision3d object that was cut or copied with the Option key held down as well. Without the Option key, the object is pasted at the location of the view set center.

Preferences command (Edit menu)

The Preferences command allows the modification of settings for basic Vision3d functions. The Preferences command is always available, even without a model open.

Changes made in the preferences dialog are immediately available in the current session and permanently saved to the preferences file upon quitting Vision3d. If, for some reason, you exit Vision3d without going through the Quit command (System crash, power failure, using the Force Quit feature in System 7, etc.), the changes you make to the Preferences dialog settings, as well as other configuration changes, the preferences files will not be updated, and the changes will not appear in the next session.



Preferences

Rendering Settings

☒ Enable feedback on new renderings
□ Enable feedback on suspended renderings
□ Keep a rendering log in the system folder
☒ Frame shaded surfaces in working views

Modeling Window Settings

Initial object display is Wireframe
Initial perspective is Orthographic
New document window Default
☒ Fit new document window to screen
☒ Display actual object colors
□ Put new document on best screen

Image Spooling Settings

□ Enable spooling for renderings
□ Enable spooling for surface maps (slow)
□ Enable spooling for sculpted objects (slow)

A preferences file is provided as part of the installation process. However, the "Strata Vision3d Preferences" file must be manually moved into the System Folder before the application will run. If no preferences file can be found when Vision3d is launched, it will alert you to this deficiency then quit.

Choosing the Preferences command displays a dialog box with multiple editable fields and pull-down menus. Any changes made in the Preferences dialog take effect immediately upon clicking the OK button or pushing the Return key. These settings are global, i.e. — they apply to all models, new or existing. Clicking the "Cancel" button ignores any changes you may have made in the dialog box.

Some preferences affect the appearance of the model; others tell Vision3d where its support files are located. An explanation of the fields in the Preferences dialog box follows.

Saving defaults

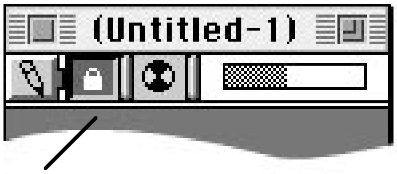
Changes made in the Preferences dialog box do not update the Vision3d Preferences file until you quit Vision3d. If the application is exited in any other manner than through the Quit command, i.e. — Type 1 error, System error, Escape Shell (while in the Debugger), Force Quit (Command-Option-ESC in System 7), this updating never occurs and your changes do not appear when Vision3d is launched the next time.

Feedback in rendering windows

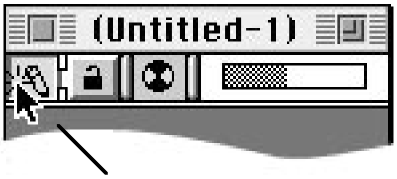
The "Rendering Settings" fields determine whether rendering windows open with the screen feedback turned on or off. This setting may be made independently for both new and suspended / restarted rendering windows.

If the feedback is enabled, you can watch an image take shape. The rendering window will show the state of a rendering as it "develops."

If the feedback is disabled, the image still renders internally, but no immediate feedback is sent to the screen. By turning off the visual feedback, a rendering can often proceed faster.



Visual feedback in the locked down or on position.



With visual feedback off, just click the re-draw button to update.

When the screen feedback is off, there may be times when you would like to see how the image is coming along. To obtain a one-time screen refresh of the image at that moment of the rendering process, click once on the button with the pencil icon. The image will be drawn and the rendering will continue on without any further screen updating until the pencil button is clicked on again.

New window arrangement

The configuration for opening new modeling windows is set through the "New Document Window" pull-down menu. The default is a single pane displaying a Front view of the model. If other configurations are available, you may select them here as the configuration for new windows.

You may define and save other configurations that will then appear in this pop-up menu. This may be done any time a model is open. Once the window is set up — including any multiple view panes desired, their sizes, and view position — select the Save Window command from the Windows menu. The dialog allows you to name the window arrangement and choose whether the window is to be global or local. Windows saved as global definitions are available in all models, and thus included in this pull -down menu in the Preferences dialog box. Windows that are not global are saved with the current model only.

Default Display method

The default display method used in new modeling windows is set with a pop-up menu for the "Initial object display" field. The choices correspond to the three display-method buttons at the top of each modeling window. These renderers are available for the modeling windows' views.

While shaded views are able to communicate extra information about the model that may not be apparent in the other display methods, you may find view redraws will be quicker in the case of complex models by using wireframe.

Default Lens perspective

The default perspective used when opening new modeling windows is set by the "Initial Perspective" field. The choices likewise correspond to the three-position perspective control found at the top of each modeling window. The three choices indicate how views in a new window can present the model. That is, how much perspective is in the display, similar to the focal length setting of a camera lens.

Orthographic views display the lines of objects in the model in parallel. There is no vanishing point, no convergence of lines with distance. Orthographic views preserve the absolute dimensions of objects, so they can be sized accurately relative to other objects or to the rulers.

Perspective displays (the Normal and Wide Angle settings) on the other hand, produce more realistic views but do not preserve actual or relative dimensions. Distant objects are projected as smaller than those closer to the viewing plane, even when they are the same size.

Foreground/background colors

The "Model Windows Colors" fields determine colors used to display the model in the modeling windows. Obviously, you should use colors that are complementary, yet with enough contrast to see the information. Changes in the foreground or background colors (or both) apply immediately to the current model and for all future models to be open. The foreground and background colors are not saved with the model.

Modeling views color depth

Objects in the modeling windows may use the simple foreground color or a representation of their actual colors. The option is set with the "Display Actual Object Colors" check box. If this item is disabled, the modeling windows display each object in the foreground color. If enabled, the applied texture is used for the color, even in wireframe. If the texture uses a map, either surface or solid, the color chosen for that object in the modeling windows is an average of the map's colors. No color variation across the map is attempted.

You should remember that objects with no texture have a default opaque matte white surface. Thus, new objects display a white representation in the windows when the actual colors option is used. If the background color is also white, the object may be invisible. To avoid potential confusion, you should pick a background color that is unlikely to be matched by object colors.

While object colors can be useful to distinguish objects from one another, especially in wireframe, screen redraws are faster when the modeling window use the foreground color instead. In the case of complex models, you may wish to temporarily turn this preference item off.

Enabling the spooling cache and location

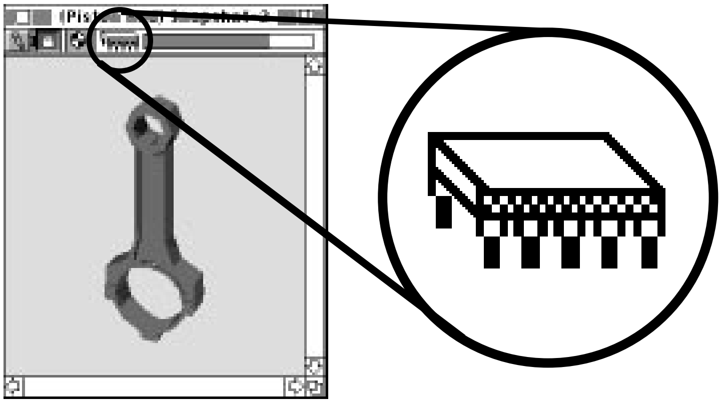
Caches are temporary hard disk storage allocations used by Vision3d to facilitate the rendering process. While several rendering processes often require the use of caches, their use usually involves a trade off between speed and memory. The compromise is to make more room available to handle images (both renderings and texture maps) that cannot fit completely in RAM.

You are given the option of using or disabling the various caches depending on your rendering priorities. The proper configuration of Vision3d's cache options can mean the difference between a rendering being able to proceed or running out of memory.

The general rule is to attempt to render without any spooling options enabled. If necessary, enable only the image spooling option first. If memory is still not sufficient, enable the spooling for surface maps. If there is not enough RAM, you can enable the spooling for sculpt objects, but this only helps if sculpt objects are in the model.

If you are consistently enduring tight memory or out of memory situations, you should consider getting more RAM for your machine. However, throwing more RAM at the problem is not the whole answer. Often there are alternative modeling methods that can make your model more efficient without adversely affecting the rendered result. These alternatives are suggested in *Chapter 3: Modeling Objects.*

The time when memory becomes most critical is when there is a rendering is in progress. In situations where memory is tight or the memory configuration is not optimal, the performance of the rendering algorithm will be less efficient.



I.C. chip appears in rendering window when the memory configuration for rendering is not optimal. Tight memory due to partition size and cache settings is the most common reason for slow rendering

You will be made aware of the problem by a small icon (representing an integrated circuit chip) that appears in the rendering window's feedback area. If you can make the icon go away through judicious adjustment of the application's memory partition and spooling options, your reward should be an increase in rendering speed.

Once a rendering is in progress, changes in the preferences and memory partition will not have an effect. You must terminate or suspend the rendering, make thememory managment changes, then try again.

Setting cache location on hard disk

The cache structure is in two parts: RAM buffers and hard disk buffers. Vision3d takes care of the RAM buffers. You control the hard-disk buffers. The first part, the "Cache Location" button, allows you to designate a location for the hard disk portion of image spooling.

A directory dialog opens when this button is clicked to point at the drive, volume and folder you wish to use. When you have opened the desired folder, so that the contents of the folder are visible in the directory portion of the dialog, you can click either on the "Select Current folder" button at the top of the dialog or on the "Here" button on the bottom right. Both buttons complete the location-selection process when clicked and close the dialog.

Spooling Renderings

This check box enables or disables caching of the rendered image on the hard disk. Segments of the image are then spooled into RAM for processing. The actual rendering process occurs in memory, but through the use of image caching , renderings too large for the available memory can still be created. Disabling this option forces Vision3d to keep the entire image in memory, which may render faster but severely limits the size of images.

Spooling surface maps

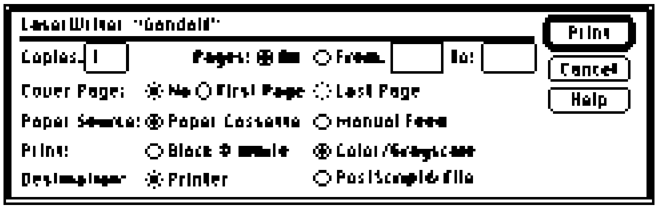
This check box enables or disables caching of surface maps. Each map used in a texture requires a buffer where the map can be "laid out" for the renderers to look at while calculating how the map should appear on the applied object(s). If enough maps are used, and all of them are in memory at the same time, there may not be enough memory left to render the image. By spooling the surface maps, i.e. — making them available to the rendering process but in manageable segments only as needed, Vision3d can handle more (and larger) surface maps without running out of memory.

Spooling Sculpted objects

This check box enables or disables caching of the sculpted object's gray-scale image from which the sculpt object is generated. The sculpted-object class converts gray-scale paint-type images to elevation data for a 3-D object. Modeling windows display the sculpt object in polygonal form strictly for manipulation in the model. During the rendering process, however, handling of the sculpt object's gray-scale representation is very similar to surface maps. The gray-scale image must be referenced much as a surface map would be, except that the image defines a geometric shape instead of surface patterns.

Print command (⌘P, File menu)

The Print command enables specifying printing options for the active document. These include the range of pages to be printed, tiling, color depth, etc. Print is available when a document is open. Choosing the Print command displays the Print dialog box.



LaserWriter "Gandalf"

Copies: 1 Pages: ● All ○ From: To:

Cover Page: ● No ○ First Page ○ Last Page

Paper Source: ● Paper Cassette ○ Manual Feed

Print: ○ Black & White ● Color/Grayscale

Destination: ● Printer ○ PostScript® File

The options provided in the dialog depend on the printer driver in the System Folder. Some options do not apply directly to Vision3d, but are presented here for understanding:

- **Copies.** Specifies the number of copies when the active document is printed. Enter a value from 1 to 999 in this field.
- **Pages.** Specifies the pages to be printed. The default is to print all pages in the document, unless values are entered in the “From” and “To” fields.

  While extra large images may require that it be tiled across multiple pages, each image document is considered to be one page long. The concept of pages in a Vision3d document is not applicable.
- **Cover Page.** Specifies whether to print a page containing the name of the document along with the time and date printed. Cover page is available only for PostScript printers. The cover page may be printed as the first page or last page of the document by choosing the appropriate option. A cover page is for identification purposes only and is not considered part of the document.
- **Paper Source.** Specifies the way to feed paper into the printer. The choices are paper cassette or manual feed. Paper cassette is the preset default. When printing to a continuous-feed printer, the paper cassette option should be selected.
- **Print.** Specifies whether to print an image in color/gray-scale or black & white. Choosing the color/gray-scale option will print color images on a color printer, and gray-scale images on a black & white printer.

  Most color printers are capable of only 8-bit, 256 color images, but can print full-color images using the System’s QuickDraw dithering routines to simulate colors not available with the printer’s limited palette. Gray-scale is likewise simulated via QuickDraw dithering patterns. Black & white restricts the output to black & white for any printer, and results in a high-contrast image.

  The color/grayscale option is available in version 6.0.1 and later of the Apple print driver. Most print devices are designed to work with the Apple print driver. Some non-Apple drivers exist, and may or may not provide the same functionality. Some devices provide System extensions that add to the standard Apple print driver rather than replace it.

  Color images from Vision3d are rendered in full color (32-bit, 16.7 million colors) or gray-scale (8-bit, 256 levels of gray). There is no option for converting to 8-bit, 256 color images in Vision3d, except through the Save/Save As commands. Additional image-editing features may be available in other programs.
- **Help.** To display Apple’s on-line information about Print options, click the Help button.

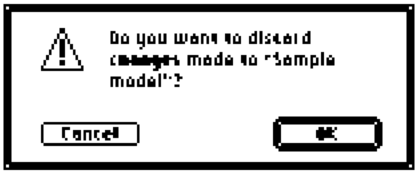
Quit command (File menu, ⌘Q)

Choose Quit to exit Vision3d and return to the Finder. Before quitting, Vision3d asks if you want to save changes to open, un-saved documents or documents with changes since they were last saved.

Any changes made to the Preferences dialog box during the session are updated to the Vision3d Preferences files at this time.

Revert to Saved command (File menu)

Choose the “Revert to Saved” command to discard the changes made to a model or image and restore the active document to the last-saved version. This command is available when a document has been previously saved, then changes are made after the last save.



Do you want to discard changes made to "Sample model"?

When executing the Revert to Saved command, Vision3d displays an alert warning of the consequences when reverting to a previous version: The current state of the document will be discarded and cannot be undone. Click “OK” to proceed with the revert operation. Click “Cancel” to return to the current active document.

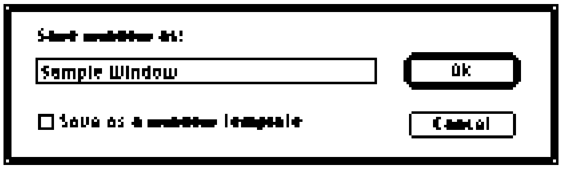
Save A Copy As command (File menu)

The “Save A Copy As” command allows saving a copy of the current document under a different name without changing the current name and file path of the document. Subsequent saves using the Save command will retain the current name and data path, not the one used with this command.

Additionally, in the case of an animation, you can use this command to change the format of the animation as it is saved under a new name. This is useful for converting a QuickTime movie to PICS format, and vice versa of course.

Save Window command (Windows menu)

The Save Window command saves the active window’s current configuration by name. This command is available whenever a document is open.



[illegible]

Sample Window

□ Save as a [illegible] template

A check box is provided in the dialog box to save the window as a template. Template window configurations are saved in the preferences file; making them available in all models.

Modeling windows in either the model work space or individual shape work spaces may be saved and retrieved by name. If the name entered to save a window is already in use, the System beep sounds and an alert appears asking if the current window configuration is to replace the one saved previously. Clicking on the “Yes” button proceeds with the save using the same name. Clicking “No” returns to the Save Window dialog to allow entering a new name.

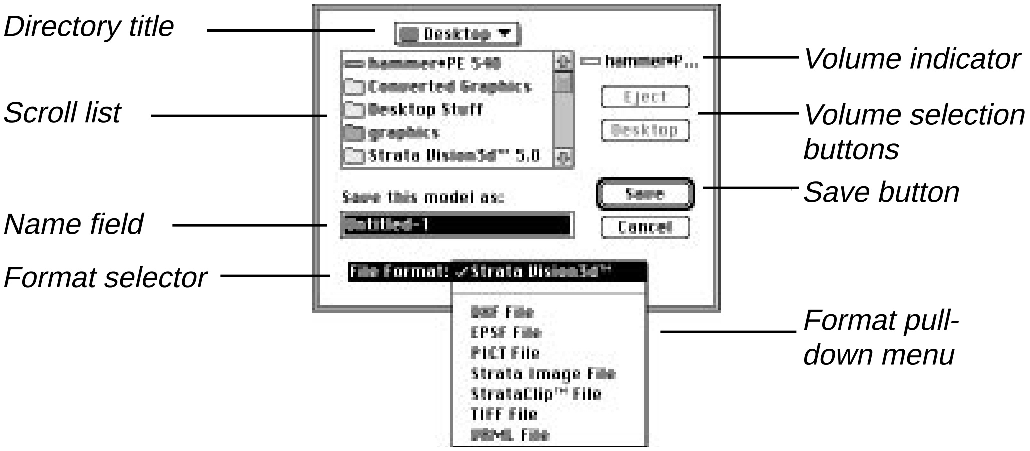
Save command (File menu, ⌘S)

The Save command saves changes made to the active document. The active document is determined by the active window and may be either a model or a rendered image. Save is available whenever changes are made to a document. The command is dimmed if no changes have been made to a document since the last save.

If the active document has not been saved, choosing Save displays the Save As dialog box, which allows naming the document and selecting the disk and folder in which it is to be saved. For previously-saved models, this command uses the established name and volume information to save the document…transparently replacing the old version with the current version. The previous version is deleted and no longer available under normal circumstances.

Save As command (File menu)

The Save As command allows saving the active document using a different name and location. The Save As command is always available, as soon as a document is open. Choosing Save As displays the Save As dialog box shown in the illustration on the following page.



Save this model as: Untitled-1

File Format: ✓Strata Vision3d™

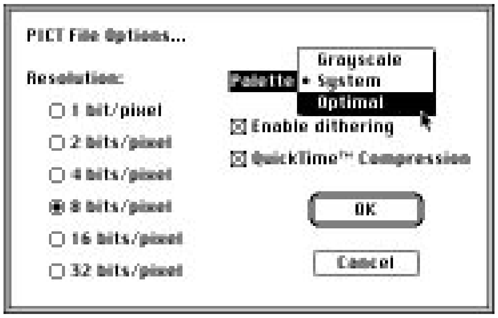
To save the active document using a new name, enter a name in the "Save current document as" field. When the dialog box opens, this field contains the current name of the document. If the document has not been saved yet, a default name is provided, i.e. — Untitled-1, Untitled-2, etc., which may be changed as desired. (The number in the name of an un-named model indicates how many new documents have been opened during the current session in Vision3d.)

When a document is given a new name during saving with this command, the data path — new name, disk volume and folder location — is retained as the current name and data path for that document. Future saves using the Save command will use this name and location to save the document. The name and saved location of the active model may be changed at any time using the Save As command.

Saving optimized palettes with images and animations

The Save As command can also be used to save rendered images. All color renderings are created in 32-bit color (16.7 million colors); gray-scale images are 8-bit (256 colors). During the save operation, you may save the image in lesser color/gray-scale depths, if needed for compatibility with other software or to save disk space.

It's important to understand that if an image is saved in a lesser color depth and its window is then closed, the original pixel information is gone and cannot be retrieved from that file.



PICT File Options...

Resolution:

○ 1 bit/pixel
○ 2 bits/pixel
○ 4 bits/pixel
● 8 bits/pixel
○ 16 bits/pixel
○ 32 bits/pixel

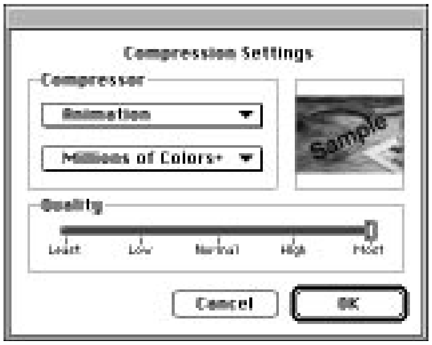
Palette: • System

☒ Enable dithering
☒ QuickTime™ Compression

After entering a name and designating where the image is to be saved in the Save As directory dialog, a second dialog opens to choose the color depth, enable or disable dithering, and a color palette. If you choose to save the image in 256 colors (8-bit) or less, the Palette field is made active to also choose the type of color palette to be used. You may choose from the default System palette, an optimized color palette, or a standard 8-bit gray-scale palette.

QuickTime compression

Additionally, you are given the option of compressing the image using QuickTime. This is available only if QuickTime is present in the System Folder as an active extension.



Compression Settings

Compressor: Animation
Millions of Colors+

Quality: Most

If QuickTime compression is chosen, a third dialog appears to set the compression method and the amount of compression desired. Vision3d always suggests the best image quality when using JPEG compression. Obviously, any of these suggested settings can be changed as desired.

For single images, “Photo - JPEG,” color, and high image quality are the default settings. “None,” millions of color +, and most quality are the settings suggested in this dialog for saving PICS files. With Quick-

Time, this compression setting dialog appears IN PLACE of the second dialog. It suggests “Animation,” millions of color +, and most quality when saving QuickTime format. More information concerning QuickTime compression is presented in the Animation chapter.

You should know that JPEG (Joint Photographic Enhancement Graphics) employs a “lossy” algorithm, in which visually redundant pixels are removed from the image as it is compressed. This is how JPEG is able to effect such significant compression ratios. Upon re-opening the image, the discarded pixels are regenerated as closely to the originals as the JPEG algorithm can determine. The tighter the compression, the greater chance for error during de-compression and the more noticeable the image quality loss.

Opening an image with an optimized palette

An optimized color palette may be attached when saving an image in 256 colors or less, whether the image originated in Vision3d or another source. Upon opening that same image again in Vision3d or other applications, the attached palette can be substituted for the current system palette.

The advantage of an optimized palette is apparent when displaying an image on monitors with 256 colors or less. Optimized palettes minimize the dithering effects visible in an image due to QuickDraw.

Under System 6.0, only color palettes can be optimized; gray scale palettes must use the standard system grays. Under System 7.0, both color and gray scale palettes may be optimized using additional system routines. While you may not notice, or even care about, the difference, it can be significant in the quality of alpha channels

Show/Hide Clipboard command (Edit menu)

The Show Clipboard command displays a window showing the current contents of the Clipboard. If already open, Show Clipboard brings the Clipboard in front of all open windows. If the Clipboard window is the currently active window, the command changes to Hide Clipboard. Hide Clipboard has the same effect as clicking on the close box of the Clipboard window.

The contents of the Clipboard are not editable in the context of the window. To become editable, the contents must be pasted into an application capable of working with the data. The contents of the Clipboard are replaced each time a cut or copy operation is performed. Also note that the clipboard memory buffer is erased when the computer is restarted or shut down.

Show/Hide Rulers command (Edit menu, ⌘, [comma])

The Show Rulers command displays rulers along the bottom and right edges of all views in the active window. Show Rulers is available when a document is open and the rulers are hidden. The Hide Rulers command toggles in the same menu location with Show Rulers. Hide Rulers is available when a document is open and the rulers are showing.

If the active window has been split into two or more panes, each pane has its own set of rulers. If a window pane/view is moved in or out with the view rotate tool (or the view is magnified with the magnify tool), thus changing the scale of the objects in the view, the rulers for that pane also change to show the new scale of objects in the view.

Rulers are available only in views containing “non-axonometric orthographic” projections of the model. These are views with no perspective and that are perpendicular to the model’s absolute X, Y or Z axis. Views that qualify have active rulers. They provide valid information about object sizes contained in the view and they correlate with the numeric values in the feedback area at the top of the active window. Rulers in views that do not qualify will be present but dimmed.

The orthographic views that can use rulers and the axes represented by the rulers are:

- **Front:** X axis on horizontal ruler, Y axis on vertical ruler.
- **Back:** Same as Front view, except numbers on each ruler are in reverse sequence.
- **Top:** X axis on horizontal ruler, Z axis on vertical ruler.
- **Bottom:** Same as Top, except reverse sequence.
- **Left:** Z axis on horizontal ruler, Y axis on vertical ruler.
- **Right:** Same as Left, except reverse sequence.

These views are the standard home positions for views, selectable from each view title’s pop-up menu. While the Isometric view is also considered a standard home position, it is not aligned to one of the axes and so cannot use rulers. It is technically an “axonometric” view. The other views are referred to a “non-axonometric.”

Assuming that objects in a model are similarly aligned with the X, Y, and Z axes, when these views listed above display parallel projections (no perspective) of the model, they are perpendicular to the object’s faces and thus will show only one face of each object in the model. The isometric view is called an axonometric view because, given the same model orientation, it shows more than one face of the objects.

Palettes command (Windows menu)

This command opens a sub-menu listing the various palettes. You may show or hide the Button bar or palettes by choosing the appropriate entry in the sub-menu.

To hide any of the palettes, use the appropriate "Hide" command in the Palettes sub-menu (Windows menu). To show a palette again, choose the corresponding "Show" command. If a palette is hidden, the last tool or setting selected on that palette is still usable. To choose another tool, the Tool palette must be made visible again.

Additionally, four small buttons are provided on the far right edge of the Button bar to show or hide palettes. You can simply click on the appropriate palette button in the button bar to toggle the desired palette as shown or hidden.

Alternately, you can click on the close box in a palette's title bar to close it. However, you still must use the Show/Hide palette command or click on the appropriate palette button in the Button bar to show that palette again. The Button bar does not have a button to hide itself.

Split View command (Windows menu)

The Split View command splits the active model window view in half vertically or horizontally, which ever is the bigger dimension. This command is active when a model window is the active window. If splitting the active view will reduce size of the two resulting view below the minimum allowable size, the command is not available.

Choosing this command has the same effect as double-clicking on either of the Split bars found in the window frame of a model window.

Undo/Redo command (Edit menu, ⌘Z)

Undo reverses the last action performed. The menu entry identifies what can be undone. For example, "Undo Object Rotation" is the command that appears after you have used the Object Rotate tool.

Some actions or operations in the program cannot be undone. When Undo is not available or a particular operation cannot be reversed, the phrase "Can't Undo" is displayed in dimmed or gray letters. Often, when actions cannot be undone, there is another command or tool that can reverse the first action.

After an action is undone, the menu entry changes to "Redo." This means that the previous action may be re-implemented by this command. Undo and Redo may be alternated as often as needed.

Chapter 2: Managing Documents

StudioPro works with original models created in StudioPro or with files modeled in other applications, then imported into StudioPro. Importation of non-StudioPro files is made possible through extensions. These extensions are either provided with the program when purchased, or are available for purchase separately from Strata Incorporated or third-party developers. For the latest information about the availability of import/-export extensions for specific file formats, please call Strata Incorporated.

Whether created or imported, once in StudioPro, the data becomes part of the current StudioPro model. When you save a model, it is saved as a StudioPro model by default. Other file formats may be saved as extensions allow to export the data for other applications to use.

However, you should know that saving a StudioPro model in other file formats may lose some of the model's surface and lighting information. You should always insure that a model is saved in StudioPro's native file format as a master, no matter what other formats are exported.

Images, too, can be read and saved in various formats, depending on the extensions present.

Opening models

Models not currently being edited are stored as documents or files on disk. When a model is being worked on, it is loaded completely into RAM and available for editing using the StudioPro tools. At least one window per model must be open on the screen to keep its data in memory. Closing the last window of a model will close the model as well.

Creating new models

When you create a new StudioPro document, no objects are in the model. The empty model is your 3-D “space” to do with as you please. Using 2-D instruments (mouse, drawing tablets, etc.) on 2-D viewing planes, the drawing operations are transformed into 3-D models and scenes. The model is displayed as 2-D representations on the screen from various directions. These 2-D representations of the model can also act as a viewing position to render more realistic images, which are then displayed on the monitor or output to 2-D devices, such as a printer or film recorder.

When a new model is begun, a single window opens containing the default view(s) of the model space as chosen in the preferences dialog box. The modeling views use the default foreground and background colors, also set in the preferences dialog box.

A window may be split to contain two or more panes within it, or you may open other windows and configure them as desired. As windows are opened, sized, moved on the desktop and views chosen for displaying in the windows, you may save view configurations by name using the “Save window” command, in the windows menu. If the window configuration is saved as a global template, it will be available in other models. These saved windows may be recalled as needed. Local view configurations are retained as part of the model and available in future sessions with that model. Each model can have its own set of saved view configurations or use the global templates.

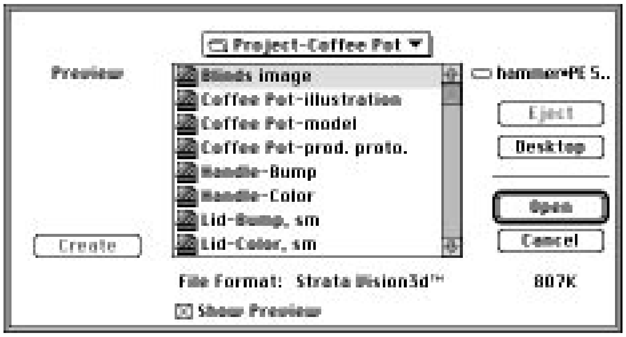
All modeling views display at 72 d.p.i. Rendered views can be of higher resolution, depending on the post-rendering need for the image.

Loading existing models

The Open command is used to open previously-saved StudioPro models. You may also use the Open command to load other 3-D files recognized by StudioPro and convert them to StudioPro format in one step. The other file formats that StudioPro recognizes depends on which import/export extensions are in place in the “Extensions” folder when StudioPro is launched for the current session. Extensions are discussed elsewhere in this manual. For information about extensions to handle specific file formats, please call Strata Incorporated technical support.

To open existing StudioPro models, as well as documents saved in other file formats, follow these steps:

1. **Choose the Open command from the File menu.** The Open dialog box appears.



Project-Coffee Pot

Blinds image
Coffee Pot-illustration
Coffee Pot-model
Coffee Pot-prod. proto.
Handle-Bump
Handle-Color
Lid-Bump, sm
Lid-Color, sm

hammer•PE 5..

File Format: Strata Vision3d™ 807K

☒ Show Preview

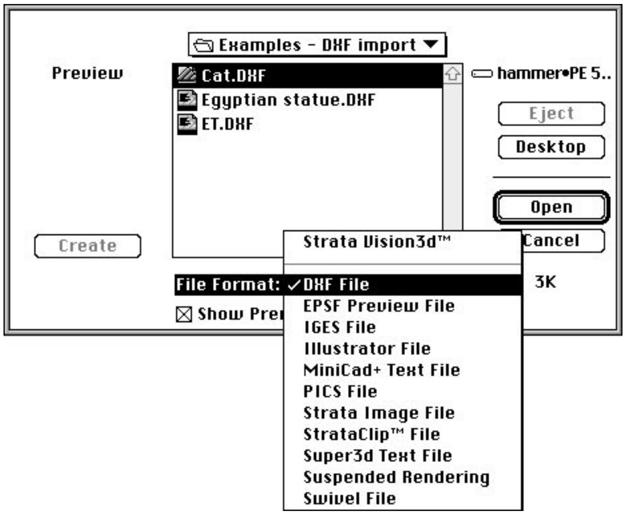
2. **Locate the folder that contains the file you wish to open.** Use the drive and directory buttons, if necessary.

3. **Click on the file name, then click on the "Open" button.** If the file type and document data agree, the file loads. You may also just double-click on the file name to achieve the same result.

Only files with formats recognized by StudioPro appear in the Open directory dialog box. When a specific file name is highlighted, the format type is displayed at the bottom of the dialog window. To be as accurate as possible in presenting files for importation, a file's format is determined by StudioPro in a two-step process:

- **By looking at its four-character "file type" code**
- **By interrogating the file itself to see if it conforms to the indicated file format.**

If the four-character code, called the "file type" is set correctly, you can use the Open command to directly load a document that StudioPro recognizes, and StudioPro will automatically select the correct file format extension to read the data.



File Format: ✓DXF File

The Open As command is also provided for those situations when the file type designation is incorrect for the document you wish to open. You can then force StudioPro to open the file in the correct format. If you force the format used to open a document and the chosen format does not match the actual data in the file, an alert will be displayed notifying you of the error and the import operation will be canceled.

When a StudioPro model is loaded, it opens with the window and view configuration in exactly the same arrangement as when the model was last saved. Other file formats open with the default window setting from the preferences dialog box.

Working with multiple models

You may sometimes want to have more than one StudioPro model open at the same time and switch between them. Alternating between models can be useful when you want to move objects and applied textures from one to the other. The number of models that can be open at one time depends on the models' sizes and the available memory.

While more than one model may be open at one time, only one model can be active at one time. Menu commands and/or the tools operate only on the active model, determined by the active window and the active view within that window. To switch between models, click on the view in the window containing the model you want to make active, if accessible on the desktop. If the desired window/view is not visible so that you can select it with the cursor, i.e. the window is completely hidden behind another window, you may choose the window (and thus, the model) from a list of open windows in the Windows menu.

Saving models

Models may be saved at any time during a session. It is always good practice to save periodically, but especially when you make a major addition or modification to a model. Then, if your equipment malfunctions or loses power, you can recover your work at the point of the last save and minimize any losses.

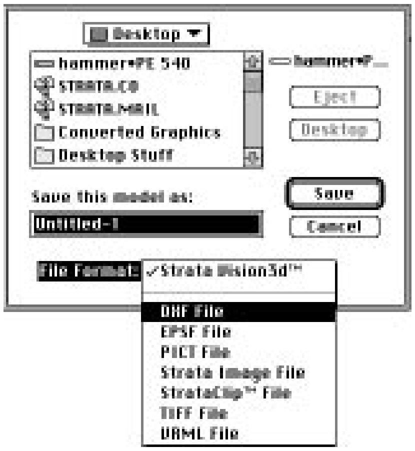
You should make sure the applicable window is active before selecting one of the Save commands. The same Save and Save As commands are used for saving models or saving rendered images of the model. Which one is saved depends on the active window. If the active window is associated with a model (that is, contains a view of a model), the Save commands apply to the model. If the active window contains a rendered image, then the Save commands act upon that image.

Saving a new model

When you save a new model for the first time, you are asked to specify the name of the document and where you want to store it. You should save a new model at an early stage to establish its name and storage location, and to preserve the work you have done to that point. Regardless of the application, it is always a good procedure to save often.

The steps to save a new model are as follows:

1. **Make sure the active window is part of the model to be saved.** If more than one model is open, make sure it is the correct model.
2. **Choose the Save command from the file menu.** If the document has never been saved, the “Save As” dialog box appears.



Save this model as: Untitled-1

File Format: ✓Strata Vision3d™

3. **Type a name for the document.** The default name of “Untitled-1” is provided in the name field, but you can enter your own name.
4. **Use the drive and directory buttons.** These are provided to choose where you wish to store the document.
5. **Click on the Save button.** Pushing the Return key also initiates the save operation.

The document is saved on disk with the specified name in the specified location. The default format used to save the model is “StudioPro model.” After a model is saved, it remains in memory — and on the screen — for further editing.

Saving a copy of a model

You can also create and save copies of a model using the “Save a Copy” command. The purpose of this command is to specify a different name and location for the copy. It operates similar to the Save As command, but the model being copied is not renamed internally and retains its original name and data path.

Recalling a previous version

The “Revert to Saved” command allows you to restore the last-saved edition of a model. All changes made to the model since the last save are discarded. The Revert command is only available if the current model has been saved and further changes have been made to the model, but not saved. The revert operation is automatic, once approved. That is, it automatically loads and replaces the open model of the same name.

Closing models

One or more models may be open in StudioPro at one time. The active model is determined by which model is represented in the active window. Closing one model automatically makes another model active, if multiple models are open. If no models are open, the palettes are also removed from the desktop.

There are three ways to close a model in the StudioPro program:

- **Click on the close box of the last window open for that model.** If more than one window is open, you may close a window without closing the model. However, closing the last window for that model will close the model.
- **Option-click on the close box of a window.** Hold down the Option key while clicking on a window’s close box. This will close all windows for the model associated with that window, automatically closing the model as well.
- **Choose “Close Model” from the File menu.** All windows currently open for that model will disappear from the screen and the model will be closed.

As previously discussed, if any changes have been made to the model since it was last saved, you will be asked if you want to save your model before closing. The model then disappears from the screen…and from memory.

Saving images

You may save images from either the modeling windows, camera windows, or rendering windows. Images are 2-D representations of the model, but contain no model information. When saved, they become completely independent of the model that was used to create them. They may contain pixels or draw commands, depending on the way the image was saved.

The Save and Save As commands are used to save images. They take the data from the active window to create the image.

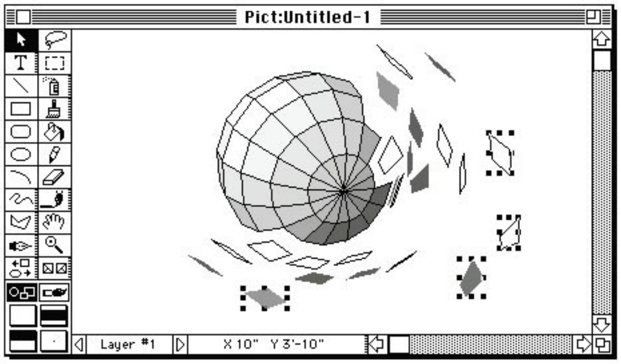
PICT is the default format for saving rendered images. However, the StudioPro program can save images in alternate file formats. The formats available depend on which extensions are installed. The format is selected using a pop-up menu in the Save As dialog box.

Saving images from modeling windows

The images displayed in the modeling windows can also be saved by using the Save As command and designating PICT as the format. StudioPro then knows to save an image of the active view of the model as a PICT file rather than the 3-D data of the model.

Views from modeling windows can be saved as draw-type PICTs

…When opened in a drawing program, the image can be dissected or altered, as this Canvas™ document demonstrates



Images saved in this manner can be wireframe, solid or flat shaded in appearance. These are the display methods available to the modeling windows. When the PICT is opened in a drawing program, it will be an object-based image

Unlike the pixel-based images saved from the rendering windows, images saved directly from the modeling windows are saved as QuickDraw commands in PICT format. In either case, this means the file will use the resolution of the device to “draw” its image of wireframe lines or shaded polygonal surfaces.

Rendering windows and rendered images

Snapshots may be taken in a modeling view or from a camera view. A snapshot automatically opens a rendering window in which the image “develops.” Modeling views offer limited adjustment of perspective, while camera views have a fully variable perspective control and are interactive with its camera “object” positioned in the model. A camera view presents the model as seen through the camera object’s view finder.

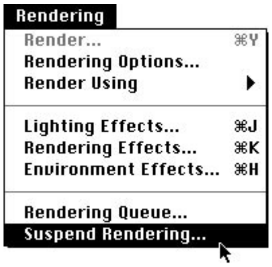
Images in rendering windows may use one of the higher-level, photo-realistic rendering algorithms provided in StudioPro. These rendering methods are capable of creating images with true reflectivity, refraction, detailed surface maps and more.

Suspending and restarting renderings

As one of the most significant features of StudioPro, suspending and restarting renderings allows you to schedule rendering time at your convenience. You may defer snapshots immediately from the Render dialog, then queue them for a more convenient time. Or you may suspend a rendering in progress and restart it later without loss of the work already performed. The rendering process on restarted renderings picks up right where it left off.

To suspend a rendering in progress:

1. **Make the rendering window active.** It contains the rendering you wish to suspend. StudioPro’s commands and tools apply only to the active view.
2. **Call the Suspend Rendering command in the Rendering menu.** If a rendering window is not the active window, the Suspend command is not active.



3. **Enter the name and location.** A standard save dialog appears. Use the directory controls to designate the location where you wish to save the suspended rendering file.
4. **The file is saved.** When the saving operation is complete, the rendering window closes automatically and the rendering terminated.

If you wish to have a rendering continue after using the Suspend Rendering command rather than be terminated, you can hold down the Option key. This would be useful if you wish to create a back up in suspended rendering format of the work done to date, but continue on with the rendering project. That way, if the rendering should fail due to a crash or power failure, you can just restart the suspended rendering and pick up the project again from that point.

In the case of an animation, the behavior of a backup suspended rendering is slightly different than for still images. Be sure to read about saving animations later in this chapter.

Suspended rendering may be restarted at any time. To restart a rendering that has been previously suspended:

1. **Start StudioPro.** Suspended renderings are completely self-contained. In addition to the partially-rendered image, the model is also stored in the file. You do not need to have the original model open for the suspended rendering to restart.
2. **Select the Open command in the file menu.** Use the dialog controls to point to the location where the suspended rendering is saved.
3. **Select the file.** When the suspended rendering file is highlighted, the file type feedback at the bottom of the dialog identifies it as a suspended file.
4. **Click the OK button or push the Return key.** The suspended file is loaded into StudioPro and the rendering process is restarted. Depending on how far along the rendering was when suspended, there may be a short delay as the renderer finds where it left off and sets up all the variables, pointers, buffers, etc. again.

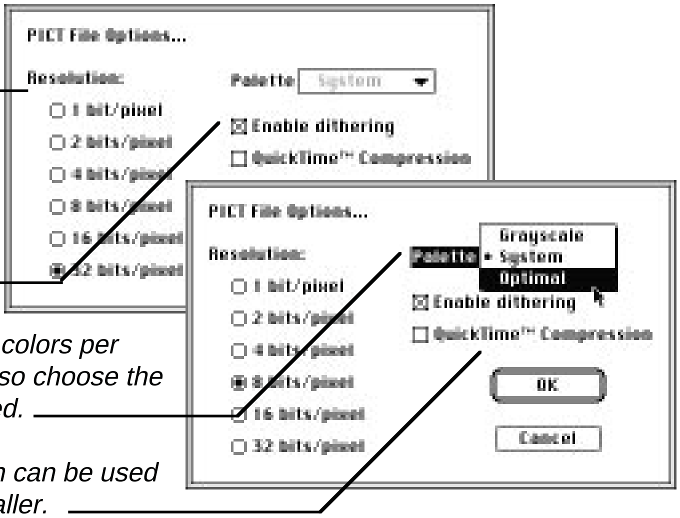
You may open as many suspended renderings as memory allows. If more than one suspended rendering is opened, you are given the option to render them all at once, sharing processing time among the windows, or queuing the renderings in sequence. Sequential rendering allows the computer to dedicate all its speed to one project at a time.

This procedure also applies to renderings that were deferred directly from the Render dialog. They are likewise saved as a suspended rendering file, but of course there is no partially completed image to retrieve.

Alternately, you may simply double-click on a suspended rendering file while on the desktop. The Finder will automatically launch StudioPro, then load the suspended rendering file. You can directly select and open multiple rendering files in this manner, with the same parallel or sequential rendering options once StudioPro is running.

Saving still images

The image in a rendering window may be saved at any point during the rendering process. The appearance of the rendering at the point when it was saved will become a separate file. Subsequent saves of the image, as it continues to render, can be given another name or replace the previously-saved image. Once a rendering is completed and the final image has been saved, the Save command becomes inactive for that rendering window, since no more changes can be made to the image. If a filter is applied to the image (through the Modify image command in the Edit menu), then the Save and Save As commands are available to store the changes made by the filter.



PICT File Options...

Resolution:
○ 1 bit/pixel
○ 2 bits/pixel
○ 4 bits/pixel
○ 8 bits/pixel
○ 16 bits/pixel
● 32 bits/pixel

Palette System
☒ Enable dithering
□ QuickTime™ Compression

PICT File Options...

Resolution:
○ 1 bit/pixel
○ 2 bits/pixel
○ 4 bits/pixel
● 8 bits/pixel
○ 16 bits/pixel
○ 32 bits/pixel

Palette

☒ Enable dithering
□ QuickTime™ Compression

When saving PICT images, you can choose the color depth

…and whether the image will be "ditherable" when using QuickDraw to display it.

For color depths of 256 colors per pixel or less, you can also choose the type of color palette used.

QuickTime compression can be used to make image files smaller.

When the Save command is called, it saves by default in PICT format. Saving in PICT format allows you to choose the color depth, color palette and dithering option. If the color depth is set at 32- or 16-bits/pixel, no palette option is required. Choosing 8-bits/pixel or lower allows you to designate the type of color palette used. You may choose a Gray-scale, System or Optimal color palette. An optimized color palette includes as many as possible of the colors contained in the image being saved. When re-opened in StudioPro or other applications that use optimized palettes, the image displays on the monitor with less dithering.

The Save As command allows you to choose from any of the other extension formats supported by StudioPro, including PICT and TIFF. You may choose the file format through the pop-up menu in the Save As dialog box.

Saving animations

StudioPro supports two formats for saving animations: PICS and Movie. The structure of these two formats is explained later in this chapter. The process of scripting and rendering an animation is discussed in the Animation chapter, later in this manual. Playing back animations is also discussed there.

While the PICS format requires no special extension to use, the Movie format does require the presence of QuickTime, a system-level init/extension. If QuickTime is not installed in the System Folder, the Movie format is not available. Further, saving an animation in Movie format also requires the presence of QuickTime during playback as well.

When the rendering of an animation sequence is initiated (via the Rendering Options dialog), you are required to specify the format PRIOR to beginning the rendering of the frames. Depending on your format choice, you will be asked to specify additional, secondary parameters. The nature of these parameters are explained in the following paragraphs.

Saving as PICS

As explained later in this chapter, PICS files are simply a collection of PICT images stored in the resource fork of a file, numbered in the sequence they should be displayed when the animation is played. As the frames of an animation are rendered in StudioPro, they are saved into the PICS file.

StudioPro thus needs to know only three things about saving an animation in PICS format:

1. **The name of the file**
2. **Where to save it** (volume/folder)
3. **The color depth at which to save the rendered images.**

Upon initiating an animation rendering, but before beginning to render even the first frame, you will be prompted for the file name and its storage location via the standard directory dialog.

Then, you will be asked to specify the color depth. This is the same dialog explained in the previous discussion on saving still images. The color depth dialog allows you to set anything between 32-bit and one-bit, even though you may have chosen gray scale in the Rendering Options dialog.

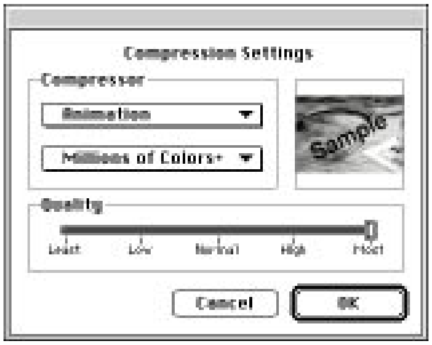
Upon clicking the OK button on the color depth dialog box, the animation begins rendering and each frame is saved as it is completed. Thus, at any time, you may look at the frames rendered to that point.

If you terminate the rendering, the saved animation will be a simple PICS file containing only those frames that were rendered and saved up to the point it was stopped.

If you suspend the rendering, the completed frames of the animation will be contained within a suspended rendering file. They are available for viewing (without damaging the suspended rendering shell) by using the Open As command and reading the file as a PICS format, even though it is technically a suspended rendering. In the case of a PICS file, the frames can be viewed from the resource fork without affecting the model data, which is stored in the data fork.

Saving as QuickTime Movie

QuickTime Movie is a more complex format than PICS. Because the QuickTime system extension is specifically designed to play an animation from hard disk (at as close to actual speed as possible), special compression methods are required when the individual frames are rendered and saved.



Compression Settings

Compressor: Animation

Millions of Colors+

Quality: Most

Instead of the color depth dialog box used for still images and PICS animations, a Compression Settings dialog box is presented to set the QuickTime compression method, the color depth, and the amount of compression (expressed in terms of image quality). The QuickTime compressors available and their compression mechanisms are discussed in *Chapter 7: Scripting an Animation.*

Upon clicking the OK button on the QuickTime Compression dialog box, the animation begins rendering and each frame is saved as it is completed. However, depending on the state of the file, you may or may not be able to view the animation directly.

If you terminate the rendering, without suspending the rendering process, the saved animation will be a fully-functional QuickTime Movie file containing those frames that were rendered and saved up to the point it was stopped. You will be able to view your animation in any application that supports the QuickTime Movie format.

If you suspend the rendering, the completed frames of the animation will be contained within a suspended rendering file. You will not be able to open and view the QuickTime Movie contained within the suspended rendering file. However, if you restart the rendering (or a copy of the rendering, if you wish to keep the suspended rendering intact) and let it render and save one more frame, then terminate the rendering without saving the suspended rendering, the file becomes a viable QuickTime Movie.

NOTE: *Regardless of the animation format chosen, the behavior of a suspended rendering file is the same. A saved suspended rendering is self-contained and will remain in this format until restarted and the next frame is rendered and saved. At that point, the suspended rendering shell is discarded. You must suspend the rendering again to store the model data along with the rendered frames.*

Opening images and animations

Saved images can be opened for displaying on the screen or for printing. These may be from any source, not just StudioPro-rendered images. StudioPro can also open images in different file formats.

The Open command is used to access images as well as models. Thus, files using any of the recognized image formats appear in the directory dialog along with StudioPro models. When a file name is highlighted, its file format appears in the dialog box below the directory. And when a file is selected for loading, the Open command knows whether to open it as a model or as an image, based on the format indicated by its file type.

You may use the Open As command to force an image to be read in a particular format when the file type code for that file is known to be incorrect. Obviously, the format chosen must match the data structure of the file. Otherwise, a “file read error” alert will appear and the reading of the file is terminated.

As presented in the previous section of this chapter, suspended renderings containing frames in PICS format can be opened for viewing with the Open As command. Suspended renderings containing QuickTime Movies will not respond to this procedure.

Copying and pasting images from an animation

Complete animations, as well as individual images from those files, may be copied and pasted as part of the extended functionality provided by the QuickTime extension. This functionality can be used on either PICS- or Movie-formatted animations. The current frame in an animation can be copied in the same manner as you would handle text in a word processing document.

Note that when copying single frames from a QuickTime Movie animation compressed with frame-differencing, there is a slight chance you may not get the entire image. With frame-differencing compression, only those frames at the whole second intervals are complete images, with frames in-between the whole seconds containing just the frame-differencing information. If you make sure you are copying only whole frames, you will be successful every time. Otherwise, the copy command may only obtain a partial image.

Copying and pasting complete QuickTime animations

To copy and paste entire animations into another document, hold down the Option key. In this manner, animations may be temporarily stored on the Scrapbook or pasted into other documents in any number of different applications. They can be played in that document by simply clicking on the image.

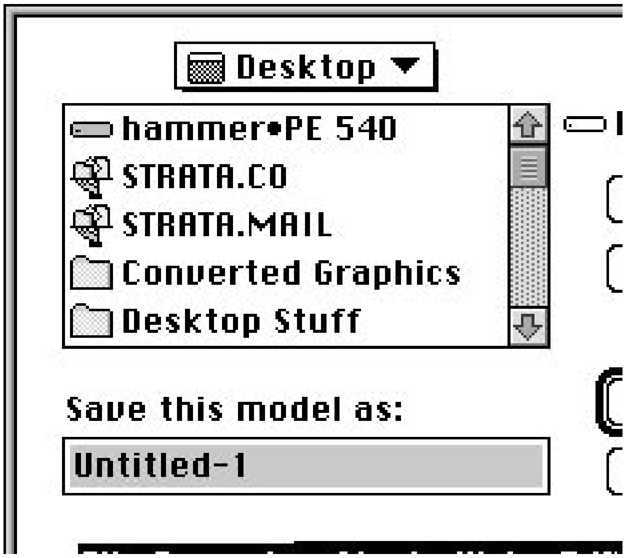
Again, QuickTime must be present for this extended capability to the editing commands. This feature is only available for QuickTime movies.

If you wish to paste an animation into an application that is not "QuickTime aware," you can use a special system extension called "Wild Magic."

Import/Export

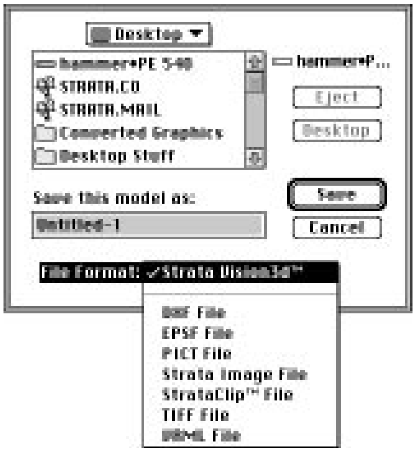
In addition to accessing native StudioPro models and images, the Open, Open As, and Save As commands can be used to load and store files of other formats. The balance of the discussions in this chapter assumes that the extensions included with the standard StudioPro software are properly installed in the Extensions folder. This process is explained in a "Read Me" found on disk 1 of the installation disks.

Models and images created with other programs can be brought into StudioPro through import extensions. Of course, StudioPro models and rendered images can also be saved in formats used by other applications.



Save this model as: Untitled-1

Using the Open command, you may read an import file and convert it to a StudioPro model directly upon loading. Or, by using the "Import" command, you may merge an imported file with an existing model. The Open As command is provided if a situation arises where you must "force" the use of a specific import filter on a file.



Save this model as: Untitled-1

File Format: ✓Strata Vision3d™

The Save command saves only in the StudioPro model format by default. However, the Save As command allows you to select alternative formats. When a new model is saved for the first time, the Save As dialog box appears. The default save format for the Save As command is also Strata StudioPro, but you have the option of changing the file format using the pop-up menu provided.

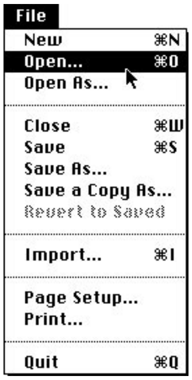
Quick Tutorial

This Quick Tutorial is meant to provide a basic understanding of the variations between the “Open” and “Import” operations. It is recommended that you complete the two examples here — Using the Open Command and Using the Import Command — one after the other so you can note the similarities and differences between the two operations.

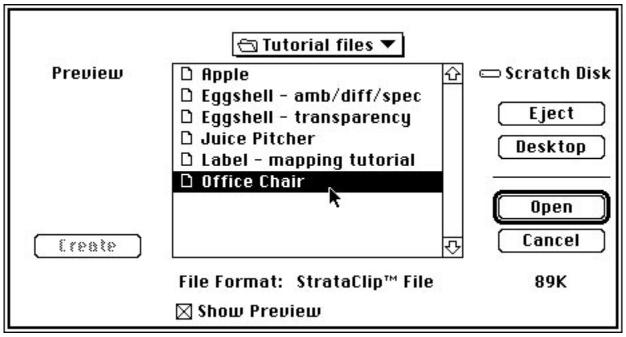
Because this is not a comprehensive presentation of all features related to these operations, you may wish to refer to the discussion on the Open and Open As commands in chapter one. Also refer to the section about opening images and animations earlier in this chapter.

Using the Open command

1. **Pull down the File menu and choose the “Open” command.** StudioPro responds with the standard Open directory dialog.



2. **Find the Office Chair file.** It’s located in the Tutorial folder inside the StudioPro folder. Note the file type indication in the dialog box when that file is highlighted in the dialog.

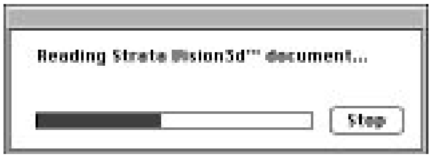


Apple
Eggshell - amb/diff/spec
Eggshell - transparency
Juice Pitcher
Label - mapping tutorial
Office Chair

File Format: StrataClip™ File 89K

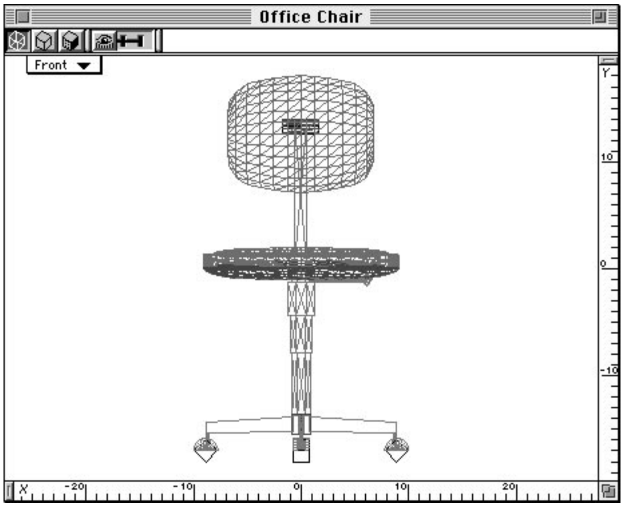
☒ Show Preview

3. **Click on the OK button.** StudioPro displays a progress bar as the file is read. When the progress bar reaches the end, the file opens as a StudioPro model.

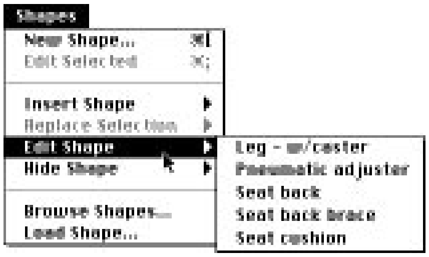


Reading Strata Vision3d™ document...

Note that the Office Chair object is already present in the modeling window that opens. It will probably appear quite large in the view. Use the Fit Views to All command in the Windows menu. This adjusts the view's scale to more closely match the chair.



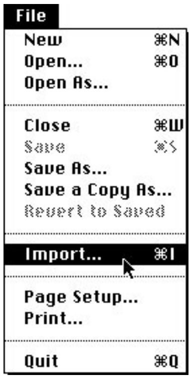
If you pull down the Shapes menu and choose the Edit Shape command, you will see a list of shapes similar to the following illustration.



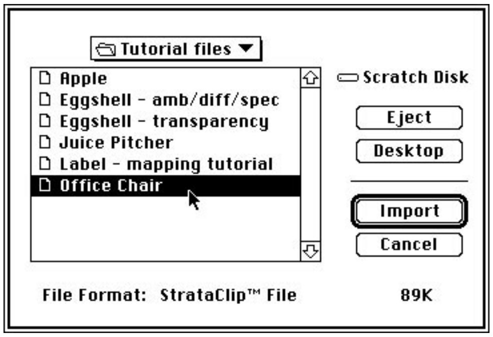
Compare this to the list of shapes you will see when you read the same model through the Import command, as demonstrated next. This may seem like a small thing, but the difference is significant.

Using the Import command

1. **Open a new model.** Don't close the first model, yet. You may want to compare the difference in the way the two models are structured.
2. **Place a sphere primitive anywhere in the model window.** This sets up the model into which you will merge the imported file.
3. **Choose the "Import" command from the File menu.** StudioPro responds by displaying the Import dialog. This is a very powerful command. It allows you to merge 3-D data from any source into the active model.

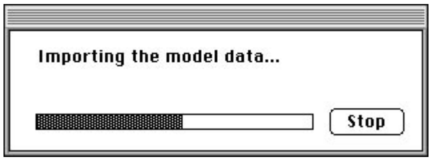


4. **Find the Office Chair file again in the Tutorial file.** Contrast this dialog with the Open dialog you saw in the previous example, specifically the lack of a preview area.



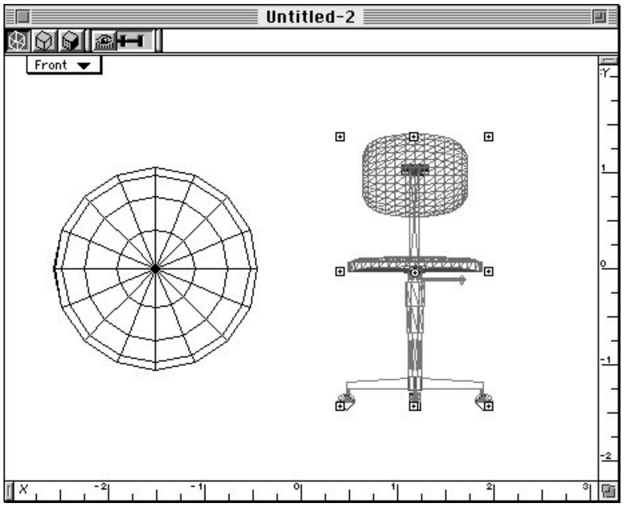
File Format: StrataClip™ File 89K

5. **Click on the Import button.** StudioPro displays the progress bar, same as before.

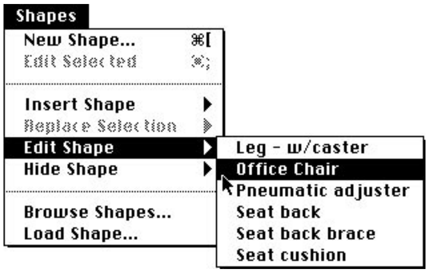

Importing the model data...

When the file finishes loading, you'll see the "loaded" cursor, indicating an imported file is ready for insertion into your model.

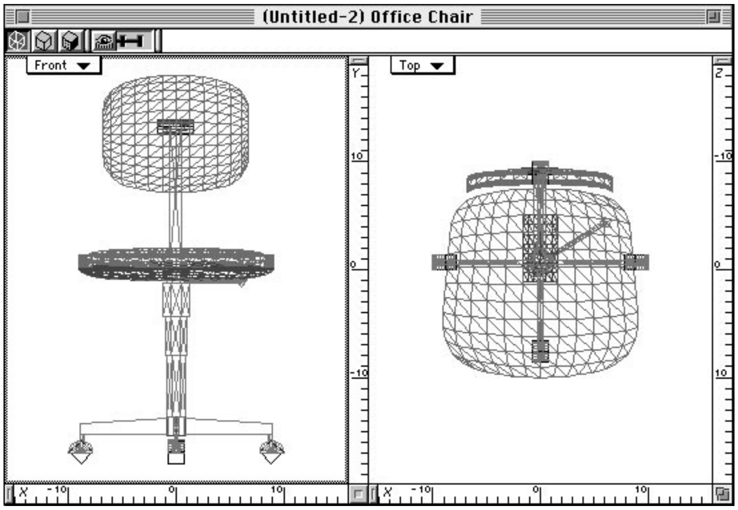
6. **Place the import file into the model.** There are two ways to do this: 1) Click once in the model window to place the import file at its actual size, or 2) click and drag a marquee in the model window. In this case, use the click-drag method. The click-and-drag method places the imported file into the existing model at the size defined by the marquee. This is a useful technique to match the scale of different files when merged together. Note that one of the detail is lost when a shape instance is scaled down. The original object is still at the original scale and may be edited as needed. The management of shapes and shape instances is discussed in *Chapter 4: Using Shapes*



7. **Pull down the Shapes menu and choose the Edit Shape command.** StudioPro displays a sub-menu showing the names of shapes within the model. Notice that "Office Chair" appears in this menu as a shape, unlike the sub-menu after using the Open command.



8. **Choose Office Chair from the sub-menu.** StudioPro displays the Office Chair's shape workspace as a separate modeling window. Note the title of the window indicates this shape window containing the chair is a workspace distinct from the instance of the object in the model window.



If you used the click-drag method of inserting the chair into the existing model, you should also note that the scale of the Office Chair instance in the model space is probably much different than its source in the shape workspace.

From these two examples, you have learned four things:

- **Using the Open command,** you can open a readable file as a Studio-Pro model
- **Using the Import command** allows you to merge the imported file into an existing model
- **Using the Import command** also preserves the imported file as a shape in its own workspace
- **Merged objects can be easily adjusted to match the scale** of the objects in the existing file

Again, for more information on working with shapes, please refer to *Chapter 4: Using Shapes*. Take the time to read this chapter thoroughly. This is an important feature that unlocks the real power of modeling.

Extensions determine files supported

The actual import formats supported by StudioPro depends on the import/export extensions present in the "Extensions" folder when StudioPro is launched. If no extensions are present, the program will only support its own native file formats: Studio Pro model and Studio Pro image, as well as PICT, of course (which is the Macintosh's native image format). PICT is the format required for surface maps in textures.

Opening and saving non-StudioPro files

StudioPro also supports files modeled in other applications that you may wish to open separately or import into an existing StudioPro model. Importation of non-StudioPro files is made possible through extensions. These extensions are either provided with the program when purchased, or are available for purchase separately. Whether created or imported, once in StudioPro, the data becomes part of the current StudioPro model. When you save a model, it is saved as a StudioPro model by default. Other file formats may be exported if desired as export extensions allow.

However, you should know that saving a StudioPro model in other file formats may lose some of the model's surface and lighting information. You should always ensure that a model is saved in StudioPro's native file format as a master, no matter what other formats are exported.

Importing

Importing capability is provided through "extensions" that add more features to the standard application. A versatile set of extensions is supplied as part of the standard StudioPro product. With these, the program allows you to import object data from several formats, including DXF™, IGES™, MiniCad+™, Super 3d™, Swivel 3d™, PICT, and EPS. Other formats may be supported via additional extensions as they become available.

NOTE: *Strata Incorporated has made every attempt to see that StudioPro works with the above-mentioned formats and products. However, because of format variations and product updates by other manufacturers, Strata Inc. cannot warrant compatibility with other software products. Please read the StudioPro Software License and Warranty Agreement.*

Exporting

StudioPro also allows you to export models in different formats. These are typically the same formats available for importing, depending on which extensions are available at the time it launches. As third-party extensions become available, you may also export in file formats not currently supported. The availability of export extensions allows you to use data from StudioPro in other applications and possibly on other computer platforms.

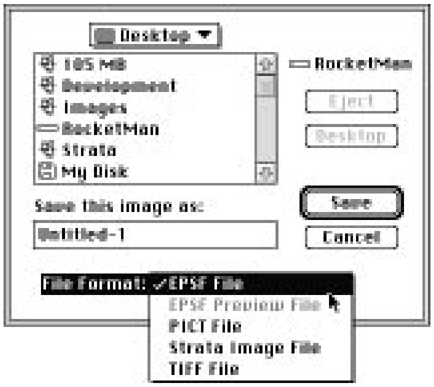
The export format is selected during a save operation. The Save command saves models only in Studio Pro format. You must use the Save As command to choose an alternate export format. If a modeling window is the active window at the time the Save As command is called, the dialog defaults to the "StudioPro model" format.

You should remember that no export file format supports all the scene description features available in StudioPro. With rare exception, none are even capable of storing lighting or surface descriptions. Even some geometry data can be lost. For example, you can't save 3-D primitives when exporting a DXF file. They are converted to less-efficient polygonal representations of the primitives' shapes.

Export formats for rendered images

By default, StudioPro saves images in the "StudioPro Image" format, which is considered the native image format for this application. This format is readable only by StudioPro for display or printing purposes. It is essentially the same as Apple's PICT format, but without any file compression. The ONLY advantage of this format is that an image saved in this manner opens quicker.

Rendered images may also be saved in different file formats. Again, the specific export formats supported depend on the extensions installed at the time StudioPro is launched. The ability to export images allows you to use them in other applications and possibly on other computer platforms.



Save this image as: Untitled-1

File Format: ✓EPSF File

The Save As command is used to export images, just as it can export models. The contents of the active window at the time the command is executed determines whether a model or a rendered image is saved. The specific format is chosen through the pop-up menu at the bottom of the Save As dialog box. Once a format is selected and the OK button is clicked, a secondary dialog box often appears with options for saving the image.

Of the formats provided for saving images, PICT and TIFF are the most commonly used. You can save images in one of these formats, then place the document in a page layout or paint program that supports the format.

File formats supported

To give you a better understanding of StudioPro's ability to use data from other applications, here is an explanation of the more common file formats used for exchanging 2-D and 3-D data between applications. Not all of these formats are supported in StudioPro, and those that are may only be in one direction (usually into StudioPro).

DXF

Stands for "Drawing Interchange" file format. This file type was developed by AutoDesk® to assist in exchanging drawings between AutoCAD® and other programs. It has been adopted by several other CAD applications as well, making it an unofficial standard for moving data between programs. Three-dimensional entities are supported in Release 10 or later of the DXF specifications.

DXF files can come from many different sources, including non-Macintosh applications. All sources should be compatible. The DXF format documentation is very explicit in how the data contained in the file should be organized. StudioPro adheres strictly to the published format specifications.

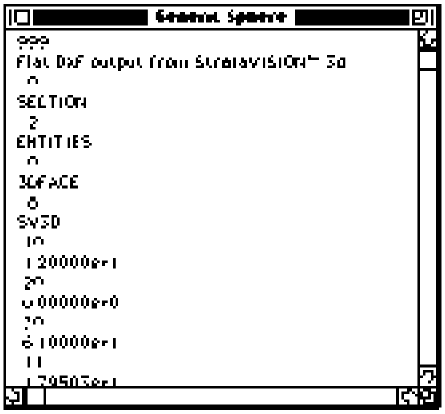
The only variation allowed in DXF text files is in line termination characters. Non-Macintosh text files typically use line feeds at the end of each line rather than carriage returns. That should be the only difference in a DXF text file; one that StudioPro automatically handles. Once the file is brought onto a Macintosh hard disk via the File Translation software, StudioPro should have no trouble reading the model data.

If you have a problem reading a DXF file exported from another application, you should examine it in a full-featured word processing program to see if it adheres to the format conventions regarding spacing and indents. Often you can compare a good DXF file with the one in question and see immediately if there are any variations in formatting.

If you have a problem reading a DXF created by StudioPro into another application, it may be the other application doesn't recognize comment lines. StudioPro puts one (and only one comment) at the very beginning of the file indicating the type and source of the file. To delete the comment, load the DXF file into a word processor and remove the first two lines only.

Anatomy of a DXF comment

A “999” means the NEXT line is a comment -- ignore both lines



```
999
Flat DXF output from StrataVISION™ 3d
  0
SECTION
  2
ENTITIES
  0
3DFACE
  8
SV3D
 10
 1.20000e+1
 20
 0.00000e+0
 30
 6.10000e+1
 11
 1.79503e+1
```

We recommend that you try deleting the comment and reading the file again. DO NOT delete any other information! The zero immediately following the comment line and subsequent information are essential for the target application to properly recognize the file as a DXF format and construct the 3D object(s).

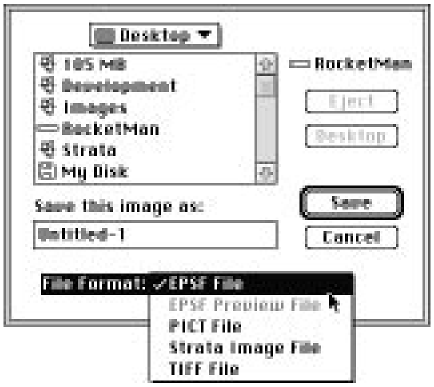
IGES

Stands for Initial Graphics Exchange Specifications. At this writing, the current IGES file specifications are titled Version 4.0. This file format was developed by a committee of industry, engineering and educational representatives assembled under the sponsorship of the U.S. Department of Commerce - National Bureau of Standards. The format defines a file structure and a language syntax for the representation of geometric, topological, and non-geometric object data. IGES is by far the most ambitious effort to standardize a file format for digital representation and data exchange. Since many sections of the specification do not involve the description of 3-D geometry or objects, StudioPro supports a subset of the complete IGES definitions.

EPS

Stands for Encapsulated PostScript. StudioPro reads EPS files to extract 2-D objects from the shapes and fills found in the preview image. StudioPro can also save in EPS format. Usually, an EPS file contains two versions of the data: 1) the actual description of the image in PostScript code in its data fork, and 2) a screen preview of the image in its resource fork.

This file type is most often used by drawing programs, but paint-type programs (as well as StudioPro) make use of the format by encoding an image in the file along with any special PostScript-based effects that can be added. For example, using the EPS file format, you may save a wireframe image of a modeling view that can be read by the Adobe Illustrator™ application.



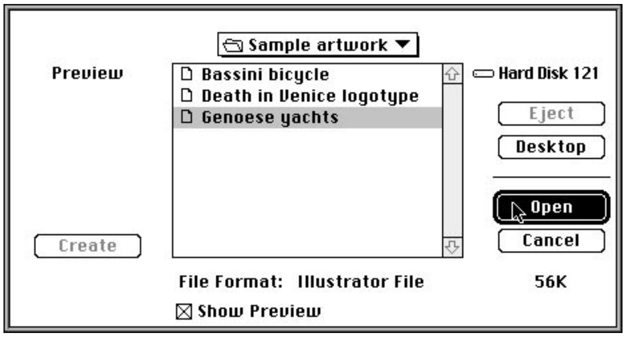
Save this image as: Untitled-1

File Format: ✓EPSF File

The EPS format saved by StudioPro is similar to the one used by Photoshop. That is, it is a pixel-based image "encapsulated" within a PostScript shell. EPS files from StudioPro can be opened in Photoshop, placed in Illustrator, or used in page layout documents.

Illustrator files

StudioPro will import files from Illustrator 3.0 or later. These files are automatically recognized when selected using the Open or Import commands under the File menu. Illustrator files are read object-by-object, accurately re-creating 2-D objects with the same bézier spline information and in the same two-dimensional spatial relationship as they were created in Illustrator.



File Format: Illustrator File

☒ Show Preview

Because of the way objects are handled in Illustrator 3.0, there are certain constraints on the way objects must be drawn in order to import them in a usable form into StudioPro. The following detailed explanations will be of greatest benefit to experienced users of Illustrator.

- **TEXT** — All types of text are supported, including freestyle, boxed, and along a path. Before saving from Illustrator, however, the text must be converted to outlines. Given that requirement, only PostScript fonts can be used, since Illustrator does not currently convert TrueType fonts to outline. Of course, letters with holes become compound objects when made into outlines. StudioPro accepts compound objects as 2-D polygons with holes.
- **INCLUDED FILES** — Placed art (that is, art placed into Illustrator), or included EPSF files are NOT supported, with one exception. If the placed or included file is (1) a valid Illustrator file, using the same format (text as outlines, etc.), and (2) the file is saved with the placed art enclosed, the importer will successfully import the complex file. However, the placed art will NOT retain its scale and position in spatial relationship with the parent Illustrator file.

  In other words, placed art is NOT supported. A better way to merge objects is to cut or copy from one Illustrator file and paste into another so all objects are in one file.
- **PATTERN FILLS or PATTERN STOKES** — Patterns are NOT supported. In StudioPro, the criteria is whether the object has a surface. If pattern fills are present, they may (in some unpredictable and as yet undocumented way) cause problems with the rest of a file being imported. The import extension should just ignore patterns and substitute the current color for fills or strokes. To be safe, DO NOT use pattern fills at all.
- **LINE WEIGHT** — Stroking is a special case. Basically it is ignored. However, if an object is not filled, it at least must be stroked. If an Illustrator object is neither stroked nor filled, StudioPro ignores the object entirely.

  Unclosed and stroked paths are brought in as 2-D bézier lines, exhibiting no thickness. This object may be extruded edgewise to create an object with surface. If the unclosed path is stroked and filled, it will default to a closed and filled object when imported into StudioPro. If the path is not filled, but is closed, it will create a hollow bézier object.
- **COMPOUND OBJECTS** — Compounds are both a blessing and a curse. Take the time to experiment with this feature. Create some

compound objects and import them into StudioPro. You will soon see the correlation between Illustrator and StudioPro. As you create more complex compound objects, you may see anomalies in their importation, as explained later. Think about the problems you see, then try to correct them in Illustrator. It is usually possible to make even the most complex compound object work, if done correctly.

Properly constructed, compound objects import into StudioPro as 2-D bézier objects with holes, making Illustrator a powerful modeling tool. When a compound is created of several objects, Illustrator groups the objects in a special way to indicate which are the main objects and which are holes within the objects.

To import correctly, the object in a compound must be the one furthest in the background (or the one created first, provided no rearranging of object order took place). All subsequent objects are holes, whether or not they are positioned inside of the actual object. The hole objects become REVERSED (this setting can be seen in the paint style dialog). When importing this compound, if any of the object's holes are outside of the object itself, especially if the holes reside outside of the bounding box enclosing the base object, unexpected results can occur.

Because the base object is always considered the one in the group that is the furthest in the background (in Illustrator), if you are not particular about the order of object creation or re-arrangement, you may create a compound that looks great in Illustrator, but the order of objects is backward, thus generating the formerly-mentioned unexpected results.

Another problem may occur when a compound is created using one or more objects that are already compounds; setting up a hierarchical compound. DO NOT USE THIS CONSTRUCTION TECHNIQUE. It doesn't even work all the time in Illustrator, and is guaranteed to give unexpected results in StudioPro.

Whenever a compound object is created, then released, there is further potential for problems unless you manually reset the affected objects. Adobe Illustrator does not reset the "Reversed" flag for objects intended as holes when a compound is released. If a new compound is created with one of these reversed objects, it reverses the wrong way, making it an object and not a hole, as may have been intended. This introduces more confusion over which is the object and which is the hole. A work-around is to select each of the released objects in turn and look at its Paint Style dialog box to see if it is reversed. If it is reversed in the dialog box, select the object by itself, make it a compound, then in the paint style dialog, click on the Reversed check box until it is empty (no "X" or gray fill), then release the compound. This effectively un-reverses the object. It may become part of a compound again and behave normally.

One last possible problem related to compounds. After a compound object has been created, if you click on the Reversed check box in the Paint Style dialog at all, erroneous reversals and un-reversals of objects in the compound will occur. On importation to StudioPro, the objects will be brought in as regions and none as holes. BEWARE! Do not change the reversed status of a compound object you wish to keep once it is created.

MiniCAD

Version 3.0 or later of MiniCAD includes a "Render in StrataVision" command in the File menu. Choosing this command prepares the MiniCAD model and exports it in a form readable by the MiniCAD (in) extension in StrataVision 3d, as well as StudioPro. The colors applied to objects in MiniCAD are included and become textures in StudioPro.

Exporting a MiniCad file in older versions

You can still import a file created in MiniCAD prior to version 3.0, but you must do it manually. The process is not involved, and is provided here for your information. Even owners of later version of MiniCAD may benefit from a knowledge of the steps involved.

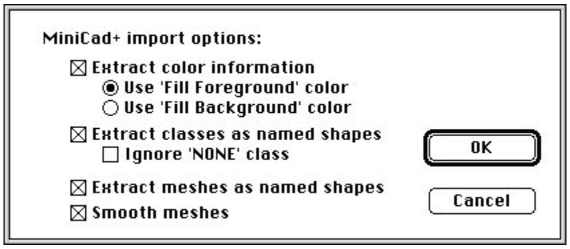
The "Export: Text" command in MiniCAD creates a text file with entity descriptions of the current MiniCad document. Prior to exporting the text file for StudioPro importation, the "Convert to 3-D" command must be executed to make all the objects in the document three-dimensional in nature. Even the 2-D polygons must be converted to 3-D to export properly.

To export a text file from MiniCAD to StudioPro, use the following steps:

1. **Create the model in MiniCAD.** It is assumed you can do this part.
2. **Select the Render command.** This is found in the 3d Creation submenu in the Δ (Delta) menu.
3. **In the dialog, click on the Strata radio button.** Then click on the OK button.
4. **Name the file and save it.** You may want to indicate its source in the name.
5. **Import the file into StudioPro.** Of course, the MiniCAD import extension must be in StudioPro's Extensions folder.

Bringing the file into StudioPro

An Import Options dialog box is available at the start of each importation of a MiniCad File. The state of the settings in the dialog box is maintained between each use of the I/O filter, including from one session to another.



MiniCad+ import options:

☒ Extract color information

● Use 'Fill Foreground' color

○ Use 'Fill Background' color

☒ Extract classes as named shapes

□ Ignore 'NONE' class

☒ Extract meshes as named shapes

☒ Smooth meshes

The meaning of the fields presented in the dialog are explained following the illustration:

- **Extract color information:** If this is selected, any color (foreground or background) applied to an object in MiniCad will be applied to that object as a texture in StudioPro.
- **Use "Fill Foreground" color:** Use the color set with the "Fill Foreground" command as the texture. If this option is not checked, no texture is created.
- **Use "Fill Background" color:** Use the color set with the "Fill Background" command as the texture. No texture is created if the option is not enabled.
- **Extract classes as named shapes:** If selected, all objects in MiniCad will be assembled into their named classes and the classes will be available as named shapes. The shapes will be placed in the appropriate position in the model. These are then available to be manipulated as you would any other shape.
- **Ignore "NONE" class:** If selected, the default class "NONE" will be ignored by the translator. Objects that would be in this class in MiniCad will be translated directly into the root model in StudioPro.
- **Extract groups as named shapes:** If selected, all grouped objects from MiniCad+ will be placed in a named shape and the shape will be placed in the appropriate position in the model. These are then available to be manipulated as you would any other named shape. If not selected, the objects will exist in an unnamed group in the model only.
- **Extract meshes as named shapes:** If selected, all meshes from MiniCad+ will be assembled into a named shape and the shape will be

placed in the appropriate position in the model. These are then available to be manipulated as you would any other shape.

- **Smooth groups:** If selected, all groups in MiniCad will be treated as a mesh of polygons and will be smoothed. If an object is not a polygon, it will simply be added to the group unaltered.
- **Smooth meshes:** If selected, all meshes from MiniCad+ will be converted into a smooth mesh in StudioPro.

  NOTE: *It is not always appropriate to bring in smooth meshes, especially in the transfer of MiniCad+ groups.*
- **Units:** All defined units in MiniCad can be read into StudioPro. All units will be converted into Strata Units at the appropriate scale.

PICS files

The PICS format is designed to store the individual frames of an animation in a single file. MacroMind®, the predecessor of MacroMedia®, introduced the format as part of the MacroMind Director™ application.

It is a simple format. Multiple PICT images are stored in the resource fork of the file. Beginning with #128, each image or frame of the animation is given a resource number corresponding to the sequence in which the images are to be displayed on playback of the file.

There is no provision for buffering the images or assisting the speed at which the animation is played. Those facilities must be provided by the software through which you are viewing the animation's frames. If a PICS file is played directly from hard disk, a playback rate near real time cannot be sustained. However, by loading all the frames of an animation into a RAM buffer ahead of time, the frames can be spooled to the screen in a smoother manner.

In addition to the playback rate, another limitation of this file format is that it cannot exceed 16 megabytes in size. The Mac OS will not support any file with more than 16 megabytes in the resource fork.

PICT

More accurately called PICT2, this image format supports raster and draw data saved in color. Up to 16.7 million colors are supported under 32-Bit Color QuickDraw. This format is an enhancement of the original PICT specifications. PICT is the native image format of the Macintosh, and is supported directly in the System ROM code. It isthe oldest generic file format on the Macintosh, and the most widely supported among the many available applications. PICT files support unlimited resolution of raster-based images, a feature StudioPro uses to advantage.

Extracting 2-D objects from B&W PICTs

In addition to viewing PICTs as images, bitmapped PICT files may be used as a modeling tool for creating objects in StudioPro. When imported or pasted into a modeling window, StudioPro performs an autotrace of the black areas in the image and extracts them as filled 2-D objects. These objects may then be manipulated just as a 2-D object drawn with StudioPro's own tools.

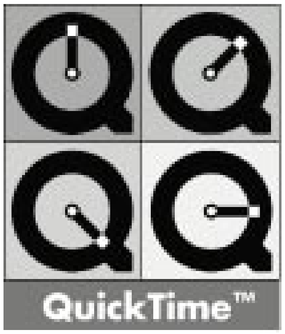
3-D Sculpt objects using grayscale PICTs

Grayscale PICT files may be used to create 3-D Sculpt objects. When imported or pasted into a model window, the grayscale information is used as elevation data to simulate actual 3-D geometry. The object is represented in the modeling window as polygonal mesh and can be manipulated as if it were a real 3-D object.

However, because 3-D Sculpt objects are not actual 3-D geometry, they have some limitations in their use. This object type is not editable in the other modelers and they cannot be rendered using the raytracing algorithm. Using the Facet command converts the Sculpt object to a polygonal object, but at a high object count and memory penalty.

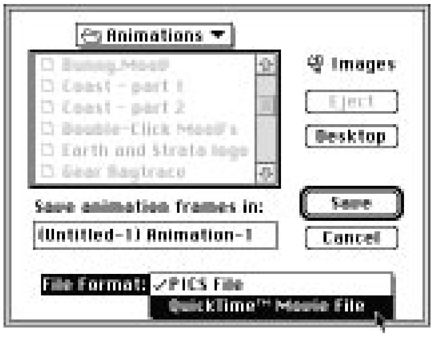
QuickTime Movie files

QuickTime is a revolutionary extension to the Macintosh system software that provides exciting capabilities to all Macintosh applications. It is a system-level architecture for the integration of dynamic data types — such as sound, video and animation — that will stimulate new directions in personal computing for years to come. Just as QuickDraw is a standard part of Macintosh system software that allows applications to integrate graphics in a consistent, mainstream fashion, QuickTime is a standard part of Macintosh system software that empowers applications like StudioPro to integrate dynamic media in a consistent, mainstream fashion.



QuickTime 1.0 or later works with all color-capable (68020 or later) Macintosh computers running system software 7.0 (actually, 6.0.7 or later) with at least 2 megabytes of memory. QuickTime builds on familiar Macintosh metaphors (i.e., cut and paste) that make it simple for users to work with the newly available data types.

When Apple's QuickTime system extension is properly installed in your machine's System Folder, QuickTime's image-handling and animation features become available within StudioPro. Specifically, QuickTime's Movie™ format is supported as an alternative for saving and playing animation files.



Save animation frames in: (Untitled-1) Animation-1

File Format: ✓PICS File

QuickTime Movies from any source can be opened and played (including sound) within a StudioPro animation window. Animations can be converted from PICS format to Movie format (and vice-versa) by simply opening and saving in the desired format under a different name via the Save A Copy As command.

Super 3d

Using the Export command in the File menu, Super 3d can export the current model document as a text file. Only information describing a model's 3-D object geometry is exported. Super 3d-specific information is not exported, because it is not understood by other programs.

Swivel 3d

StudioPro is able to read Swivel 3d's native file format directly. As with other formats, the objects become part of a StudioPro model once imported. However, only the geometric object descriptions are read by StudioPro. Swivel 3d's proprietary hierarchical linkage and motion constraint capabilities are not supported. StudioPro instead offers an alternative linking hierarchy (which has no motion constraints) designed to work best with its animation features.

TIFF

Stands for Tag Image File Format. TIFF is a versatile method of storing bit-mapped images at specified resolutions. Just like the PICT format, color and gray scale variations are supported. However, TIFF format is typically more efficient for high-resolution, full-color images. Many of the page layout programs prefer TIFF for handling images with higher than screen resolution. High-end color-separation and pre-press applications usually prefer TIFF files as well

RIB

Stands for RenderMan® Interface Bytestream. This file was developed by PIXAR® as a scripting language to control its RenderMan renderers. RIB files offer a full scene description capability. They can contain a detailed description of a scene to be rendered, including the objects by their geometry, placement, etc., and by the textures applied to their surfaces. Textures defined in StudioPro can be translated and exported as part of the RIB file, or you may substitute RenderMan shaders at the time of rendering. The lighting in a scene is also contained in the file, including information on the location and characteristics of every light source used to illuminate the scene.

The extension for exporting RIB files is not shipped with the standard StudioPro package. It is available separately from Strata Inc. for a nominal cost.

Additional formats (via third-party extensions):

Other formats may be available for import, export, or external rendering purposes that are not presented here. These may be subsequently made available from Strata Inc. in response to new products or market trends, or provided by other software companies to allow StudioPro to connect with their products.

For the latest information about additional extensions available from any source, you can call Strata Incorporated extensions product manager at (801) 628-5218.

Commands used with managing files

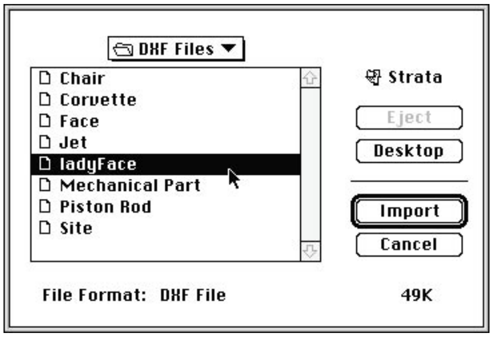
The following commands are considered pertinent to the file management and importing functions of the software. A detailed explanation is included with each listing.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

Import command (File menu, ⌘I)

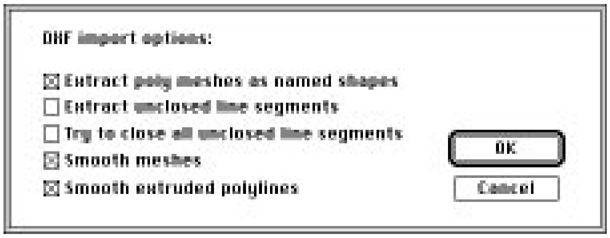
The Import command enables you to merge StudioPro models and object data from other formats into the current model. Import is available whenever a model is open.

Imported data is loaded into the model in Shape workspaces, then inserted into the model work space by clicking once or dragging the cursor in the active view. Clicking once inserts the data at a default scale; dragging the cursor defines the scale to fit the existing objects in the view.



File Format: DXF File   49K

When a file is selected for importation via a directory dialog box, the “Type” field indicates the file format in which it was saved. Clicking on the OK button initiates the import operation; clicking “Cancel” returns to the active window.



DXF import options:

☒ Extract poly meshes as named shapes
□ Extract unclosed line segments
□ Try to close all unclosed line segments
☒ Smooth meshes
☒ Smooth extruded polylines

You are given several import options that determine how the imported file is to be translated into StudioPro data. Each file import extension has a set of these options applicable to the format being read. Most give you the choice of smoothing the 3-D polygonal entities contained in the file and of identifying and naming shapes.

Import options set in these dialogs are retained from session to session. That is, the next time you import a file of a particular format, the Import Options dialog for that format appears with the same settings you chose the last time it was used.

Using the import/export extensions shipped with the product, StudioPro can import 3-D data from files using the following formats: DXF, MiniCAD, Super 3d, Swivel 3d Professional. Other formats may also be supported with extension import filters provided by Strata Inc. or third-party developers. If the file appears in the Import directory dialog and, when selected, its file format is correctly identified in the Type field at the bottom of the dialog, the chances are excellent that the file will import successfully.

Strata Incorporated *has made every attempt to see that StudioPro works with the above-mentioned formats and products. However, because of format variations and product updating by other manufacturers, Strata Inc. cannot warrant compatibility with other software products. Please read the StudioPro Software License and Warranty Agreement.*

Because all data brought into the current model through the Import command are treated as shapes, you should handle their editing and texture applications as you would other Shapes created directly in the program. Please refer to chapter 4 for an explanation of managing shapes and shape instances.

Extracting 2-D polygons from pixel images

In addition, the Import command can also create 2-D data from paint and drawing programs. In this manner, 2-D shapes can be extracted from black and white PICT and non-Illustrator format EPS files. These files must be saved in bitmapped (1-bit) form to work. If saved in color or grayscale form, the image is converted to a 3-D sculpt-type object.

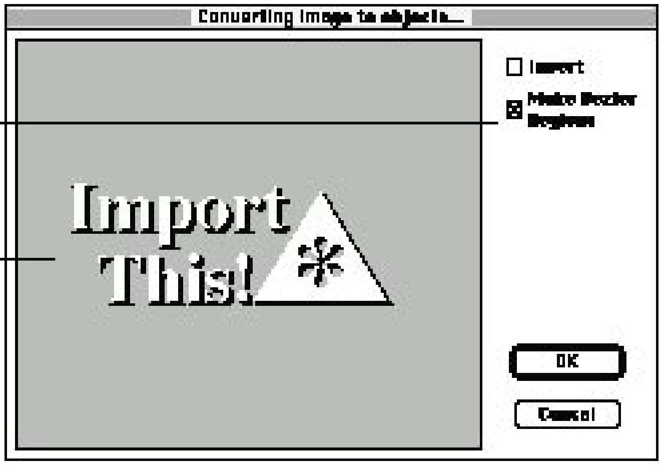
Again, the technique for extracting 2-D objects from bitmap images is essentially an autotracing process.

There are two types of EPS files: those from PostScript drawing programs, and those from paint programs that save pixel-type images encapsulated in a PostScript shell. Both have a PICT format preview image stored in their resource fork. Non-illustrator EPS files are included in this group because StudioPro uses the PICT preview to perform the extraction. Illustrator files are treated differently.

Obviously, the larger the image, the more accurate the extracted object's geometry will be — that is, more faithful to the original image. This is easy to accomplish in PICT format, but EPS files have a fixed preview image size, no matter how large the PostScript drawing inside. In this case, the options are to convert the PostScript information to Illustrator format and use the better importing method available to that file format, or re-save it in PICT format to gain access to the larger image.

The "Make Bezier Regions" option converts the pixel based image to a spline object.

The same PICT image can be imported in regular or inverted mode.



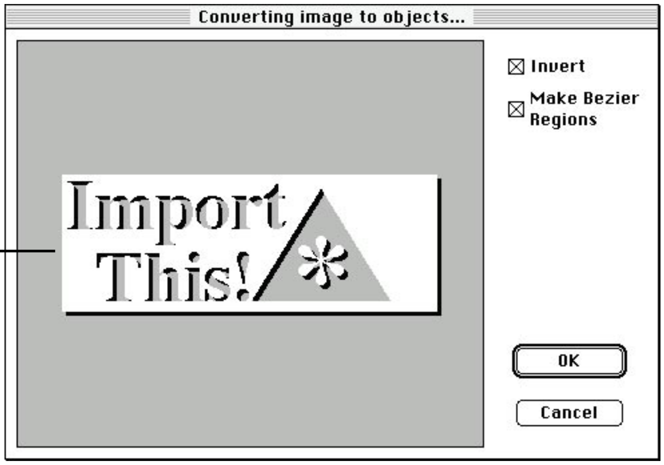
Converting image to objects...

☐ Invert

☒ Make Bezier Regions

OK

Cancel



Converting image to objects...

☒ Invert

☒ Make Bezier Regions

OK

Cancel

When an image file is selected for importation, StudioPro first interrogates the file and displays a Preview of the 2-D shape or shapes it will be able to extract. You may import the shape normally, where black is interpreted as surface and white is empty space. The preview will display the black surface areas as white in this preview, which is how they render as a StudioPro object without a texture applied.

Or you may invert the extraction process via a check box option in the dialog; resulting in the white areas of the image treated as surfaces and black as holes in the surface or empty space outside of the object shape. Again, in the preview, the areas of the image that will become object surfaces upon importation appear as white.

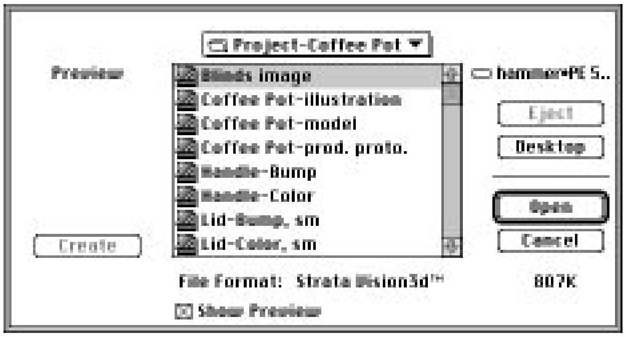
Clicking on the OK button reads the file and completes the extraction process. Clicking on the Cancel button terminates the importation of the file and returns to the active view.

Extracting 3-D objects from gray scale images

The Import command also recognizes gray scale images. But rather than converting the image to 2-D polygons, its gray values are interpreted as elevation data from which a 3-D object can be created. Once inserted into the model, you may handle it just like any other Shape.

Open command (File menu, ⌘O)

The Open command allows opening an existing StudioPro model. Open is always available, since more than one model can be open at a time. If not enough memory is available to open an existing model, an alert is displayed when the command is executed.



File Format: Strata Vision3d™

Choosing Open displays the Open dialog box. Depending on the import/export extensions currently installed, this dialog will also recognize alternative file formats that may be read into StudioPro. StudioPro models, DXF, Swivel 3d, Super 3d, PICT images, etc. may be opened with this command. The directory dialog box shows only those files matching the formats supported by the extensions.

Open As command

The Open As command allows opening files containing a pixel-based image or 3-D object information in non-standard formats. All files are displayed in the directory dialog. When a target file is selected, the

CHAPTER 3: MODELING OBJECTS

This chapter is a discussion of the modeling capability within Vision3d. Items discussed include the interface and operation of the various modeling tools, some helpful guidelines for effective modeling, and information you may find useful on the internal workings of some modelers.

The modeling tools in Vision3d can be classified in several ways. There are two-dimensional drawing tools, 3-D primitives, modelers that edit the geometry of 2-D objects, modelers that convert two-dimensional objects into 3-D objects, and modelers that modify 3-D objects. Some of these tools and modelers are built into Vision3d. Others are extensions to the main program. These extensions “plug in” to Vision3d seamlessly, similar to the way that system extensions work with Macintosh System 7.0 and later (only they’re more stable).

As you learn more about using Vision3d, it will become clearer that you can create objects equally well by more than one method or set of tools. There are no right or wrong ways to build objects in Vision3d, but some methods are usually more efficient for a given project than others. The goal is to use the modeling methods that is best, ie. — that saves time, effort, and RAM. Initially, you may have to experiment to get a feel for which modeling tools and techniques will be appropriate in circumstances you may face. Before long, you’ll develop an intuition for the ways that best fit your style and allow you to build the objects and models you want to create.

Built-in modeling tools

Some modeling tools are “hard coded” into Vision3d. They are part of the core program. The tools and commands to use these built-in modeling tools are visible on the Tool palette or in the Modeling menu, although the commands are not always available for use. When you use these modeling tools, their operations typically are performed directly in a model window.

Some of the more often-used commands are also represented by a button of the Button bar. The Button bar and its operation is explained later in this chapter.

Included Modelers

Vision3d ships with a number of modelers that are not built-in, but instead are extensions to the core program. When these modelers are present in the Extensions folder within the Vision3d folder, their commands are visible in the Modeling menu just like the built-in modeling tools. They, too, are represented by a button on the Button bar.

These extensions usually perform their complex modeling operations on existing objects or create new objects in different ways than the built-in tools can. Extensions typically do not work on selected objects in the model window, but rather in their own specialized dialogs or windows. Then they return or insert the objects into the model window when completed.

About Extensions

Extensions are add-on software tools developed by Strata Inc. or by third-party vendors to extend the capabilities and features of Vision3d. Some extensions ship with Vision3d. Others must be purchased separately. Extensions are separate files on the desktop that are placed in the Extensions folder within the Vision3d folder. Do not confuse this Extensions folder with the Extensions folder found in the System Folder. (Strata Inc. used this concept first, back before System 7 was released.)

Extensions typically fall into one of five categories: Modeling extensions, Import/Export extensions, Texture extensions, Image extensions, and Animation extensions. If you look in the Vision3d Extensions folder, you'll see folders for each of these extension types.

NOTE: *Vision3d extensions only have to be located somewhere in the Extensions folder. They do not have to be in any specific sub-folder within the Extensions folder. The various sub-folders are set up during installation for organization purposes only.*

Extensions are installed by placing them inside the Extensions folder. Whenever the Vision3d application program is opened, it scans the Extensions folder to determine which extensions are available. Available extensions show up as commands in Vision3d menus and as buttons on the Button bar. In order for extensions to be available for use, they must be within the Extensions folder inside the Vision3d folder. The "Extensions" folder name and locations is *NOT* optional; it must have this EXACT name (capitalized, no extra spaces, etc.). Otherwise, Vision3d will not see it correctly. As long as the extensions are inside the Extensions folder, which in turn is within the Vision3d folder, extensions are available for use.

Working with objects imported from other applications

A powerful feature of Vision3d is its ability to import files from other programs and merge them in models that may include existing objects created in Vision3d and other 3-D modeling software packages. Once files are brought into Vision3d using the "Import" command, they are treated just as any Vision3d object. Once in Vision3d, imported files have equal access to all available modelers and tools. They require no special treatment.

Quick Tutorials

The object of the exercises in this Quick Tutorial is to give you a basic understanding how different 2-D and 3-D modeling tools work, and how various modelers and tools work together to build objects. When you finish each exercise you may Save or discard the objects as you wish.

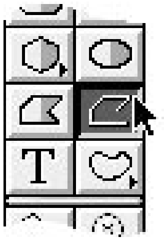
Exercise 1: Making a goblet

In this exercise, you'll make a 3-D goblet using the 2-D polyline drawing tool, the 2d Sculpter modeler and the Lathe modeler. This will help you understand how to use these tools and modelers, as well as learn a useful technique for drawing cross-sections for lathing.

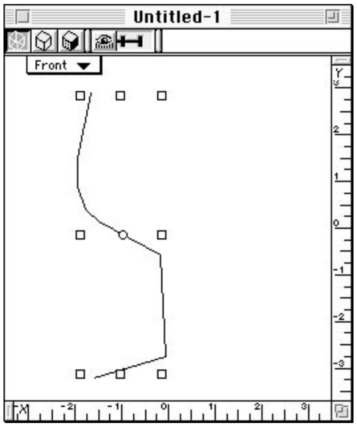
1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.

   You will hear this again and again throughout the manual. This view configuration is the recommended arrangement for the modeling phase of any project. Remember, "Use orthographic for modeling, perspective for rendering."

2. **From the Tool palette, select the open polyline tool.** This tool draws ONLY open 2-D lines.

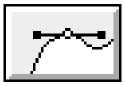


3. **In the model window, draw a cross-section of a goblet, similar to the illustration below.** Starting at the top, click once, then move the cursor down to draw the first line segment. When you're ready to change direction and draw the next segment, click once to create a vertex point. Then move the cursor to draw the second segment. Continue working your way down, clicking once at each vertex. If you make a mistake, simply push the Delete key to back up one segment.



Don't worry about making the drawing perfect. In step 4, you will use another modeler to edit and fine tune the cross-section. When you've drawn the last segment, double click on the last vertex point to end the drawing. Vision3d responds by re-drawing the cross-section in the model window, with selection handles surrounding it.

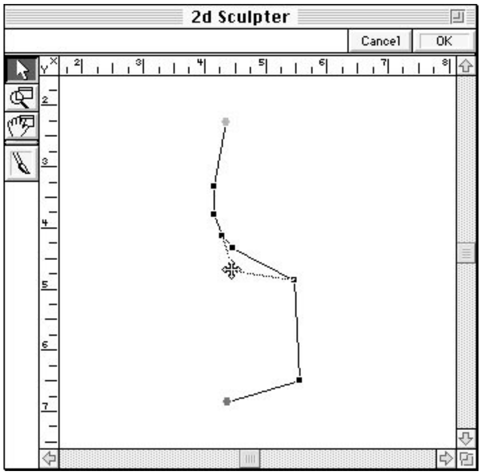
4. **With the goblet cross-section selected, choose the "2d Sculpter" command from the Modeling menu.** Optionally, you can also click on the Button bar's button that opens the 2d Sculpter.



You know an object is selected when eight selection handles appear around it. Vision3d responds by opening the 2d Sculpter dialog window.

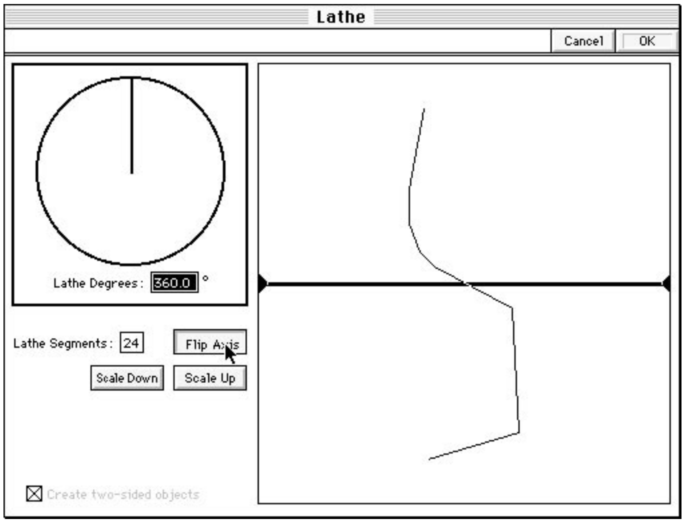
5. **In the 2d Sculpter window, edit the cross-section.** Click on vertex points and drag to move them and refine your goblet profile. You may add or subtract points as needed to create the desired curves.

For an explanation of how to add vertex points, please refer to the discussion on the 2d Sculpter modeler later in this chapter.



By moving individual vertex points in the 2d Sculpter modeler, you can fine-tune the geometry of any 2-D object. Vertex-level editing in this modeler provides more precise control over the shape of your object than is provided by the polyline tool used to draw the original cross-section.

6. **Click on the OK button in the top right corner of the 2d Sculpter window.** Vision3d responds by replacing the original polyline object in the model window with the edited cross-section from the 2d Sculpter dialog window.
7. **With the edited cross-section still selected, choose the Lathe modeler.** This can be done by pulling down the Modeling menu and choosing the Lathe command from the Swept Surfaces sub-menu. Alternately, you can simply click on the Button bar’s Lathe button on the Button bar



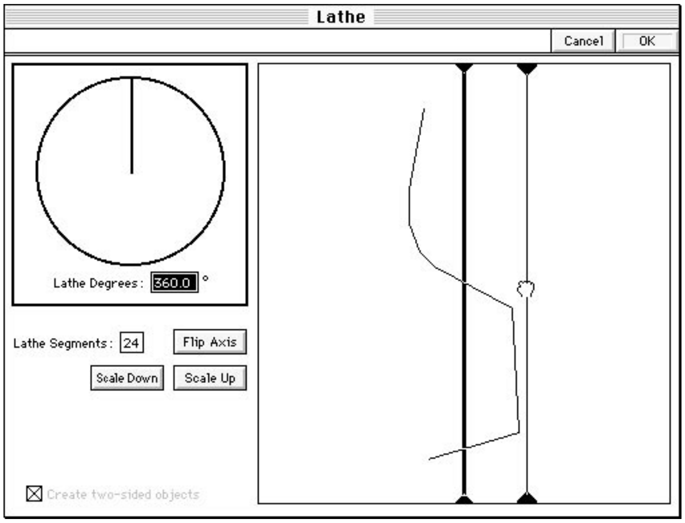
Lathe Degrees: 360.0 °

Lathe Segments: 24

☒ Create two-sided objects

Vision3d responds by opening the Lathe modeler window.

8. **Set the lathe axis position.** Use the “Flip Axis” button, if necessary, to turn the axis to a vertical orientation. Then move the cursor to a position over the axis. Notice how the cursor arrow changes to a hand icon, indicating a Grabber tool, when positioned on the axis. With the hand icon on the axis, click and drag the axis into position as shown in the following illustration.



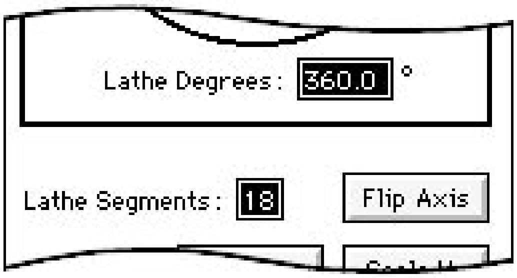
Lathe Degrees: 360.0 °

Lathe Segments: 24

☒ Create two-sided objects

You should position the axis so that it does not intersect the edge of the base.

9. **Check Lathe dialog settings.** Make sure the lathe degrees filed is set to 360° and the lathe segments is set at 18.

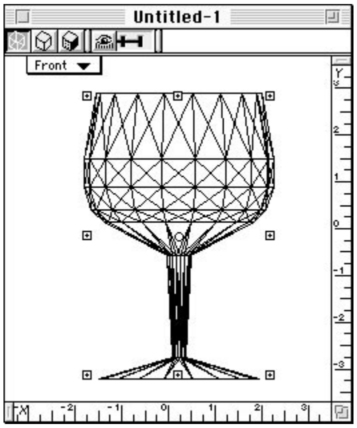


Lathe Degrees: 360.0 °

Lathe Segments: 18

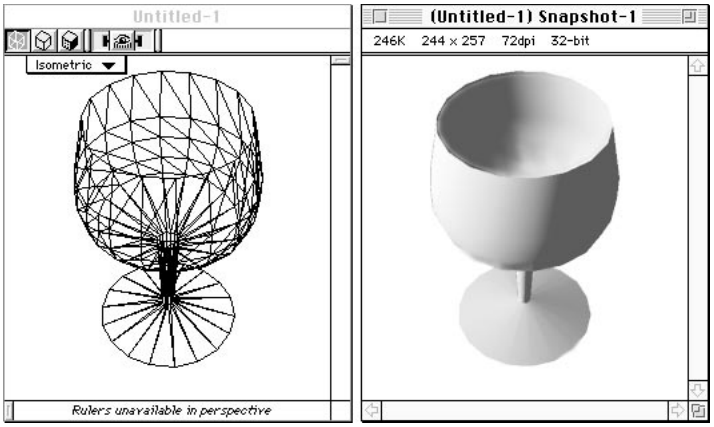
These are the values that give the best results without generating an excessive number of polygons. Note that there are no default settings for this modeler. Each time it is opened, the settings you used the last time will still be there.

10. **Click on the OK button at the top right corner of the Lathe window.** Vision3d responds by showing a progress bar as the Lathe operation occurs.



The lathed goblet is inserted into the model window, replacing the edited cross-section.

11. **If desired, change the view position and perspective setting to get a better look at your creation.** Choose "Isometric" from the view title pull-down menu at the top left of the model window. Now switch the three-position perspective control to Normal. Vision3d responds by re-drawing your goblet from the new view angle and with perspective.


Untitled-1
Isometric
Rulers unavailable in perspective
(Untitled-1) Snapshot-1
246K 244 x 257 72dpi 32-bit

If the goblet is too small to see clearly in the Isometric view, pull down the Windows menu and choose the "Fit Views to All" command. Vision3d responds by re-drawing the goblet to fit comfortably within the Isometric view.

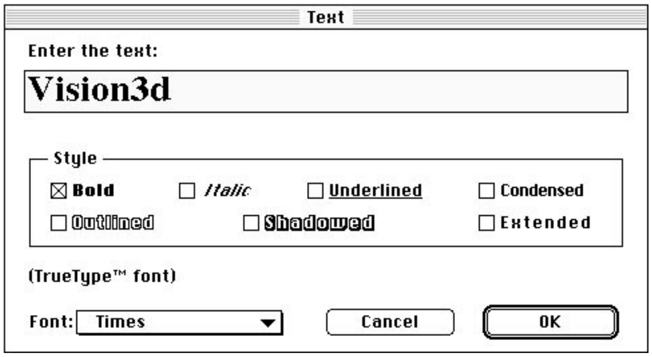
From this exercise you can see that the lathe modeler creates 3-D objects by revolving a 2-D template, or cross-section, around a predefined axis. The resulting 3-D object is given surface geometry based on the 2-D template used.

Exercise 2: Extruding text

In this exercise you will create 3-D text using the Text tool and the Extrude modeler. You will then understand how to use this tool and modeler, as well as learn how to scale an object larger or smaller.

1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Select the Text tool from the tool palette.** Notice how the cursor changes when you then move it into the model window with this tool selected.
3. **Click once in the model window.** Vision3d responds by opening the Text dialog window.

   The Text dialog box provides fields for entering the text, choosing the desired font, and adding other type effects, such as italic or bold. Note that there is no point size setting. Point size is not a relevant term in the context of a Vision3d model. That is because the text characters are extracted into 2-D objects that have scale within the model space.



Text

Enter the text:

Vision3d

Style

☒ Bold □ *Italic* □ Underlined □ Condensed

□ Outlined □ Shadowed □ Extended

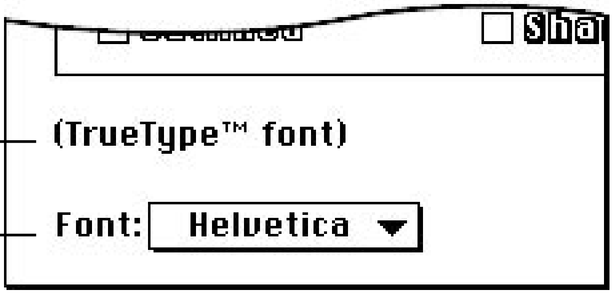
(TrueType™ font)

Font: Times

Please refer to the Text tool discussion later in this chapter.

4. **Type any word or phrase into the "Enter The Text" box.** While the field scrolls to make additional room for a long phrase or sentence, it is recommended that you limit the number of characters entered to a single word or phrase.

5. **Choose a font and Style for your text.** The type of font you choose — whether Postscript or TrueType (if using System 7.0 or later) — is indicated in the dialog box.



The type of font (Postscript, TrueType or bitmapped) is indicated here.

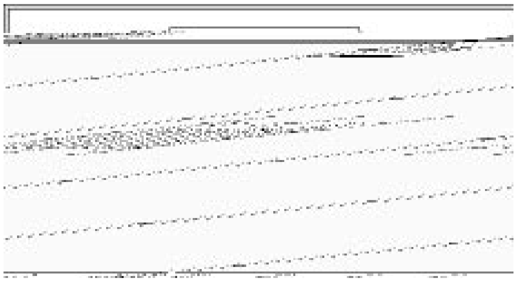
(TrueType™ font)

Choose a font using the pop-up menu.

Font: Helvetica

The Text tool supports any installed font, including bitmapped, PostScript® and TrueType® fonts. Fonts are installed in the System Folder. Bitmapped or screen fonts are usually the least desirable. TrueType fonts are smoothed by a mechanism built into System 7. Of course, PostScript fonts must have a System extension called ATM (Adobe Type Manager™) installed to use them with the highest accuracy.

6. **Click on the OK button.** Vision3d responds by extracting your entered text and drawing objects representing those characters in the model window.

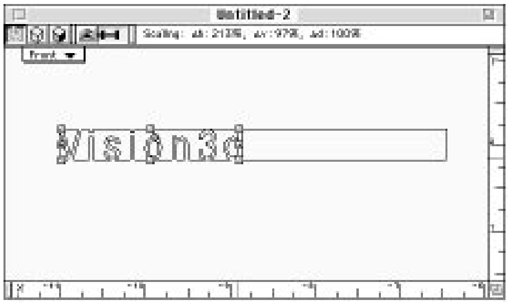


Notice your text-objects are 2-D only. Also notice that the objects are grouped as a single entity for ease of moving, scaling and maintaining their spatial relationship.

If these objects were still in font format, their spatial relationship would be referred to as "kerning." However, the font kerning information is not available, since these objects are no longer connected to the font that created their geometry…they are now 2-D objects in the geometry of characters. This is an important distinction because it changes the way you can work with the objects.

The issue of "kerning" text-shaped objects is not directly addressed in this tutorial. For some background on this area: In Vision3d's model space, kerning of text objects is accomplished by ungrouping the objects and moving them as desired, then re-grouping again. Please refer to the discussion of the Group and Ungroup commands later in this chapter, if needed.

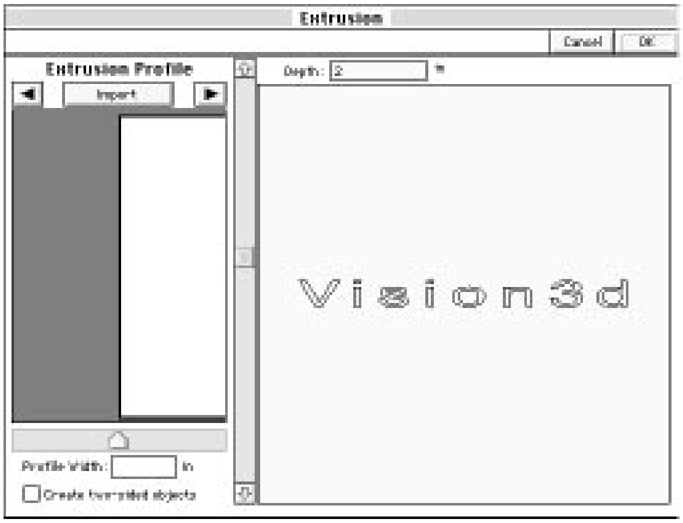
7. **With your text objects still selected, choose the Object Scale tool from the Tool palette.** This tool allows you to scale your text, making it larger or smaller. Notice how the cursor changes when you move it into the model window with this tool selected.
8. **Scale your text larger.** Clicking and dragging on the right or left center Selection handles will re-scale an object on the X axis. Clicking and dragging on the top or bottom center handles will re-scale the object on the Y axis. Clicking and dragging on any corner handle re-scales an object on both the X and Y axes simultaneously. Holding down the Shift key while dragging on a corner handle constrains object proportions during the re-sizing.


Untitled-2

Scaling: ah: 213%, av: 97%, ad: 100%

Re-scale your text to some pleasing proportions and size in the view; the actual scale is not important for this exercise. If necessary, use the "Fit Views to All" command afterward, as you did in Exercise 1.

9. **With the text still selected, pull down the Modeling menu and choose the "Extrude" command.** Vision3d responds by opening the Extrusion dialog window.


Extrusion

Depth: 2 "

Profile Width: in

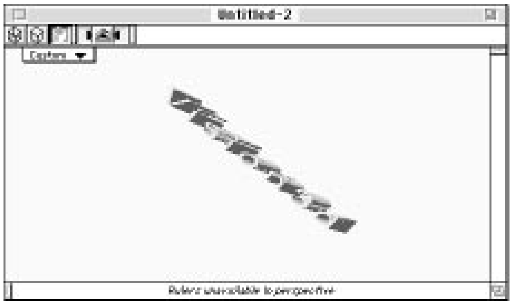
□ Create two-sided objects

Note that the extrusion profile on the left of the dialog shows a straight geometry. This exercise does not address the use of other profiles. Please refer to the Extrude modeler discussion later in this chapter for information on this extrude option.

10. **Set the Extrude depth.** Move the slider at the left of the extrusion window up or down until the extrusion depth is set at about 2 inches. You can see the extrusion depth in the Depth feedback field at the top of the window.

Alternately, you can enter the extrude depth value directly in this field. If you need as exact extrude depth, this is the preferred method.

11. **Click on the OK button at the top right corner of the dialog window.** Vision3d responds by displaying a progress bar as the extrusion operation occurs. When the extrusion is complete, Vision3d draws your extruded text in the model window. Notice this 3-D text has replaced the original 2-D text.
12. **If desired, change the view direction and perspective to get a better look at your creation.** Choose "Isometric" from the view pull-down menu at the top left of the model window. You may also want set the lens control to Normal and the display method to Shaded for a more natural appearance. Vision3d responds by re-drawing your extruded text from the new viewing position with a vanishing point perspective.



If the extruded text is too small to be seen clearly in the Isometric view, use the "Fit Views to All" command again, as you did in Exercise 1. Vision3d responds by re-drawing the extruded text to fit within the Isometric view.

From this exercise, you now know that the Extrude modeler creates 3-D objects by pushing a 2-D template, or outline, through space to give it depth.

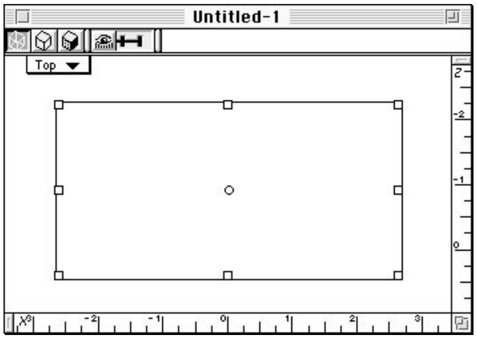
Exercise 3: Warping a flat rectangle

In this exercise, you will deform a 2-D rectangle, i. e. — change its geometry, using the 3d Sculpter modeler to move a vertex point in 3-D space. By simply editing this vertex, you will turn a flat, planar 2-D polygon into a three-dimensional object.

Further, you will understand how to use the filled rectangle drawing tool and learn the fundamental operation of the 3d Sculpter modeler.

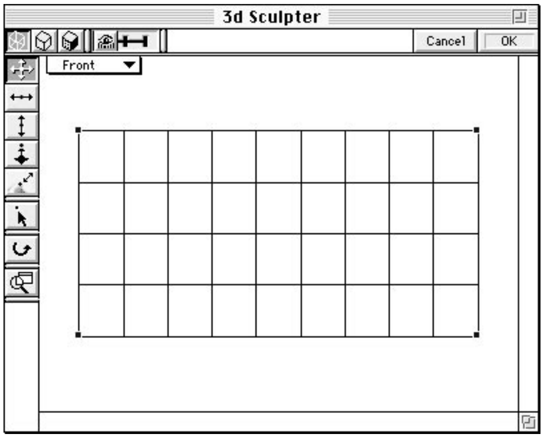
1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.

2. **Change the view orientation to Top.** Pull down the view title menu at the top left of the model window and choose Top.
3. **Choose the filled rectangle drawing tool from the Tool palette.** Notice how the cursor changes when you move it into the model window with this tool selected.
4. **Click in the model window and drag a marquee to draw a rectangle.** A 2-D rectangle is drawn in the model window.



Remember, you're looking at the rectangle from the Top view. If you switch to a Front, Back, or side view you can see that the rectangle has no depth.

5. **In the Top view, with the rectangle selected, pull down the Modeling menu and choose the "3d Sculpter" command.** Vision3d responds by opening the 3d Sculpter dialog window.

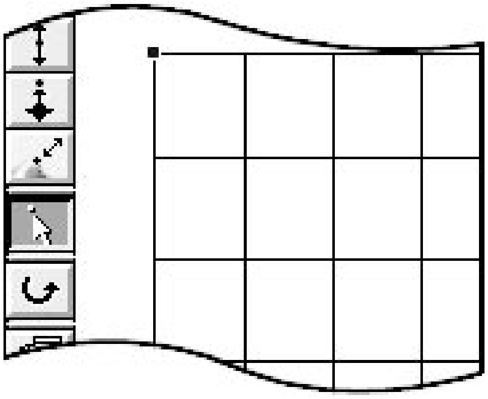


Notice this dialog window has its own display method, lens position and lens controls that work just as they do in the model's main window(s). These controls affect the 3d Sculpter window only and do not change the display method, lens position and view direction in the model window.

The 3d Sculpter dialog window opens in the Front view by default, yet you are seeing the rectangle from the top, not the side. In the model, your rectangle is oriented to the Top view as you drew it by applying a 90° rotation internally. In the 3d Sculpter modeler, as with the other modelers, the rotation transformation is temporarily removed, so that you see the "pure" object. When it is returned to the model after editing, the rotation will be re-applied again to orient it as you drew it.

For this exercise, you will work in both the Front and Top views while in the 3d Sculpter modeler.

6. **Choose the Add Vertex tool from the 3d Sculpter tool palette.** Notice how the cursor changes when you move it into the 3d Sculpter dialog window with this tool selected.

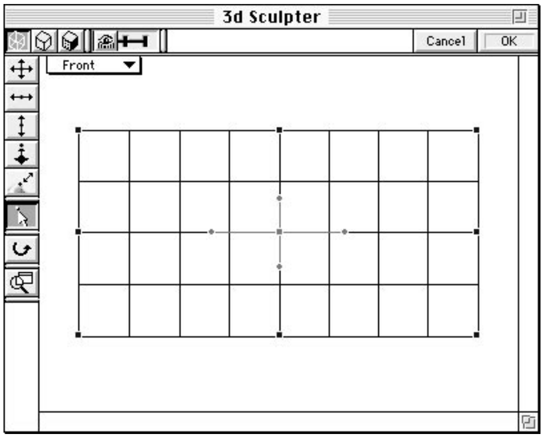


Choose the "Add Vertex" tool.

A 2-D rectangle is the simplest of all bézier objects. It is made of a single bézier surface patch. Note the four vertex points, one at each corner of the surface. In more complex objects, multiple surfaces are used to construct its geometry, yet each patch has the same four vertices at each corner of each surface patch.

This is an important point to remember when editing more complex bézier objects in the 3d Sculpter modeler. It can help you find your way around a multi-patch bézier surface.

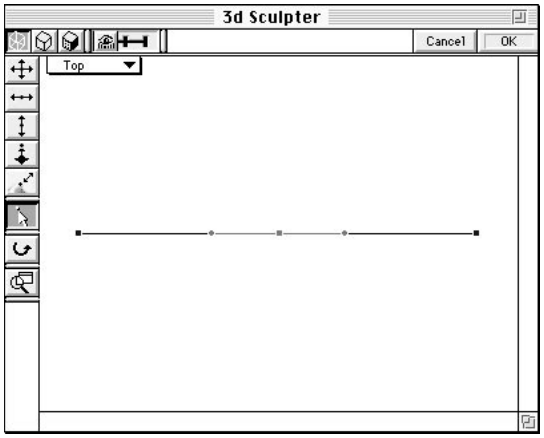
7. **Click once near the center of the rectangle.** The 3d Sculpter modeler responds by placing a new vertex point at the location on the surface that you clicked.



The original surface patch is sub-divided into four. Again, each patch has a vertex point at its four corners. But this time, the vertex point in the middle is shared by all four patches. You can tell this is the case, because when the point is selected, four direction handles appear.

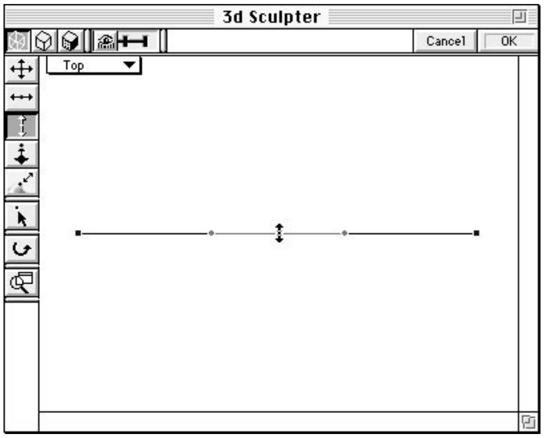
NOTE: *If you're familiar with Adobe Illustrator® or Aldus® Freehand™, two highly respected 2-D drawing applications, you have already seen the use of direction handles to control bézier splines. The vertex points and direction handles visible on objects in 3d Sculpter perform the same functions, only in three dimensions instead of just two.*

8. **With the center vertex point selected, change the 3d Sculpter view to Top.** Pull down the view title menu in the 3d Sculpter dialog window and choose Top, just as in the model window. The 3d Sculpter dialog window is re-drawn from the new view orientation.



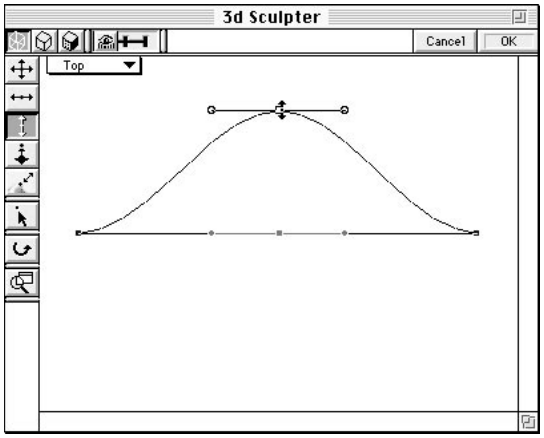
You are now seeing the rectangle from its side. Since the object is two dimensional, all you see is a thin horizontal line in the view. But you should also see the selected vertex point in the center.

9. **Choose the Vertical Move tool from the 3d Sculpter tool palette.** This is the third tool down on the Tool palette. Its icon shows two arrows facing up and down.

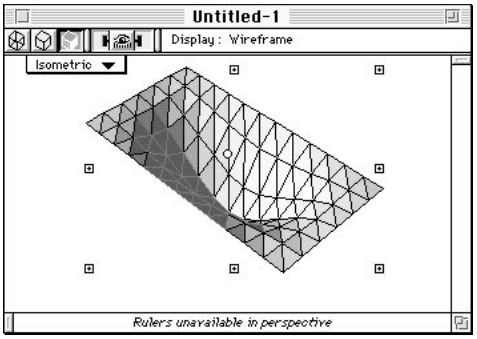


This tool is constrained to move the selected vertex point(s) vertically ONLY. This is relative to the view, of course. While the first tool in the palette can also move a vertex point vertically, it is not constrained unless you hold down the Shift key.

10. **With the cursor positioned on the center vertex point, click and drag the vertex point straight up toward the top of the window.** Notice how the thin line is warped from straight to a shape resembling a bell curve.



11. **Click on the OK button in the top right corner of the 3d Sculpter dialog window.** Vision3d responds by returning the warped rectangle — now considered a 3-D object — to the model in place of the flat 2-D rectangle.
12. **If desired, change the view position and perspective to get a better look at your creation.** Choose “Isometric” from the view title pull-down menu at the top left of the model window. Also, switch the lens control to Normal and the display method to Shaded.



Display : Wireframe

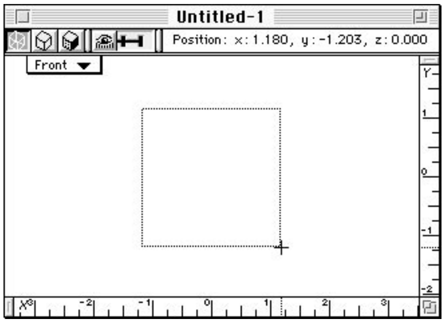
Vision3d responds by re-drawing your warped rectangle from the new view orientation and lens angle. If the warped rectangle is too small to see clearly in the Isometric view, use the “Fit Views to All” command as you did in Exercise 1. Vision3d responds by re-drawing the warped rectangle to fit within the Isometric view.

In this exercise, you learned that the 3d Sculpter modeler creates new 3-D objects by deforming or changing the shape of existing 2-D and 3-D objects. The 3d Sculpter does this by allowing the user to push or pull vertex points on either bézier or polygonal objects.

Exercise 4: Making a Skin object

In this exercise, you will use the Filled Rectangle tool and the Skin modeler to build a section of duct. You will learn how to use the tool and the Skin modeler, how to reposition objects in the model window and how to duplicate and replicate objects.

1. **Open a new model.** Again, make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Choose the Filled Rectangle tool from the Tool palette.** Notice how the cursor changes when you move it into the model window.
3. **In the model window, draw a filled rectangle.** Click and drag until the marquee is about the same proportions shown in the illustration on the next page.



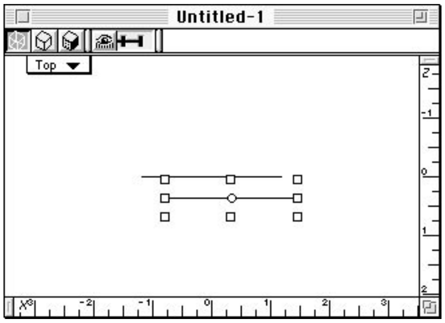
Position: x: 1.180, y: -1.203, z: 0.000

Releasing the mouse button draws the object in the model. Vision3d responds by drawing a filled rectangle in the model window.

You are going to "clone" this rectangle, then use the multiple rectangles as cross-sections for the final object. These cross-sections will function as ribs, over which we will stretch a "skin," or polygonal surface. That's why the modeler you will use is called the Skin modeler.

4. **Change the View orientation to Top view.** Notice in the Top view how the rectangle appears as a line with no thickness, because it is a 2-D object.

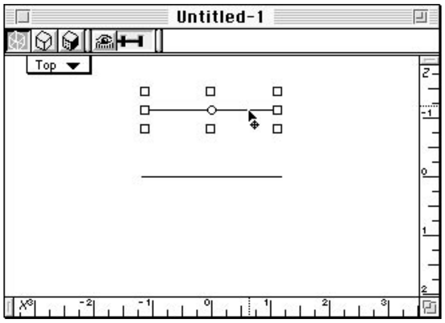
5. **With the rectangle selected, pull down the Edit menu and choose the "Duplicate" command.** Vision3d responds by drawing another rectangle identical to the first, with a slight offset.



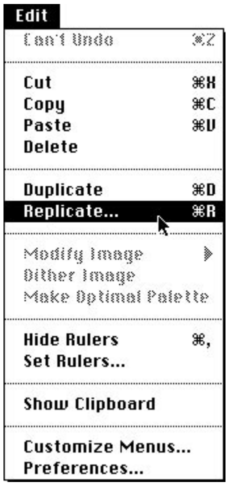
Notice the new rectangle is now selected. Being a copy of the original, this second rectangle also appears as a line in the Top view. You may want to view the two rectangles from the Front and either the Left or Right views, just to better understand what is happening in the model.

6. **In the Top view, move the selected rectangle upward on the screen.** Choose the View Move tool from the Tool palette if not already chosen.

Click on the top center selection handle and drag the object up. Position it in the model window as shown in the following illustration.

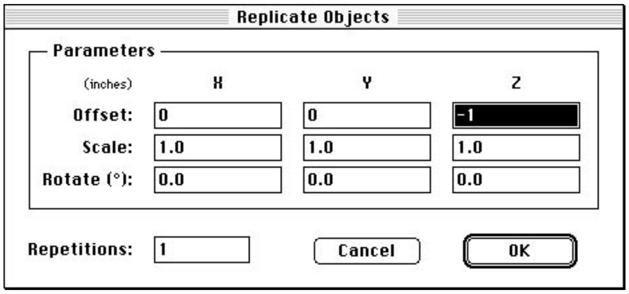


7. **With the upper rectangle selected, pull down the Edit menu and choose the "Replicate" command.** Vision3d responds by opening the Replicate dialog window.



Replicate is a fully-optioned version of Duplicate. Where the Duplicate command creates one object with a fixed spatial relationship to the original, Replicate allows you to specify the transformation you want applied to the duplicated objects, as well as how many objects you want to create. For more information, please refer to the discussions on the Duplicate and Replicate commands later in this chapter.

8. **Enter values in the Replicate dialog.** Match the settings in the Replicate dialog window with the settings shown in the illustration on the following page.



Replicate Objects

Parameters

(inches)	X	Y	Z
Offset:	0	0	-1
Scale:	1.0	1.0	1.0
Rotate (°):	0.0	0.0	0.0

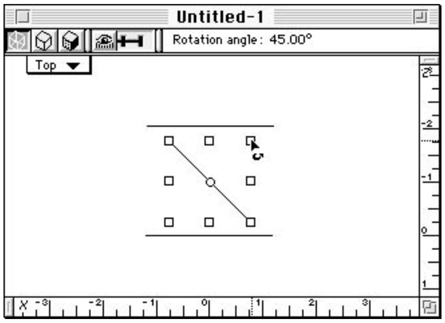
Repetitions: 1

9. **Click on the OK button.** Vision3d responds by drawing new objects according to the specifications you entered in the Replicate dialog window.

   In this case, you asked that it create just one more new rectangle identical in size and orientation to the first two, but offset downward one unit, whatever unit you are using in the model. As before, this third rectangle appears as a line in the Top view. You may need to use the fit Views to All command in the Windows menu to bring all three objects into view.

   Notice how both the replicated object and its original are selected in the model window.

10. **Rotate the center cross-section.** Choose the Object Rotation tool from the Tool palette. Select the center rectangle. Holding down the Shift key, click on any corner selection handle and drag to rotate the object 45°.

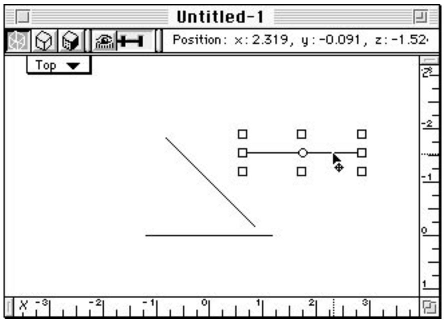


Untitled-1

Rotation angle : 45.00°

   Note that a +45° rotation is different from a -45° rotation. Holding down the Shift key during an object rotation constrains the rotate operation in 45-degree increments. Watch the feedback area at the top of the model window to see when you've rotated the rectangle +45 degrees, then release the mouse button. Vision3d responds by re-drawing the cross-section with a 45° rotation.

11. **Reposition the top cross-section.** Choose the Pointer tool from the Tool palette. Select the top cross section. Click on the center right selection handle and drag the cross section to reposition it as shown in the following illustration.

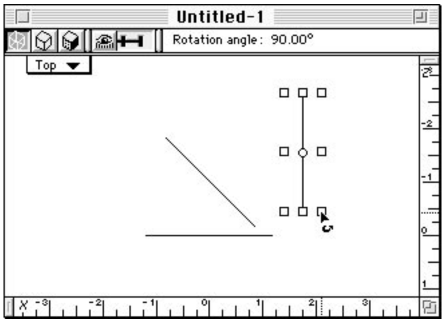


Untitled-1

Position: x: 2.319, y: -0.091, z: -1.52

Top

12. **Rotate the top cross-section.** Choose the Object rotate tool from the Tool palette again and drag a corner handle on the top cross-section to rotate the object vertically.



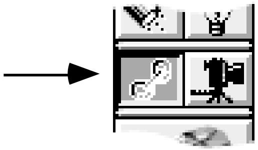
Untitled-1

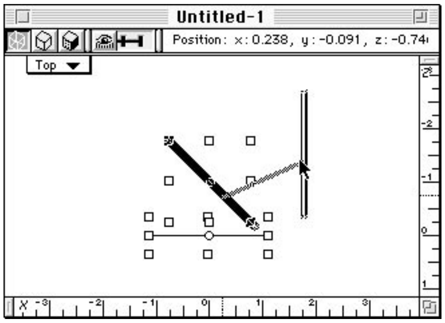
Rotation angle: 90.00°

Top

Holding down the Shift key, click on any corner selection handle and drag to rotate the object +90°. In this case, +90° looks the same as -90°. Make sure you rotate it a +90°. This is to insure that the beginning vertex point on the three ribs are all in the same relative position on each rib. The importance of this detail is explained in the discussion of the Skin modeler extension later in this chapter.

13. **Choose the Link tool from the Tool palette.** This tool allows you to specify which 2-D objects are to be used as ribs for skinning, as well as the sequence in which they will be skinned.

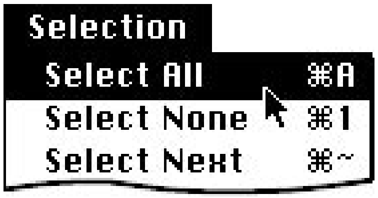




14. **Link the three cross-sections.** Starting at the bottom, click on the lowest cross-section, then drag until the cursor is on the 45° cross-section and release the mouse button. As you click on the lowest cross-section, it becomes highlighted. As you drag, a line appears between the objects that are being linked. When the cursor is positioned over the 45° cross-section, that object will also be highlighted. When you release the mouse button, those two objects are linked.

    Repeat the linking operation, this time clicking first on the 45° cross-section, and linking it to the 90° cross-section. The three cross-sections are now linked in order.

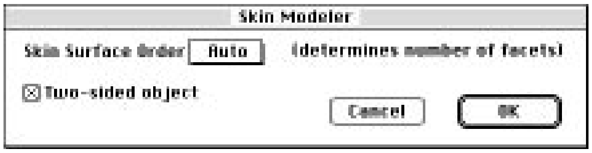
15. **Select the cross-sections.** Pull down the Selection menu and choose the "Select All" command. If you have a 14-inch monitor or smaller, the Selection menu is condensed into the Edit menu to conserve screen real estate. So, if you have a 14-inch or smaller monitor, pull down the Edit menu and choose the Select All command from the Selection pop-out menu.

    Alternately, you can use the keystroke equivalent of Command-A (⌘A). Another method of selecting all the ribs is to drag a selection marquee around them.

16. **Skin the cross-sections.** Pull down the Modeling menu and choose the Skin command from the Skinning sub-menu.

    Vision3d responds by opening the Skin Modeler dialog box. Set the skinning parameters as shown in the illustration on the next page.

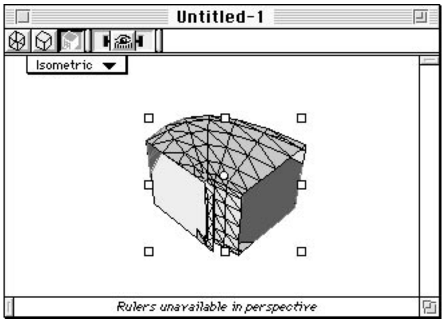


Skin Surface Order: Auto (determines number of facets)

☒ Two-sided object

Clicking on the OK button causes a skin to be lofted over the linked cross-sections. Notice that because the ribs were filled, the ends of the skinned object are closed and the object appears solid.

17. **If desired, change the view angle to get a better look at your creation.** As in the other exercises, choose Isometric from the view title pull-down menu at the top left of the model window. Also switch the lens control to Normal and the display method to Shaded.



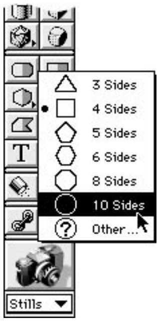
Your object is displayed from the new view angle. Use the "Fit Views to All" command as you did in Exercise 1. Vision3d responds by re-drawing the object to fit comfortably within the Isometric view.

In this exercise you learned how the Skin modeler creates 3-D objects by generating a surface or skin over a series of 2-D ribs, or cross-sections, that are defined and linked in a specific order by the user.

Exercise 5: Making a coil spring

In this exercise you will learn how to use the Sweep modeler to lathe and extrude a 2-D circle at the same time, creating a coiled object that looks like a spring. The Sweep modeler is useful for making objects that are spiral in form, such as springs and threads on bolts.

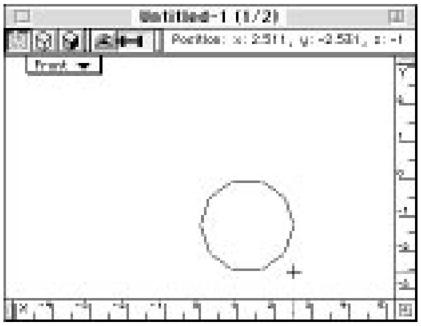
1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Choose the filled Regular Polygon tool from the Tool palette.** Hold down the mouse button so the tool's sub-menu appears.



Choose the 10-sided option from the sub-menu. You will use this tool to create a circle with only ten sides. This saves significant memory over the full Circle tool, and it works just as well for most purposes.

Notice how the cursor changes form when you move it into the active model window.

3. **Draw a circle.** This time position your cursor at the 0, 0 coordinates on both the X and Y axis rulers before you start drawing.



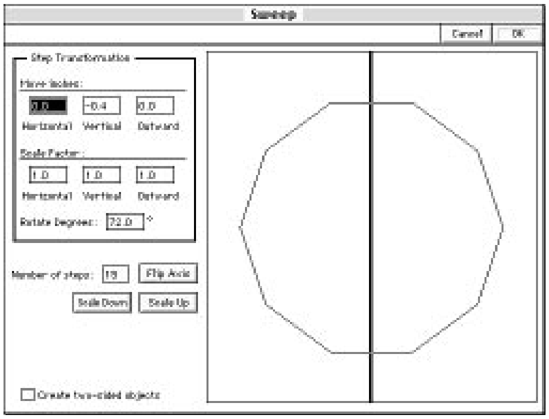
Position: x: 2.511, y: -2.531, z: -1

Holding down the Shift key, click and drag a marquee of EXACTLY 2.5 inches in diameter. (Inches is the default units. If you have changed them to something else, draw an equivalently-sized object.) By starting at 0, 0, you can easily see the dimensions on the rulers as you draw.

The Shift key constrains all 2-D drawing tools, so objects are drawn equally proportioned on all sides. When you release the mouse button, a perfectly proportioned object is drawn in the model window.

4. **With the circle selected, choose the Sweep command from the Swept Surfaces sub-menu, in the Modeling menu.** Alternately, you can click on the Button bar's Sweep button.

Vision3d responds by opening the Sweep dialog window. The Sweep modeler performs a combination of lathing and extruding on the same object.



Sweep

Cancel OK

Step Transformation

Move inches:

0.0 Horizontal · -0.4 Vertical · 0.0 Outward

Scale Factor:

1.0 Horizontal · 1.0 Vertical · 1.0 Outward

Rotate Degrees: 72.0 °

Number of steps: 18

□ Create two-sided objects

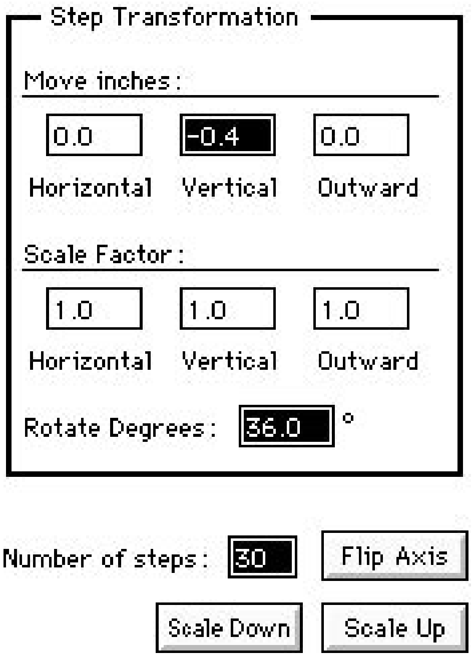
Where the Lathe and Extrude modelers have fixed axes, the Sweep modeler's axis can be moved dynamically during the sweep operation. For more information on the fields in this dialog, please refer to the Sweep modeler discussion later in this chapter.

5. **Enter the sweep parameters.** For this exercise, you can just match the settings in the Sweep dialog window with the numbers shown in the following illustration.

Step Transformation numbers determine what happens for each step.

Set the numbers as shown here. Note that the highlighted numbers are the ones doing all the work.

This field sets how many times the Step Transformations will be done.



Step Transformation

Move inches:

0.0 Horizontal · -0.4 Vertical · 0.0 Outward

Scale Factor:

1.0 Horizontal · 1.0 Vertical · 1.0 Outward

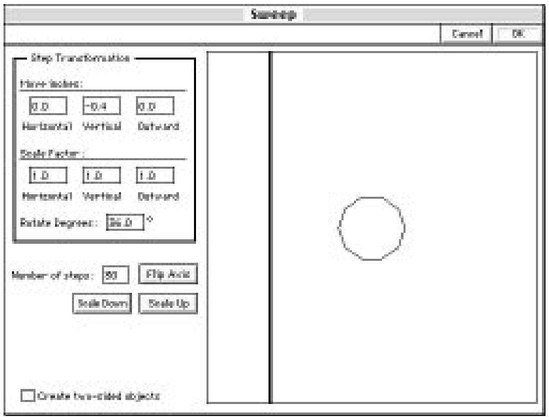
Rotate Degrees: 36.0 °

Number of steps: 30

The fields in the Step Transformation box at the top left of the dialog allow you to type in numerical values for movement and scale of the 2-D object on the X (horizontal), Y (vertical) and Z (depth) axes. There is also provision for degrees of object rotation. The number of steps determines how many times the step specified in these fields are performed. These operations are called transformations. During each step, the transformations specified in the Step Transformation fields are repeated and the object is constructed…step-by-step.

6. **Position the sweep axis.** On the right side of the dialog is the template window. Your 2-D object appears here, along with the sweep axis, which is the heavy black line. This axis is positioned in the same manner as you did in the Lathe modeler exercise. Click twice on the Scale Down button to provide enough room to offset the Sweep axis.

   Use the following illustration as a guide for positioning it relative to the object.



Sweep

Step Transformation

Move Inches:

Horizontal	Vertical	Outward
0.0	-0.4	0.0

Scale Factor:

Horizontal	Vertical	Outward
1.0	1.0	1.0

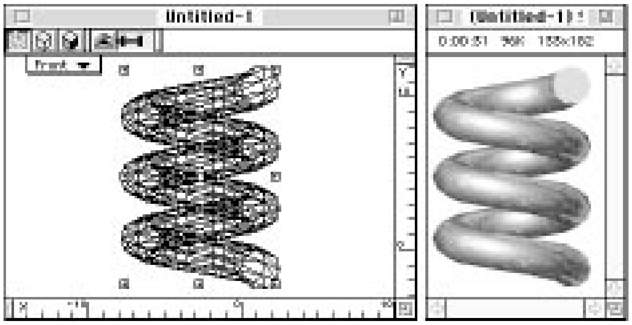
Rotate Degrees: 36.0 °

Number of steps: 50

□ Create two-sided objects

   If necessary, use the Flip Axis button to change the axis orientation from horizontal to vertical. Then, position the cursor anywhere on the axis. Notice how the cursor changes to a hand icon, indicating the Grabber tool is active. With the hand icon visible, click and drag to move the axis into position.

7. **Click on the OK button.** A progress bar is displayed as it generates the swept object. Because the Sweep modeler is more complex than the other modelers you used in these quick tutorials, and you've asked it to make a large number of transformations as the surface is created, it may take a few moments to generate the object.



When the sweep operation is complete, the swept object is drawn in the model window. If necessary, use the "Fit Views to All" command to scale the view so your swept object will fit comfortably in the window.

NOTE: *A Sweep object is given a smoothed surface that normally renders without faceting. It has a fixed breaking angle of 30° by default. (Breaking angles are explained in the Modeling Extensions section, later in this chapter.) If you find the surface appears faceted when you render the spring, you can use the Smooth command from the Resurfacing sub-menu of the Modeling menu. You can set the breaking angle to anything between 0° and 180°. With the proper breaking angle for the surface, the swept object will render with a smooth surface.*

In this exercise you learned how the Sweep modeler can twist, scale and rotate a 2-D object on all three axes, based on the settings and sweep axis that you define. When fully mastered, this modeler can become a powerful addition to your modeling repertoire.

Corkscrews, bolts with threads, nautilus and conch sea shells, DNA strands, and coiled objects of all types are examples of the kind of objects this modeler is typically used to create.

Modeling Methodology

Here are some main points you should remember when you are up to your elbows in polygons, polygonal meshes, and bézier surface patches:

1. **You should use orthographic views for modeling and perspective views for rendering.** During the modeling phase, perspective can get in the way and fool you about the true size and position of objects. Orthographic views allow you to see absolute size and position without the distraction of perspective. Once the model is assembled, perspective is very important for realism. Typically, you would use

orthographic views in the modeling windows for modeling and perspective views in the camera windows for renderings. This is why camera windows have a continuously variable focal length control, while modeling windows provide only a 3-position control.

2. **You should try to use primitives and instantiate shapes as much as possible for memory efficiency.** As discussed elsewhere in this manual, primitives are mathematically-described entities requiring less memory to describe. They also render faster. They are the most desirable for modeling whenever you have the option. Even many of the complex objects about us in the real world are assembled from smaller, simpler object of basic geometry. The question becomes one of whether it is better to use several primitives to create the desired shape or one polygonally-defined object. There is no absolute trade-off point where one method works better than another…you must develop a "feel" for what works best in your approach to a modeling task.
3. **You should use surface maps for surface detail rather than try to explicitly model the geometry.** Objects are not usually as efficient as using surface maps. In addition to the additional modeling time required, the rendering time increases with the number of polygons or bézier patches added to the model. Using surface maps has less negative impact on rendering speeds.
4. **You should think about what rendering effects you really need and which ones you can do without.** Eliminating some less-important options can significantly improve rendering speeds. Often, the first response is to enable ALL the options because you want the image to look good! But, after it's rendered, you notice that there wasn't any refraction in the model or that the shadows aren't actually noticeable from the viewing position you chose for the rendering.
5. **You should consider lower resolution and less detail for animations.** The eye is very forgiving of objects in motion. You can often get away with screen resolution for an animation because nothing stays in one place long enough to notice. You may not even need to anti-alias the images.

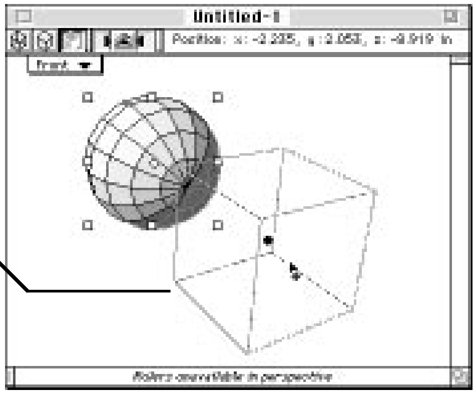
Transformations and bounding boxes

As you construct models, the need to move, rotate or scale objects inevitably arises. The process of performing one of these operations is referred to as applying a transformation. Each object or group (of objects) has a set of transformation parameters stored with the object's description, even if they are all set to zero (which would mean the object is positioned at the exact center of the model space, aligned to the X, Y, and Z axes and with a size of one Strata Unit).

NOTE: *Vision3d actually maintains an internal "history" of transformations on each object in a model's database. That's why scaling an object before rotating it produces a different result than rotating first, then scaling. The Transform dialog shows only the last set of transformations applied.*

You may interactively move, rotate and/or scale objects, i.e. — apply a transformation, using the object manipulation tools on the tool palette, or you may use the Transform command in the Modeling menu to enter the transformation values directly.

When using the object manipulation tools, objects being moved, rotated or scaled are represented by bounding boxes during the actual tool use. Then, once the operation is finished, the modeling windows redraw to show the changes to the object(s) involved. Bounding boxes are used instead of the actual object for on-screen speed and responsiveness to the tools.


Untitled-1
Position: x: -2.205, y: 2.053, z: -8.919 in
Front
Rulers unavailable in perspective

The manipulation of objects is represented by bounding boxes

This improves response to tool operations in the modeling windows

When using the Transformation command, the object simply redraws to show its new parameters when the OK button is clicked. Numeric control is sometimes desirable when you know exactly the values to enter and accuracy is paramount. However no visual feedback is provided in this method and that can present its own limitations.

If the results are not as expected using the Transform dialog, you can make further adjustments or use the Undo command and try again.

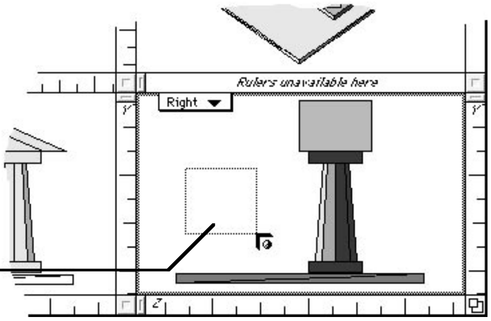
Merging models and adjusting for scale differences

Another very useful feature is the ability to merge data from several sources into one model. This is accomplished through the Import command in the File menu. As explained in chapter 2, this command may be used to merge two Vision3d models, i.e. — bring the objects in the second model into the first model, as well as to import 3-D data from other applications into an open Vision3d model.

When the Import command is called, a directory dialog is opened to find the file you wish to merge with the currently open model. As with reading files through the Open and Open As commands, the particular file formats supported by the Import command depend on the extensions available to Vision3d at launch of the application for the current session. Thus, only those file formats that Vision3d recognizes will appear in the directory as you move from folder to folder.

Once the desired file has been found and read into Vision3d, the second file's object data is inserted into the current model by clicking or dragging the cursor in the active view.

Other 3-D objects can be merged using the Import command



Once the file has been read, click or drag the cursor in the active view to insert into model

Clicking once in the view inserts the imported objects at the scale they were created in the other model. That scale will be converted to the units currently being used in Vision3d, so that you should see the objects represented in their accurate scale.

Dragging the cursor allows you to re-size the objects relative to the scale of objects in the current model. In this manner, you can override the scale used during their creation and easily match them to the scale being used in the current model. When objects are re-sized during importing, any textures applied are also scaled to maintain the same relative appearance on the objects' surfaces.

You may further adjust the scale of imported objects after insertion into the model, as well as their textures, if needed, with the scale tool, Transform command and/or texture-editing commands.

Fundamental elements of construction

Throughout this manual, several terms or phrases unique to Vision3d's operation will be used. A Glossary of Terms is included in the appendices at the back of this manual; it is assumed you are familiar with them. Some of these terms are presented in the discussion here in an effort to lay a foundation for understanding the fundamentals of 3-D objects and how Vision3d keeps track of and manipulates objects to build your models.

The creation and manipulation of any entity, whether simple or complex, single or multiple, is termed modeling. Models can be defined graphically or as mathematical equations. Typically, graphically-described models are more meaningful. Graphical models are often referred to as geometric models, since the components are represented with geometric entities such as primitives, polygons, lines, circles, curves, etc.

You are concerned here only with geometric modeling. Thus, the term model means a computer-generated, geometric representation of a recognizable object. The tools provided in Vision3d empower you with the ability to graphically and interactively model these geometric models to any level of detail you may wish to pursue.

The basic set of Vision3d objects are called primitives. Primitives are mathematically defined; making them the most efficient of any object.

Complex Models are constructed using various combinations of these primitives. Objects requiring geometry not provided by the primitives can be manufactured by using a primitive object (usually 2-D) as a template and employing the special talents of a modeler to construct the 3-D geometry of the desired object. Depending on the modeler chosen, the object is constructed of grouped polygons, a polygonal mesh, or a bézier surface.

Lathe, Extrude, and Sweep are examples of modelers that make objects with of a polygonal mesh for the surface. These modelers start with the most basic unit of all — the polygon, and generate composite groupings of polygons, oriented as needed in 3-D space, to define any geometric shape conceived. Polygonal surfaces appear smooth when rendered because of the surface "normals" attached during construction.

3d Sculpter is a modeler that uses bézier surfaces as its primary construction method, although it can deal with both types of objects. Bézier objects are displayed in the modeling views as polygonal representations (thus appearing similar to polygonal objects). However, they can be stored internally with fewer data points and are generally more efficient to handle and render. The basic element of bézier objects is called the surface patch.

An explanation of how polygonal and bézier objects are handled differently in Vision3d is given in the next topic of this chapter.

The vertex point — polygon building blocks

A vertex is a point in 3-D space. Everything polygonal in nature builds from this single, simple point in space…in the following logical progression:

- **Each vertex** is linked to the next vertex by one edge of a polygon,
- **Three or more vertex points** (called vertices) define a closed polygon,
- **A polygon** is a geometric building block for defining a surface,
- **Multiple polygons** may be positioned, oriented, and grouped to form objects,
- **Objects** may be grouped together to make more complex objects,
- And finally, **objects** may be assembled to create scenes.

One important point to remember is that the number of vertices in a polygon determines how much memory is required to store it. A single four-vertex polygon takes a fixed amount of memory to store and represent. It doesn’t matter how large or how small the polygon, the same amount of memory is used. Thus if an object is defined by 100 four-sided polygons and another by 1,000 four-sided polygons, the second object will take ten times more memory to store in a model than the first…even if they are the same size.

A minimum of three vertices are required to define a polygon. The vertices of a polygon may be convex or concave, depending on the position of the vertices on either side of it. Polygons may have holes in their surface, or their edges may even double back in such a way that the surface intersects itself.

Since each vertex of a polygon may lie anywhere in 3-D space, a polygon can be planar or non-planar as well. In a planar polygon, all vertices lie on the same plane in space. Vision3d does not support non-planar polygons.

The modeling tools in Vision3d will never generate a non-planar polygon. If imported files contain non-planar polygons, the import extensions attempt first to construct triangulated mesh equivalents of these polygons. Failing that — possibly because the polygons are too complex — they are selectively discarded, leaving holes in surfaces where the offending polygons were located.

Discarding non-planar polygons is a very rare occurrence. If you find a particular application consistently producing geometric data that Vision3d is not handling properly, you should review the modeling methods used in that program to determine if a different construction technique would be more successful. You may also want to contact Strata Incorporated technical support to see if an improved import extension is available that meets your needs, or at least make them aware of the problem for future enhancements.

Smoothing curved surfaces (from flat polygons)

The basic problem with polygons is that they are flat! It is easy for the renderers to handle planar polygons, but that convenience is not without cost. Standard polygonal objects render with surfaces that appear faceted.

Faceting is most noticeable on curved surfaces. It becomes a problem because the orientation of a polygon directly affects its brightness as it faces toward or away from the light source. As the orientation of those flat polygons change across a curved area, the shading of each polygon changes in the same flat, abrupt manner.

There are basically two ways to achieve smooth-shaded surfaces in polygon-based models:

- **Represent the object with finer polygons that do not show the faceting as coarsely.**
- **Attach "normals" to vertex points and interpolate the shading across the surface.**

Vision3d can use either method, although it is obvious the second approach is most efficient. Some imported files, however, will not give you the choice, because of the way the polygonal surface is defined.

Using finer polygons

Many CAD applications use this method of surface definition. Importing files from CAD programs is the most likely to produce objects in Vision3d with very fine polygons. Vision3d has no direct method for forcing the polygonal fineness of an object's surface, although there are some techniques that can influence the construction one way or another.

Subdividing into finer, smaller polygons is an inefficient approach to smoother surfaces. Because the faceting is finer, the surface appears smoother. If the polygons are small enough, the faceting detail will be so fine that the surface will appear smooth when rendered.

But how smooth is smooth enough? An object that appears smooth on a 72 d.p.i. monitor will not necessarily print smooth to a laser printer, Imagesetter, or film recorder. Compounding the problem is that each level of polygon refinement in the surface exponentially increases the number of polygons required to represent an object; requiring more and more memory and taking longer and longer to render.

Attaching normals

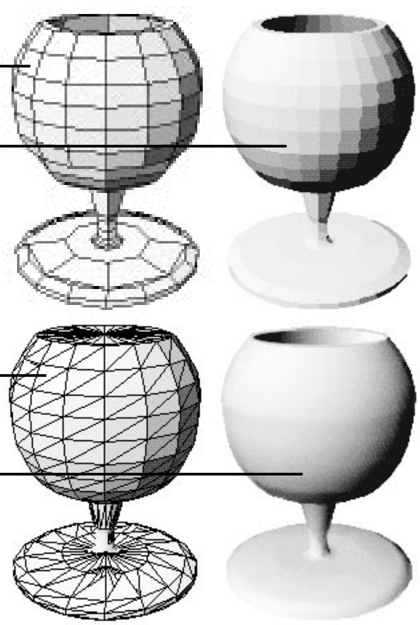
The orientation of a standard polygon is represented in Vision3d by a single vector pointing outward perpendicular to its surface. Renderers use this information to render the surface in all its faceted glory.

Yet smoothing curved polygonal surfaces is relatively easy given the right conditions. A visually smooth surface can be obtained by associating the normals for each polygon making up a curved surface with the vertices describing those polygons, then interpolating the shading calculation across all each polygon and averaging the polygon-by-polygon variations across the surface. This is the rendering method used by Vision3d's Gouraud and Phong renderers. The Raytracer, RayPainting, and Raydiosity algorithms also employ a similar, but different, surface-smoothing algorithm in their calculations.

This smoothing technique can reduce the shading anomalies of objects using polygonal representation. However, not all polygonal surfaces have normals of the type the renderers can use. Vision3d supports two types of polygonal surfaces:

- **Standard**
- **Triangulated Mesh**

Standard polygons are any polygon with three or more sides. Each polygon is separately described by its own vertices. There is no limit to the number of sides that can be used to describe a standard polygon, but it has only one normal vector defining which way it is facing. Triangulated polygons are those that are part of a mesh. Triangulated polygons share vertices with other triangulated polygons adjacent to it in the mesh surface, and as a result cannot be separated from the mesh without first breaking the surface apart. As the name implies, each polygon in the mesh has only three vertices and three sides. More important, when smoothed correctly, it also has three normal vectors attached to each vertex point.



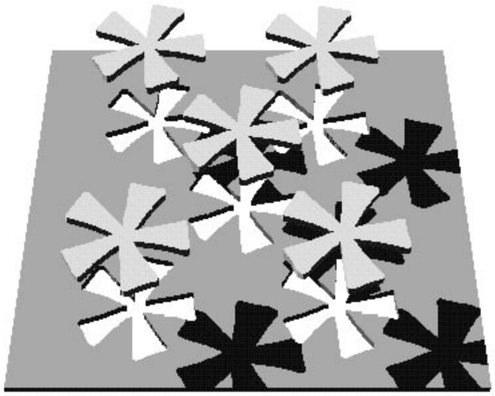
Standard polygonal surfaces

…render with a faceted appearance

Triangulated mesh surfaces

…render with a smooth-shaded appearance

Standard polygons can have holes in the surface, as well. The number of holes and the number of sides to the holes is likewise unlimited. Triangulated surfaces can have holes, but usually do not, since the calculations to maintain the integrity of the mesh around a hole are very complex.



Standard polygons can be any shape and have holes in their surface

To smooth or not to smooth

Triangulated surfaces may or may not render as a smooth surface. Triangulated mesh surfaces are only smooth when adjacent polygons share vertex points and those vertices have normal vectors attached to them. Non-mesh surfaces, even if they are composed of triangle-shaped polygons, can still render faceted.

If the geometry provided in imported files is defined using standard polygons, Vision3d will usually be able to import them and convert to triangulated mesh — appropriately smoothed, of course. Some applications will generate a polygonal mesh prior to exporting. This may be compatible, but usually Vision3d can do a better job at the mesh conversion and vertex normals calculations. If you have a problem with data from certain sources, see if you can re-smooth it with Smooth command. Failing that, see if it can be exported from the other application in non-mesh form.

Regular polygonal objects have surfaces where each polygon is defined by its own vertices, totally independent of other polygons that may be grouped with it in the object surface. Or perhaps you have purposely faceted an object to dissect it. After removing some of the object for a particular modeling purpose, you may then want to smooth the surface again.

Standard polygonal surfaces can be smoothed using the Smooth command in the Modeling menu. The smoothing operation sweeps a set of selected polygons and returns a triangulated mesh with appropriate normal vectors at each of the vertex points. If the surface is made up of three or four-sided polygons, the smoothing operation is straight forward.

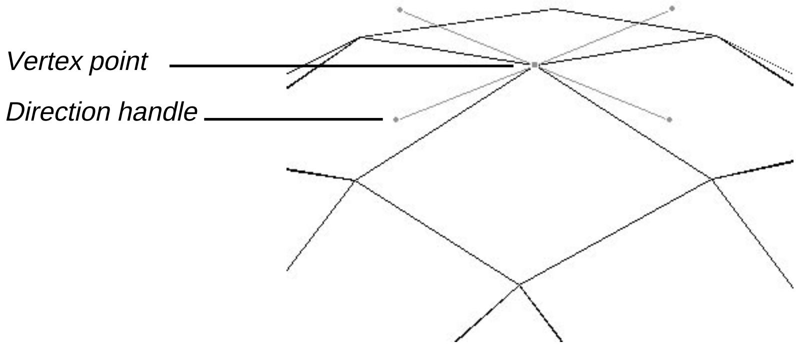
Why Bézier objects are different than polygonals

In the modeling window where you assemble the scene for your project, you do not know (or care) how many patches are used to make an particular object. Bézier objects look similar to polygonal objects in the modeling window, and you rarely need to differentiate between them. For a given geometry, a properly constructed and smoothed polygonal mesh surface renders identical to an assemblage of bézier surface patches.

But if polygonal objects are generally considered the “brute force” approach to modeling — even if mitigated by triangulation algorithms and surface normals for smoothing — bézier objects are elegant by comparison. That is not to imply they are easier to construct, or that the software required to maintain and render them is simpler. But rather that the amount of information required to describe an object’s geometry is so much less. A bézier surface is described by one or more surface “patches." Just like a polygonal object, the more patches used, the more memory required to store the object in the model and the longer it takes to render — but a bézier object is still more efficient.

Vertices or handles?

No discussion of bézier objects can go very far without introducing how surface patches are controlled. The number of surface patches becomes a factor when you bring an object into 3d Sculpter for editing its geometry. This is where you can see and select vertices that can alter the appearance of the object.



For uniformity in this manual, the terms "vertex point" and "vertices" are used interchangeably, to refer to both the points for manipulating bézier surfaces as well as those that describe polygonal meshes. Additionally, the term "handles" is used to refer to both the selection handles you see on a selected object in the model window as well as the direction handles that appear in 3d Sculpter on a bézier vertex point when it is selected.

Making curved surfaces from flat patches

The number of vertices present in the 3d Sculpter window indicates the number of surfaces patches on a bézier object. It is important to remember that each surface patch has four vertex points, one at each corner, even when the corners of a patch are not in rectangular form.

Another important point is that each surface patch is essentially a flat surface that is "wrapped up" into the desired geometry by positioning the vertices and bending the surface of the patch with the direction handles. Understanding these two axioms can help when you are editing a bézier object in the 3d Sculpter modeler and trying to make sense of how the surface of the object is constructed.

In most situations, a specific vertex point is shared by adjacent patches. Sometimes, however, multiple vertices from adjacent patches can be co-incident and appear to be one vertex point, until you start moving them. You must be careful to keep co-incident vertices together, or the surface of the object will split open. The number of direction handles visible on a vertex point when it is selected indicates how many patches that point controls.

Please refer to the discussion of the 3d Sculpter modeling extension later in this chapter for more information on editing bézier objects.

Tools Palette

The Tool palette contains the primary tools used to draw, transform, and manipulate basic objects in a model. Each tool is represented by an icon that indicates its function. Tools are selected by clicking on their icon. A tool's selection is confirmed when its icon becomes highlighted by appearing on the Tool palette in an inverted state.

Only one tool may be selected at one time, although some modifier keys can temporarily toggle tool functions for operating convenience. Regardless of which tool is currently being used, you can always access the menu bar, adjust views, size windows, move from one view to another, or choose another tool.

The Tool palette is essentially a floating mini-window. This means it can be moved around the screen and positioned for your convenience. The palette is intended to float above all model windows, always available for selecting tools. To move the Tool palette, drag the palette by its title bar.

The Tool palette is normally visible when a model window is open as the active window on the desktop. The Tool palette may be hidden three different ways:

- **By clicking on the close box in its title bar.**
- **By using the "Hide Tool Palette" command** in the Palettes sub-menu of the Windows menu.
- **By clicking on the Tool palette's hide/show button** on the Button bar.

To show the Tool palette again on the desktop, use one of the following methods:

- **Select the "Show Tool Palette" command in the Palettes sub-menu.** This command toggles with the "Hide…" command, depending on the display status of the tool palette.
- **Click on the Tool palette's show/hide button on the Button bar.**

The position of the tool palette on the desktop and its show/hide status are remembered from session to session. When a session is begun, the tool and other palettes are positioned, and displayed or hidden, just as they were when you quit the last time.

The tool palette is not displayed in situations where the Vision3d tools are not applicable, i.e. — a rendering window is the active window or Vision3d is not the active application. Bringing up desk accessories or choosing another application while in MultiFinder will also temporarily hide the tool palette. It re-appears when control is returned to a Vision3d modeling window.

Tool groupings on the palette

The tools are arranged on the palette according to their functions in the following grouped sequence:

- **View Management tools**
- **Object Manipulation tools**
- **3-D Primitives tools**
- **2-D Drawing tools**
- **Light Source tools**
- **Link and Camera Object tools**
- **Imaging tools**

Some two-dimensional objects, such as polygons and polylines, are drawn a section at a time. This type of tool is a special case, as explained later in this chapter. All other objects are sized and drawn in one step, using one of two methods:

- **FROM THE CORNER** — An object is normally positioned by one of its corners. To draw an object this way, just position the cursor where you want the object to start (in the view of your choice), press the mouse button, and drag the cursor any direction. A drawing marquee appears; growing or shrinking to indicate the size of the object as the cursor is moved. When the object is the desired size, release the mouse button to draw the object.
- **FROM THE CENTER** — With the centering method, position the cursor where you want the object’s center, hold down the Option key, then press the mouse button, and drag the cursor any direction. The drawing marquee grows and shrinks symmetrically around the point where the cursor was clicked in the view. When the desired size is reached, release the button.

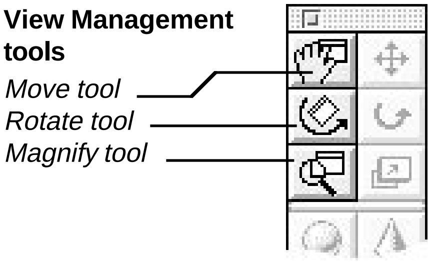
Of course, the Shift key may also be used in conjunction with the Option key (or without it) to constrain the proportions of the object.

As objects are drawn, they are positioned at the depth of the View Set center, parallel to the active view. When an object is drawn into the model, all views re-draw to show the new object. An object automatically becomes selected upon drawing; surrounded by the eight selection handles which may be used to further edit the object, i.e. move, rotate, scale, re-size, lathe, extrude, etc.

View Management tools

The view management tools are provided to control how views in the modeling windows display the model. They may work with just one active view at a time or adjust all the views in a view set in unison. The view management tools do not affect objects in the model in any manner.

View Management tools



With these tools, you may adjust the location of a view (or set of views) in 3-D space (which can also determine the views' distance from objects), the orientation of a view, the direction a view faces, and the scale of the image displayed in a view.

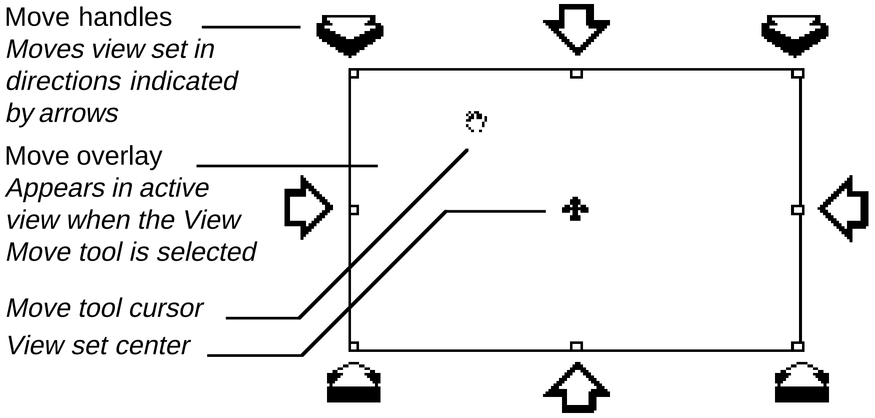
The view management tools operate only with overlays that appear in the active view when one of them is chosen. The overlays contain specialized handles that interact with the tool's cursor to effect the desired transformation on the view plane. In the case of the magnify tool, the overlay is not made visible, since no handles are required for its operation.

View Move/Grabber/Scrolling tool

This tool lets you scroll the view through a model that is too large to completely fit in the angle of view set for that window. As with other software, this tool is used to scroll a model in the window. However, it is important to understand that in Vision3d, the objects are not moved in the model; rather the view set is moved to display another area of the model space.

The view move tool operates by simply positioning the cursor anywhere in the active view, then clicking and dragging in any direction, including diagonally. This is called the “Grabber” method, because you are grabbing the view and moving it in an unconstrained manner. When the mouse button is released, the view set re-draws to show its new position in space, relative to objects in the model. Other views linked to the same view set center will also move and re-draw to show their new location.

You may limit movement of the view set horizontally or vertically by dragging one of the overlay’s center handles in the direction you wish to move. Using these handles automatically constrains the movement to only one axis at a time.



Move handles
Moves view set in directions indicated by arrows

Move overlay
Appears in active view when the View Move tool is selected

Move tool cursor

View set center

The view move tool also moves view planes closer to or farther from the view set center on the depth axis (Z axis, relative to the view). This operation uses the corner handles of the overlay to push or pull the view plane. During the in/out movement, numerical feedback is provided at the top of the active window showing the actual distance from the view set center

Normally, using the in/out handles in the active view moves all views of the view set closer or farther in unison. In this manner you can control all views in a view set so they will all display the objects in the model at a uniform scale.

Holding down the Option key allows you to move only the active view; leaving the other views of the view set alone. This may be useful when you wish to use one view as a close up port, while keeping other views for an overall display.

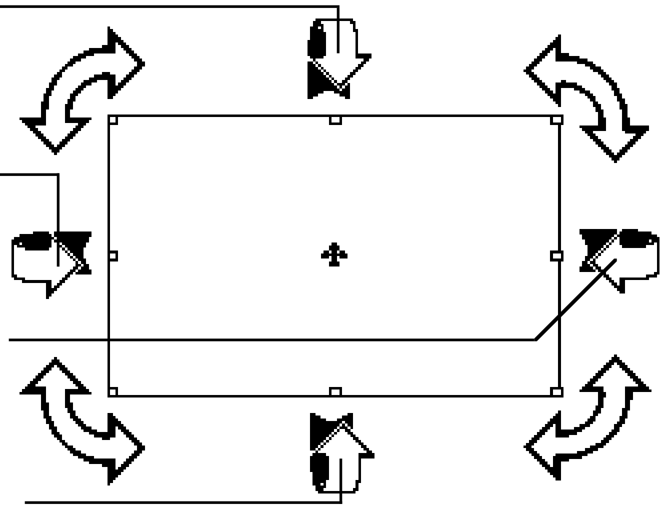
Modifier keys that apply to the View Move tool:

A. **Shift key** — Constrains the scrolling direction while using the "Grabber" method to either the vertical or horizontal axis.

B. **Option key** — Moves only the active view plane in or out while dragging the cursor on one of the overlay's corner handles. The other views in the view set are left stationary. Without the Option key, all view in the view set move closer to or farther from the view set center in unison.

View Rotate tool

This tool is used to rotate views around the view set center point. Selecting the View Rotate tool displays an overlay on the active view, similar to the one used by the View Move tool. This overlay contains different handles that the tool can use to rotate the view around the model on any axis. The overlay is visible in all views of the active window.



Top Center Handle
Drag cursor down to rotate view plane down vertically.

Left Center Handle
Drag cursor right to rotate view plane right horizontally.

Right Center Handle
Drag cursor left to rotate view plane left horizontally.

Bottom Center Handle
Drag cursor up to rotate view plane up vertically.

ALL Corner Handles
Drag cursor clockwise or counter-clockwise to rotate view plane around in/out axis.

The location of a window's view set center determines the center of rotation for rotating the views in that window. Each view is independently adjustable. The view can be rotated on one axis at a time, using the appropriate handle. During rotation, the window's feedback area shows the degrees of rotation relative to the previous location.

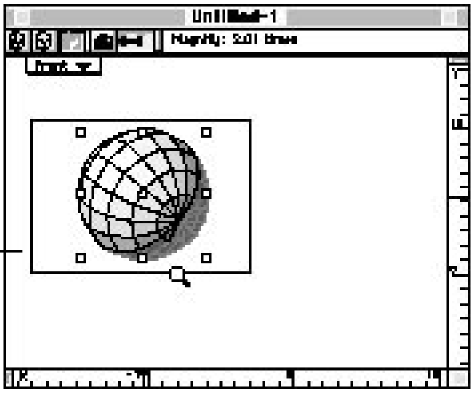
Modifier keys that apply to rotating views:

A. **Shift key** — Constrains the view rotation on the chosen axis to 45° increments. Without the Shift key, rotation is completely variable.

B. **Command key** — Allows the rotation of the view to be in less than 1° increments. Without the Option key, rotation is in whole degrees.

Magnify/Zoom tool

The magnify tool is used to magnify or reduce the scale of the image displayed in a particular view. Its purpose is to adjust the magnification factor of the view to control how much of the model is seen in a view. The magnification factor is different and independent of the field of view (perspective) control.



Example of "Drag to Zoom" use of Magnify tool

View centers on and magnifies the area indicated by the drag marquee

(With Shift key held down, the view set remains stationary)

There is a limit on how far a view can be magnified or reduced. That limit is determined by the mathematical precision of the view display. If you attempt to go beyond the limit, an alert appears and the system beep sounds. If you were allowed to exceed the math precision, the views would not show the objects correctly.

There are several alternatives to excessive magnification (or reduction). If you use one or more of the following mechanisms, you should be able to achieve the image desired in the view well before exceeding the software's math precision limitation:

1. **Move the view plane closer to (or farther from) the objects in the view.** Use the View Move tool, which operates on an overlay in the active view. Remember, to move only the active view in or out without affecting the other views attached to the same view set, hold down the Option key while using the tool.

   Alternately, you can first use the Fit Views to All or Fit Views to Selection commands to globally adjust the view set position, view distance, and view scale in one step. Then you can fine-tune the view display with the appropriate view management tools.

2. **Increase (or decrease) the focal length of the view.** This can be adjusted for each window independently using the perspective control at the top of the window. It has limited range, however, and may not be sufficient for your needs.

3. **Increase (or decrease) the scale of the objects.** This approach should be used as the last resort to solve view display and rendering anomalies. It is usually suggested for models that have been built extremely small or large. Although it may not be convenient for complex models, since it involves altering the objects so that they are of a more appropriate size, some situations may require this adjustment in order for the software to function at its best.

Magnifying (or reducing) an image does not move the view, nor change the actual size of the objects. It just changes the apparent size in the view. The magnify tool operates in two ways:

- **Click to zoom (fixed increment).** Each time the magnify tool is clicked in the active view, the image is magnified or reduced by a factor of two. The default mode is to magnify, indicated by a plus symbol in the cursor. Holding down the Option key changes the symbol in the cursor to a minus, indicating the tool will reduce the view image when clicked.

  The single click method also moves the view set center. When magnifying, the view zooms in on the area where the tool's cursor was clicked in the view. This also changes the location of the corresponding view set center to align on that area of the model.

  Zooming out does not move the view set center, since the point where you click will always show in the view.

- **Drag to zoom (variable increment).** Position the cursor in the view and drag — toward the center of the view to magnify or away from the center to reduce. This method variably adjusts the magnification factor up or down, depending on the direction and amount of the cursor drag.

If a "drag to zoom" operation is performed, followed by a "click to zoom" action, the view moves to the next magnification or reduction point used by the single click method (which is an even multiple of the fixed zoom factor).

Modifier keys that apply to magnifying views:

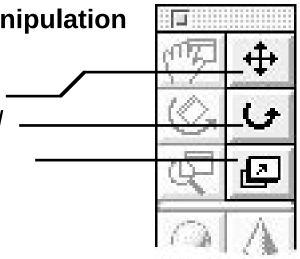
A. **Option key** — With the single-click method, holding the Option key down reduces the image.

B. **Shift key** — Using the cursor-drag method, holding the Shift key down magnifies the view without moving the view set center

Object Manipulation tools

The object manipulation tools take up the top right part of the tool pallet. They are used to select objects, transform, or generally manipulate two- and three-dimensional objects, as well as local light source objects.

Object Manipulation tools



Move tool

Rotate tool

Scale tool

In essence, there is only one tool in this section of the palette: THE POINTER. However, the pointer's operation is modified by which tool is selected, so that it functions as three separate pointer tools.

Selecting an object enables these tools to perform their specific operation. Depending on the pointer tool chosen, the cursor changes shape when positioned over one of the handles that appear around selected objects.

Pointer cursor

Object Move cursor

Object Rotate cursor

Object Scale cursor

In all three tools, the pointer can be used to select objects and/or light source icons. Objects are selected by clicking on them in the active view. The selection remains in effect until you change it. Once an object or group of objects is selected, you can transform or edit them without affecting the rest of the model.

The act of selecting or de-selecting does not modify objects in any way; it only allows other tools or commands to operate on those objects.

You typically must select an object or set of objects before performing any operation on them.

Switching tools from the keyboard

A method is provided for conveniently toggling between a modeling tool and one of the pointer tools. For example: If you are using the sphere tool and wish to quickly move an object before proceeding, you can easily use the keyboard shortcut: the Space bar. Rather than taking the time to move the cursor to the Tools palette and click on the Object Move tool icon, simply tap the Space bar once and the last used pointer tool is temporarily selected. When done with the pointer tool, tap the Space bar again to go back to the previous tool.

You may choose one of the three pointer operation modifiers and toggle between it and the current tool each time the Space bar is pressed.

Selecting objects with any of the pointer tools.

The simplest selection technique is to click on the surface of an object in the active view. Whether an object is selected when clicked depends on the Mouse Filter settings in the Selections menu. Please refer to an explanation of the Mouse Filter command later in this chapter.

If you wish to select more than one object, hold down the Shift key while clicking one at a time on multiple objects. The Shift key maintains previous selections while extending the selection to other objects. Additionally, holding down the Shift key while clicking on an object that is already selected within a collection de-selects that object while retaining the other selections. Using this method allows selection of non-adjacent objects so they may be manipulated in unison while maintaining their spatial relationship.

If you click on overlapping objects, only the front-most object is selected. However, a set of selection commands are provided in the Selection menu (or sub-menu on smaller monitors) to give access to objects within or behind other objects. You may also use a view from another direction to gain access to hidden objects.

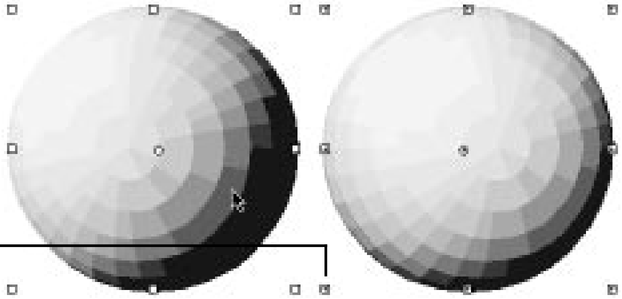
There are two selection techniques available in Vision3d:

- **Drag a selection marquee around the item or items to be selected.** Normally, an item is selected only when completely enclosed within the selection marquee. By holding down the Command key, all items touched by the selection marquee become selected.
- **Extend or reduce a selection group by "shift-clicking."** Holding down the Shift key while selecting an object maintains previously made selections.

NOTE: *To de-select all objects, click anywhere in the view away from any objects. You may also use the Select None command in the Selection menu. (If you do not have a Selection menu, look for the Selection command in the Edit menu.)*

When an object is selected, eight manipulation handles appear around it. These handles are visible in all open views showing that object. The presence of the handles indicates that the selection attempt was successful and provides engagement points for the object manipulation tools or a target for other modeling tasks that must have selected objects on which to operate.

Appearance of handles varies with “GROUP” status:



The appearance of the selection handles differs for single objects or groups/shapes. Thus, when a selection is made, you can immediately tell if it is a group or an instantiation of a Shape by the presence of a plus inside each handle.

Modifier keys that apply to selecting objects:

A. **Shift key** — Extends the selection of objects. Without the Shift key, only one object can be selected at a time.

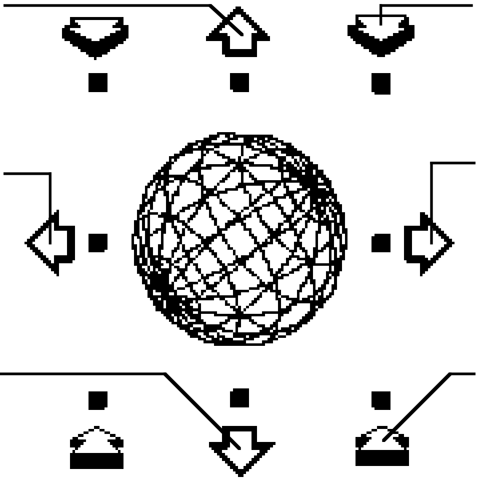
B. **Command key** — Selects all objects that are within or touching a selection marquee. Without the command key, only objects completely within the marquee are selected.

Object Move tool

The Object Move tool operates in the active view on selected objects. The move operation may be done either by click-dragging on the surface of an object or by dragging by selection handles. Using the handles offers additional constraint and control not available with the simpler method.

Moving by dragging handles

The Object Move tool is used to move objects horizontally and vertically, as well as in and out. This tool can also be used to move camera objects and light source icons in the same manner. Selection handles effect object movement as shown in the following illustration:



Top Center Handle
Drag cursor up to move object up vertically in view.

Top Corner Handles
Drag cursor down to pull object toward view.

Left Center Handle
Drag cursor left to move object left horizontally in view

Right Center Handle
Drag cursor right to move object right horizontally in view

Bottom Center Handle
Drag cursor down to move object down vertically in view

Bottom Corner Handles
Drag cursor up to push object away from view

Using the selection handles allows you to move an object on any axis you choose from the same view. Dragging an object without using the selection handles allows movement horizontally and vertically in a view, but not in or out.

Modifier keys that apply to handle-constrained movement:

A. **Shift key** — Not required for moving objects by dragging on a selection handle. Movement using the Move tool with selection handles is constrained automatically by the drag function of the chosen handle.

B. **Command key** — Temporarily disables the grid detents while the object is being moved. The key must be held down AFTER the move operation is begun to have the modifying effect described here. Holding down the Command key BEFORE the tool is clicked on the object to begin dragging causes the object origin points to remain stationary. (See the next paragraph.)

C. **Command key** — Drags the origin points (both object and texture, if visible) without moving the object itself. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

D. **Option key** — Leaves a copy of the object in the original position.

E. Control key — Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other, as well as the object. With the Control key down, either the object origin point or texture origin point (if visible) can be moved without moving the other point or the object itself.

Moving by grabbing on the surface

This method allows you to choose and drag any object by its surface regardless of the object manipulation tool currently selected. This is referred to as a limited-function move.

To move an object in this manner:

1. **Position the cursor over the selected object,** but not on one of its selection handles.
2. **Press down the mouse button and drag the object.** A bounding box marquee appears to indicate the location of the object during the move.
3. **Release the mouse button.** The screen re-draws to show the object's new location.

To constrain object movement in either the horizontal or vertical axis, hold down the Shift key while dragging. To leave a copy of the object in its original position, hold down the Option key before beginning the drag operation.

If more than one object is selected, multiple marquees appear and move in unison. Objects stay a constant distance from the viewing plane while using this drag method.

Modifier keys that apply to free object movement:

A. Shift key — Constrains the cursor movement to straight horizontal or vertical lines.

B. Command key — Drags the origin points (both object and texture, if visible) without moving the object itself. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

C. Command key — Temporarily disables the grid detents while the object is being moved. The key must be held down AFTER the move operation is begun to have the modifying effect described here. Holding down the Command key BEFORE the tool is clicked on the object to begin dragging causes the object origin point to remain stationary.

D Option key — Leaves a copy of the object in the original position.

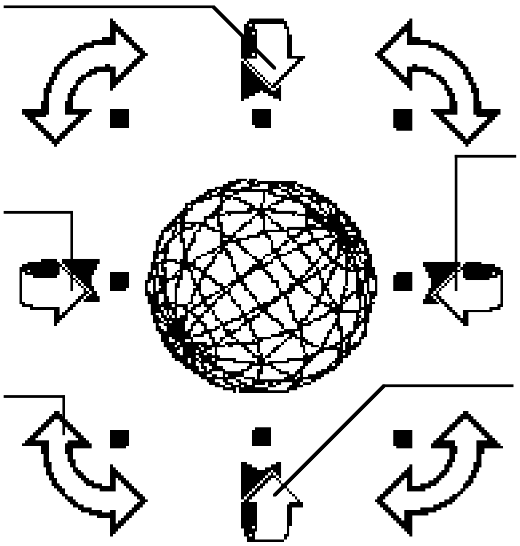
E. Control key — Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other, as well as the object. With the Control key down, either the object origin point or texture origin point (if visible) can be moved without moving the other point or the object itself.

Object Rotate tool

The Object Rotate tool operates in the active view on selected objects. The rotate operation may be done either by click-rotating on the surface of an object or by rotating by selection handles. Using the handles offers additional constraint and control not available with the simpler method. The unconstrained rotation method is sometimes called "trackball" rotation.

Rotation by dragging handles

The Object Rotate tool is used to rotate objects. This also includes camera objects and light source icons. An object's selection handles are used by the pointer tool to grab the object and rotate it around its object origin point. Rotation can be made on all three axes, as shown in the following illustration:



Top Center Handle
Drag cursor down to rotate object around horizontal axis

Left Center Handle
Drag cursor right to rotate object right around vertical axis

All Corner Handles
Drag cursor to rotate object clockwise or counter-clockwise around in/out axis

Right Center Handle
Drag cursor left to rotate object left around vertical axis

Bottom Center Handle
Drag cursor up to rotate object around horizontal axis

Rotating an object or set of objects changes the orientation of the object(s) in 3-D space. The viewpoint from which they are rotated remains stationary.

The axis of rotation for this method of rotating an object is determined by the location of the object origin point. The object origin point is the round dot that appears with the selection handles on an object. Normally, this point is at the geometric center of the object, but can be moved if desired.

Refer to the Origin Points topic later in this chapter for more information on their use. Additional material regarding origin points and animation paths is also presented in *Chapter 7: Scripting Animations*.

Modifier keys that apply to rotating objects using the selection handles:

A. **Command key —** Drags the origin points (both object and texture, if visible) without moving the object itself. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

B. **Option key —** Leaves a copy of the object in its original position.

C. **Shift key —** Constrains the rotation to 45° increments.

D. **Control Key —** Instead of rotating the object, the rotate tool rotates any surface maps that may be part of a texture applied to the selected object.

E. **Control key —** Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other, as well as the object. With the Control key down, either the object origin point or texture origin point (if visible) can be moved without moving the other point or the object itself.

Rotation by trackball method

Clicking and dragging on the surface of an object with the Object Rotate tool rotates the object in an unconstrained manner. Rotation can be horizontally, vertically or diagonally.

The Shift key does NOT constraint the rotation direction.

Note also that the object origin point is not the axis of rotation for the trackball rotation method. Rotation is around the geometric center. This difference of functionality between the two methods of rotating objects can be used to advantage in situations when the object origin point has been positioned at other than the geometric center for a specific purpose, but then the need to rotate the object on itself arises. Using the trackball method, you do not have to temporarily re-center the object origin point to rotate the object, then move it again.

Modifier keys that apply to rotating objects using the trackball method:

A. **Command key** — Drags the origin points (both object and texture, if visible) without moving the object itself. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

B. **Option key** — Leaves a copy of the object in its original position.

C. **Control Key** — Instead of rotating the object, the rotate tool rotates any surface maps that may be part of the texture applied to the selected object.

D. **Control key** — Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other, as well as the object. With the Control key down, either the object origin point or texture origin point (if visible) can be moved without moving the other point or the object itself.

Scale tool

The scale tool is used to re-size or stretch objects and light source icons.

Scaling an object allows you to enlarge or reduce it horizontally, vertically, or its depth…or on all three axes at once, relative to the original object. An object is scaled in the direction the cursor is dragged.

Normally, the depth of the object is not changed, unless constrained with the Shift key (see modifier keys list for this topic). Without the Shift key, an object's proportions can be "distorted" with the Scale tool. Additionally, the depth of an object can be adjusted independently by stretching the object in another view.

Modifier keys that apply to scaling objects:

A. **Command key** — Keeps the origin points (both object and texture) stationary during a scale operation. Normally the origin points reposition in the direction of the scale operation and in the same proportion as the re-sized object. For example, if the object origin point is in the geometric center of the object being scaled, it will move to the geometric center of the newly sized object, unless the Command key is held down.

B. **Command key** — Moves the origin points (both object and texture, if visible) without moving the object. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

C. **Option key** — Scales an object from its geometric center.

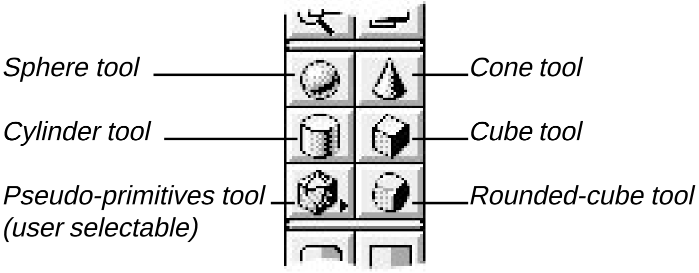
D. **Shift key** — Maintains an object's proportions while enlarging or reducing its size. The object will grow or shrink on all three axes according to the path you drag.

E. **Control key** — Instead of scaling the object, the scale tool scales any surface maps applied to the object.

F. **Control key** — Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other as well. With the Control key down, either the object origin point or texture origin point (if visible) can be moved without moving the other point or the object itself.

Primitives tools (3-D objects)

This section of the tool palette contains tools that draw 3-D primitives. Primitives enjoy a special status in Vision3d. They take the least amount of memory and render the fastest of all object types. During modeling projects, it is best to use primitives whenever possible.



Primitive objects may be drawn two ways:

- **With a single click.** This inserts the chosen primitive at a default size and with equal proportions.
- **With a drag marquee.** The 3-D object's height and width are determined by dragging the cursor in the active view. The depth is equal in dimension to the smaller of either the height or width. Holding down the Shift key while drawing a primitive constrains it to equal proportions on all three axis. Without the Shift key, the object can be distorted.

Either way, once they are drawn in the model, these objects may be worked with and edited on three different levels:

- **Transformations.** Using the object manipulation tools on the Tool palette, a primitive may be moved, rotated and scaled. This is a low-level type of object editing with no memory penalties.

- **Revisions.** Through the Edit Primitive command in the Modeling menu, a primitive may altered, but still retain its original object type (that of primitive). Maintaining the object's type is desirable for memory efficiency.
- **Conversions.** Using one or more of the available modeling extensions (listed at the bottom of the Modeling menu), a primitive's object type may be converted to either individual polygons, polygonal mesh, or a bézier object for extensive changes to its geometry.

Editing a primitive object's geometry on this level can be very powerful, but also more memory intensive. Once an object of any type or class is converted to another form, it cannot be converted back to its original type, nor edited with its native modeler. Thus, once a primitive is changed to a bézier object, it is no longer a primitive.

Modifier keys that apply to drawing 3-D objects:

A. **Shift key** — Constrains the tools to draw objects with the same dimensions on all axes. Without the Shift key, objects can be distorted.

B. **Option key** — Draws an object from its center, instead of the corner.

Sphere

Use this tool to draw spherical-shaped objects.

A sphere is represented in the modeling windows as a polygonal object. The apparent fineness of the polygons making up a sphere's surface are not an indication of surface smoothness. You will note that the closer the object is to a view (that is, the larger the object appears in a view), the finer the polygons appear. The sphere you see in the modeling windows is just a representation of its shape. As a primitive, the curved surface of spheres is automatically smoothed when rendered.

Cone

Use this tool to draw cone-shaped objects.

The cone is always drawn with the point facing up (positive Y Axis, relative to the active view). The height of the cone may be adjusted when using the drag marquee method of drawing the object. If the cone's proportions are modified during drawing, the base is also distorted from a perfect circle. Like the sphere tool, the cone's curved surface is smoothed during rendering.

Cylinder

Use this tool to draw cylindrical-shaped objects.

Both ends of a cylinder are always identical when drawn. A cylinder is drawn with its ends oriented to the window's Y axis. If the cylinder's proportions are modified during drawing, the ends are also distorted from perfect circles. The cylinder's curved surface is smoothed when rendered. A cylinder may be made solid or hollow using the Edit Primitive command in the Modeling menu.

Square-corner Cube

Use this tool to draw cube-type shapes in the active view.

Unlike similar-appearing objects that may be created using separate polygons for the six sides of the cube, a cube primitive is NOT ungroupable. Like other primitives in this set of 3-D tools, it is defined as a single mathematical entity.

Round-corner Cube

Use this tool to draw cube-type shapes with rounded corners.

The radii of the rounded corners is fixed. Like the standard cube, this object is a single entity and not un-groupable.

NOTE: While the two cube primitives are single, un-groupable objects, you can use the Facet extension to break them into individual polygons, which can then be separately selected for moving, deleting, or applying different textures as desired. After editing the faceted object, you can simply group the polygons or use the Smooth modeler to make it a single entity again.

Pseudo-primitives

This tool allows the insertion of an object chosen by the user. Objects drawn with this tool are called pseudo-primitives because you can draw them just like a primitive, but they are usually constructed by one of the modelers and therefore not internally supported like primitives are. It is designed as a convenience for inserting often-used objects.

The objects available for use with this tool are stored in a Primitives folder in the Vision3d folder with the application. Several objects are included in the folder by Strata Inc.

Installing your own primitives

You may design your own pseudo-primitives in the software, then save them as single-object models. The object may be two- or three-dimensional in type. Only the geometry can be used. All other data, such as textures, lighting, etc. are ignored for the purposes of this tool.

All economies in modeling techniques should still be considered in the creation of these pseudo-primitives. Each insertion of a pseudo-primitive takes the same amount in memory as when it was originally made.

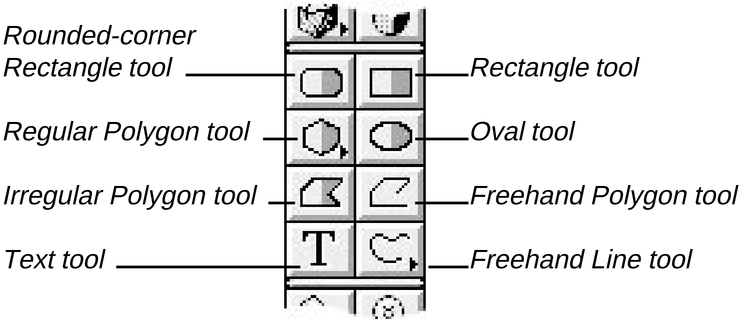
Pop-out menu

To choose a shape, click and hold down the mouse button on the tool icon. The objects available are displayed in a sub-menu. The selected shape may be changed as often as needed. The object chosen is retained between sessions.

If no qualifying extension objects are found in the Primitives folder, a default object, called "polyhedron," is built into this tool. It is always available.

Drawing tools (2-D objects)

The two-dimensional object tools are used to draw 2-D objects. By definition, these objects have height and width, but no depth. Typically, they serve as templates for the lathe and extrude tools, but may be left as two-dimensional objects.



The size and proportion of the object are determined by the starting and ending position of the cursor as it is dragged in the active view. The object is positioned in the active view at the depth of the view set center, when drawn.

Each tool, except Open Polyline and Freehand, has a filled and frame representation showing on the Tool palette. You can select the filled version by clicking on the right side of the tool or the frame version by clicking on the left side. A filled object renders with a surface, while the frame-only version renders hollow.

If you need to alter a 2-D object after it has been drawn, you can use the Edit Primitive command in the Modeling menu. This command allows you to change its status between filled and hollow, as well as other parameters relevant to the specific object selected. Please refer to the Edit Primitive command discussion later in this chapter for an explanation of the editable parameters available in the Edit Primitive dialog.

If you need to change a 2-D object's actual geometry, use the Reshape or 2d Sculpter commands in the Modeling menu. The Reshape command is present in the menu only when a 2-D object is selected. Both commands open the 2d Sculpter modeler, in which you can push and pull the vertex points that define the object's form. Alternately, you can simply double-click on a 2-D object to bring the object into the 2d Sculpter modeler. Please refer to the 2-D Sculpter discussion later in this chapter for an explanation of how to use this modeler.

Modifier keys that apply to drawing 2-D objects:

A. **Shift key** — Constrains the tools to draw equally-proportioned objects, i.e. — `squares, circles, etc. With the polygon or polyline tools, the Shift key constrains line segments to 45° increments.

B. **Option key** — Draws an object from its center, instead of the corner. The Option key does not apply to the irregular polygon or polyline tools.

Round-corner Rectangle/Square

Use this tool to draw rectangles with rounded corners. The radius of the corners is adjustable through the Edit Primitive command after it has been drawn.

Square-corner Rectangle/Square

Use this tool to draw rectangular-shaped objects. It can draw squares or rectangles of any proportions The object always have square corners.

Regular Polygon

Use this tool to draw multi-faceted shapes with equilateral sides. The number of sides for the object is adjusted by holding the mouse button down while the cursor is on the tool icon. You may then choose the number of sides from the pop-up menu. The number of sides chosen for this tool is maintained between sessions.

Elliptic/Circle

Use this tool to draw oval-shaped objects. It can draw exact circles or ovals of any proportion.

Irregular Polygon

Use this tool to draw irregular, multi-sided objects. The object is drawn by defining straight line segments that connect to make up the complete shape. This is a closed polygon object, i.e. — the beginning and ending points are the same.

To create a irregular polygon object, click the cursor where the first line is to begin. Click again to define the end of the first line and the beginning of the next line. Line points may be deleted while drawing the object using the Delete key. The most recent point is deleted first, repeated deletes remove polygon points in reverse sequence of their creation. When you are done with the completed shape, just double click upon entering the last point. The begin and end points will automatically connect with a line, if they do not already touch.

Open Polyline

Use this tool to draw multi-segmented, open line objects.

The polyline tool creates objects whose shape is defined by connected straight line segments. This object is NOT closed, i.e. — even if the begin and end point touch, the object is not a closed polygon.

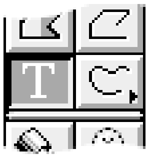
The Open Polyline tool uses the same technique as the Irregular Polygon tool.

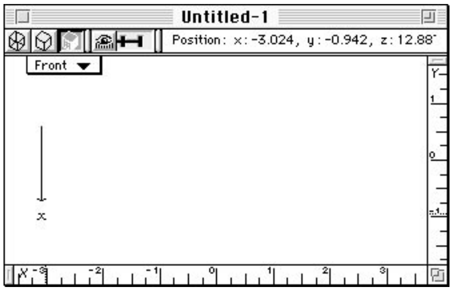
Text

Use this tool to enter text-shaped polygons into a model.

All fonts currently available in the System are accessible to this tool. The location and size of the text is designated by dragging the cursor vertically in the active view.

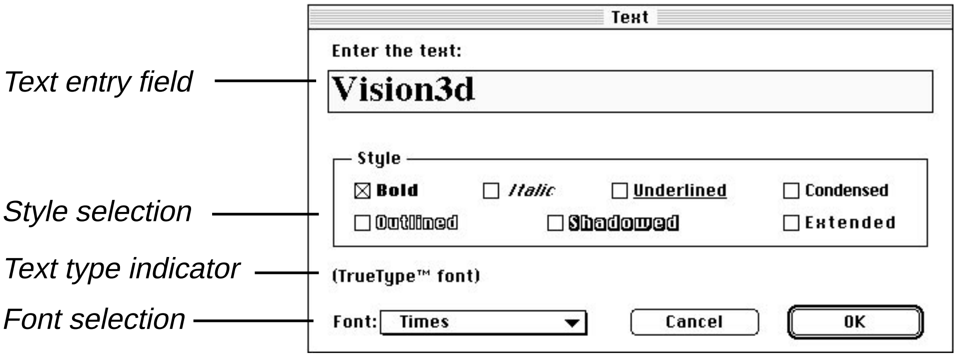
Drag tool vertically to size text to desired hight.





Upon release of the mouse button, the text dialog box appears to enter the characters. All fonts installed in the system are usable in Studio Pro. The specific font you wish to use is selected in the Font field's pull-down menu.

If an outline font technology is available, it may be used to smooth the font shapes and yield an improved bézier-type extraction of the text shapes. Under System 6.0 software, only ATM is available for font smoothing. The use of ATM is indicated when you choose the font. If the font cannot use ATM, because there is no PostScript type 1 document present for that font, the dialog indicates the characters will be constructed from the bitmapped font. The characters are then extracted as bézier objects using an autotrace technique.



Text

Enter the text:

Vision3d

Style

☒ Bold ☐ Italic ☐ Underlined ☐ Condensed

☐ Outlined ☐ Shadowed ☐ Extended

(TrueType™ font)

Font: Times ▼

Under System 7.0 or later system software, Vision3d detects the presence of either TrueType or ATM outline font systems and employs the appropriate smoothing method in the bézier extraction process.

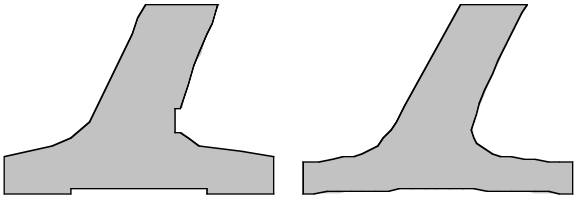
NOTE: *If the font chosen for the Text tool is present in the System Folder in both ATM and TrueType form, preference is always given to using the TrueType font. This is true in any application, not just Vision3d.*

Bézier Spline Support for Text

Text-type objects are created as fully-smoothed 2-D shapes from the fonts. Again, both Adobe Type1 and Apple TrueType fonts are supported. As bézier splines, fewer points are required to define the geometry and less memory is used. Best of all, the resulting objects are more accurate to the actual font form.

Enlarged views of the letter "A" (left leg) in Palatino font:

Strata Vision3d creates a bėzier representaion of fonts for a smoother, more accurate shape



Polygonal extraction *Bézier extraction*

The font shape remains a bézier representation until another modeler is used to modify the object, i.e. — extruding, lathing, sweeping, etc.

The bézier splines are then converted to polygonal mesh. (Note that the Skin/FlexObject modeler retains the bézier information for the ribs, as well as creating a bézier skin.)

Bézier to polygonal mesh conversion
— Controlling the fineness

A feature of the bézier to polygonal mesh conversion process is that you can control how fine the resulting mesh surface is. That is, you can set how many polygons will be generated in the resulting extruded, lathed, or swept objects.

The fineness of the conversion depends on the apparent size of the 2-D object(s) in the view. The smaller the objects, the coarser the conversion. The larger the objects, the finer the conversion. This is most noticeable on curved edges, as shown in the illustration on the next page.

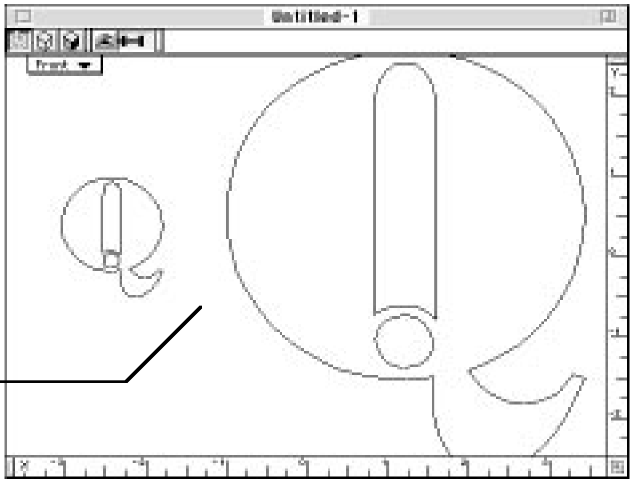
There is a minimum refinement threshold. So, even in instances where the 2-D object being used as a modeling template may appear very small in the view, the resulting extruded, lathed or swept object will always have at least an acceptable quality level.

However, if the appearance of the 3-D object is important to the final rendered image, you can use this technique to make sure the object is constructed at a higher standard. Of course, you do not need to make the actual size of the object larger, you can just use the magnify tool to zoom in on the object so that it appears bigger in the view. Notice how the large “Q” object is even bigger than the view. The closer you zoom in, even if only part of the object is showing in the view, the finer the surface.

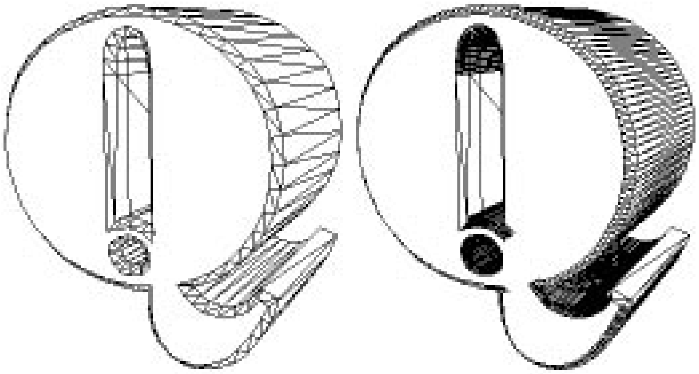
As in every other aspect of 3-D modeling, there is a trade off. In the above example of extruding these two text objects, everything was equal except the apparent size in their respective views. The one on the right required 2,033 polygons. The coarser 3-D object on the left generated 341 polygons in its construction and rendered in less than half the time.

The fineness of the surface for an extruded text object depends on the apparent size of the object in the view

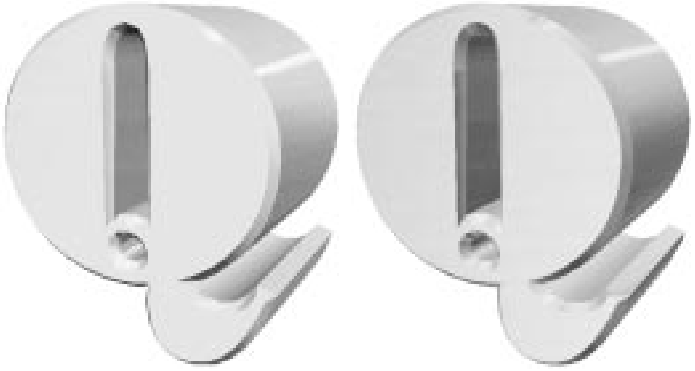
Compare the two letters at right, with radically different sizes before extuding



When these two objects are extruded, there is a significant difference in the number of polygons created to extrude the mesh surface



Notice the slight faceting of the rounded edges of the object on the left, when rendered

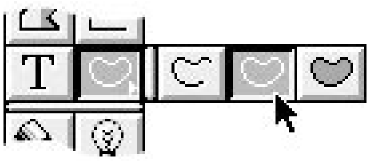


NOTE: *These two objects were actually modeled in completely separate models. They were exactly the same size, but the one on the right was magnified in the view to the scale shown here before extruding. They are shown together in the same modeling window for simplicity and to show their sizes relative to each other (and the model window) when extruded.*

Bézier Pen

Use this tool to draw bézier objects in a completely freehand method.

You may draw filled objects, hollow objects, and open lines. The type of bézier object drawn must be chosen from a pop-out menu BEFORE drawing the object.



After an object is drawn with this tool, it may be modified with the Edit Primitive command in the Modeling menu. Its geometry may be further edited using the 2d Sculpter modeler. This modeling extension allows you to move the vertices defining the bézier line.

Open Bézier line

Use this tool to draw open curved lines. The Open Bézier tool, of course, does not create closed polygons and therefore cannot be filled.

Curved lines are drawn by clicking the mouse button and dragging the cursor in the active view to describe the shape of the line desired. Releasing the mouse button ends the drawing of the line.

No surface is generated when this object is drawn. If left in this state, the line will not appear in a rendered image, even though it is visible in modeling windows. Typically, these objects are used as templates to extrude, lathe, sweep, etc. to construct a 3-D object that does have a surface.

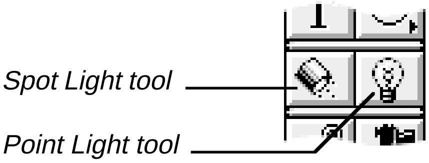
Closed Bézier object

Use this tool to draw closed curved-line objects.

The boundary of the closed object is drawn by clicking the mouse button and dragging the cursor to describe the free-form shape. The line defining the closed bézier object is visible as it is being drawn. Because it is a closed object, when the mouse button is released, the line automatically connects between the begin and end points, if necessary.

Light Source tools.

The next section of the Tool palette contains tools to insert spot and point light sources in a model. Light sources provide illumination to a model so objects will be visible when rendered.



Spot and point lights are considered local sources, designed to illuminate small areas of the scene. This is in contrast to the directional light, which is considered a global source, designed to illuminate the entire model like sunlight striking the Earth.

Light source icons are visible in the modeling windows so you can position them as needed, but they do not show in a rendering. Only the effects of their lighting are visible in the rendered image.

Light sources can be inserted in a model by two methods:

- **With a single click.** This inserts the chosen light source at a default size and with equal proportions.
- **With a drag marquee.** The light source icon's height and width are determined by dragging the cursor in the active view. The depth is equal in dimension to the smaller of either the height or width. Holding down the Shift key while drawing a light source constrains it to equal proportions on all three axis. Without the Shift key, the icon can be distorted.

Of course, you can move, rotate and scale the light source as needed, once it is in the model.

All light sources have an infinite illumination distance by default. Global light sources are not adjustable for illumination distance, but local light sources can be given an attenuation factor so that the distance their light travels from the spot or light source is limited. This parameter is set in the Edit Light Source dialog box, which opens each time a light source is inserted in the model. This is the same dialog box that opens when you choose the Edit Primitive command in the Modeling menu.

If you wish to suppress opening the Edit Light Source dialog, thereby automatically accepting the default settings for the light source, simply hold down the Control key while drawing the icon.

NOTE: *The actual locus of the luminescence from a spot or point light source is finite within the icon. The size of a light source icon does not influence the intensity or distribution of the light. However, the size and shape of the icon can influence the distribution of soft shadows using the Raydiosity rendering algorithm. If the field that controls soft shadows is set greater than one in the Raydiosity dialog, the shape of light source icons determine the penumbra effect for shadows cast by that light source.*

Modifier keys that apply to light sources:

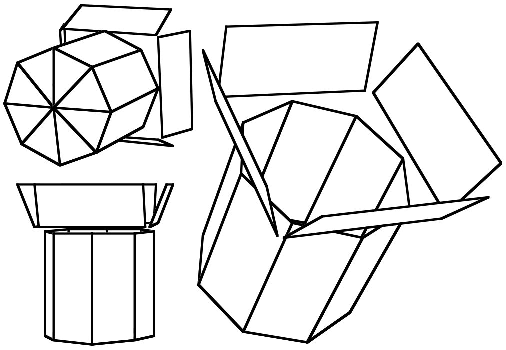
A. **Shift key** — Constrains the tool to draw the icon with the same dimensions on all axes as it is drawn with the drag-marquee method. Without the Shift key, the icon can be drawn with any proportions.

B. **Option key** — Draws a light source icon from its center, instead of the corner.

C. **Control key** — Suppresses opening the Edit Light Source dialog and accepts the light source's defaults. Without the Control key, the Edit Light source dialog opens each time a light source is inserted.

Spot light tool

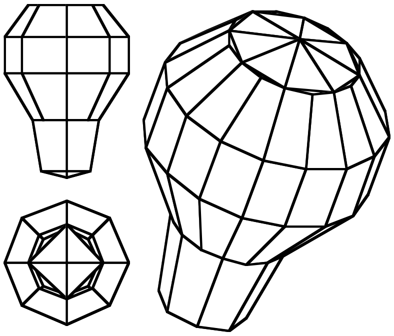
Use this tool to insert spot light sources in a model. Spot lights shine in the direction determined by the icon's orientation. Spot lights are further adjustable for intensity, cone angle, and attenuation factor.



Spot light sources use an icon that looks similar to an actual spot light. The shape of the icon is intended to indicate which way the light is shining. A spot light is drawn facing away from the active view. That is, the back of the light source is what you see when the icon is drawn in the view.

Point light tool

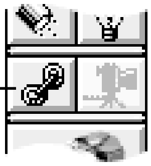
Use this tool to insert non-directional light sources in a model. Point lights shine outward in all directions from the source.



Point light source use an icon that looks like a light bulb. Even though its illumination shines in all directions, the shape of the icon helps you orient it consistent with the rest of your model. A point light is drawn right side up in the active view. That is, the top of the icon is pointing toward the positive Y axis, relative to the active view's orientation.

Linking tools

Use the Link tool on the Tool palette to establish a special connection between two objects. This is called a link. The link connection is used primarily for constraining object movement during animation sequences.

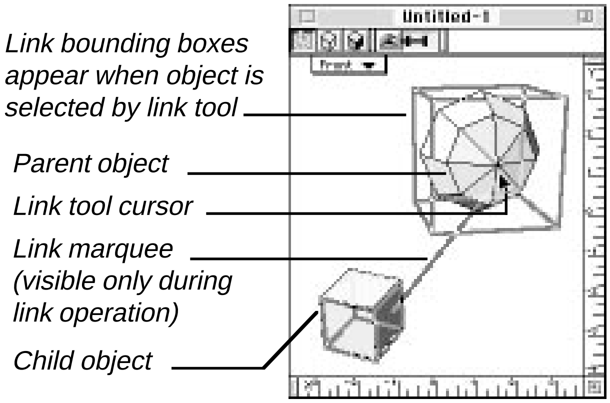


Link tool

A link connects a “child” object to a “parent” object. You should know that the link feature currently implemented in Vision3d does not constrain the range of movement or degree of rotation when moving the child. The link is intended only to help child objects behave in a coordinated way when the parent is moved, i.e. — the child follows along when the parent is moved.

Link tool

The Link tool makes the link connection between the child and parent. The child is identified through selecting it first, then dragging the link cursor to the parent. Each object becomes highlighted in turn as the link is made. Note that the child's object origin point is moved to the geometric center of the parent.

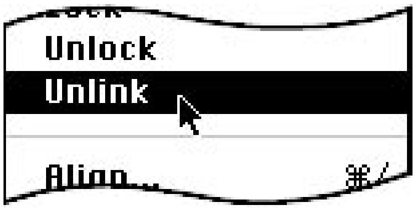

Untitled-1
Front
Link bounding boxes appear when object is selected by link tool
Parent object
Link tool cursor
Link marquee (visible only during link operation)
Child object

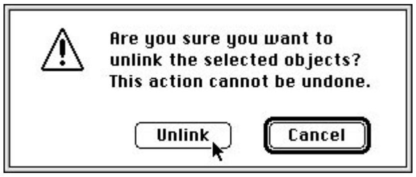
Only two objects may be involved in any one link operation. However, a linked object may be further linked to another object in a subsequent use of the tool. For example, several 2-D objects may be linked in a linear sequence for use as ribs in preparation for skinning, or a parent of a previous link may become the child in another link.

If an object is selected as a child for linking to a parent that is already a child to another object, the first link is automatically broken. No alert is given to indicate the change in linking. It is the user's responsibility to keep track of which objects are linked together. This is not as hard as it sounds.

Unlink command

The Unlink command in the Modeling menu is provided to break the link between previously-linked objects. This performs a global severing of links for all selected objects. Linked objects that are not selected are not affected by this command.



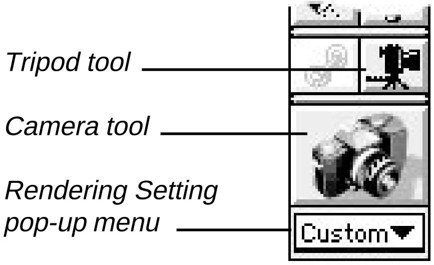


Are you sure you want to unlink the selected objects? This action cannot be undone.

Choosing the Unlink command causes an alert to appear asking if you are sure you want to break the link between the selected objects. If you choose to break the link , this step cannot be undone.

Imaging tools

The imaging tools provide the means for setting up camera objects inside the model and for taking snapshots for rendering from any view. While modeling windows allow you to work and view the model generally from the outside, camera objects allow you to open a window to view the model from any position inside the model.



Camera Object/Tripod tool

The tripod tool inserts camera objects in the model. To add a camera object to a model, just select the Tripod tool, then click and drag within the active view. The camera object is inserted into the model space at the same position as the active view, facing away from the view in which it was placed.

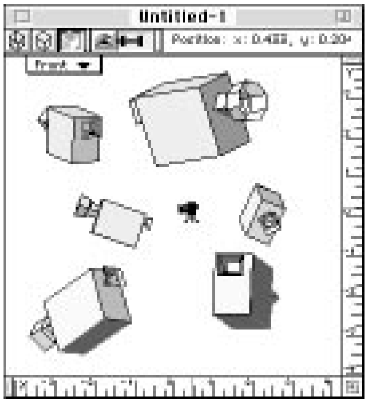
There is no limit to the number of camera objects that may be inserted. These objects can be positioned anywhere in 3-D space and independently pointed in the direction desired. Once inserted, you may open a camera window connected to a camera object's "viewfinder." This can be done by either double-clicking on the camera object or choosing it by name from the Camera Objects sub-menu in the Window menu. Camera windows are discussed in *Chapter 1: General Operation*.

Note that while camera objects are visible in modeling views, they do not appear in renderings. You do not have to worry about a camera object "spoiling" a rendering by being seen.

Camera objects are inserted into the model in the same manner as 3-D objects are drawn. You can insert a camera object in two ways:

1. **Single click.** The object is inserted at the location where the cursor was clicked and just in front of the view plane. The camera object is facing away from the view through which it was inserted. The camera object's icon is shown as a default size.
2. **Cursor drag.** The object is handled in the same manner as the single click, except that the size of the object's icon is set by the marquee created by dragging the cursor.

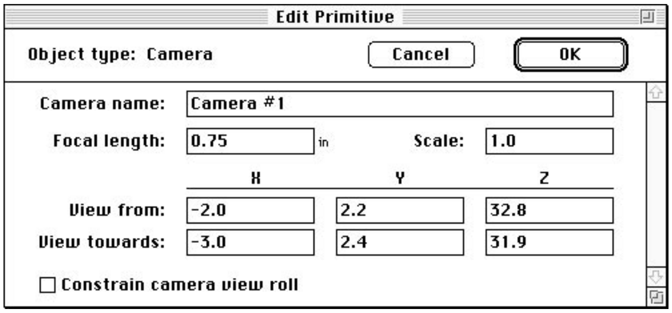
The icon has distinguishing features that help orient it in the model. The object has a definite front that must be pointing at the objects you wish to see in the camera window.



Icons representing Camera objects may be moved, rotated and scaled just like any other object in the model. When a camera object icon is moved or rotated, a line appears as part of the bounding box. This indicates the direction the camera object is pointing. It is provided as an aid to aiming the camera in the modeling views. Scaling a camera object's icon larger or smaller does not change the camera window's scale or perspective. The size of the icon is adjustable strictly as a convenience for handling the object in the model.

Adjusting a camera object's parameters

Once inserted, a camera object has several parameters that control the way it operates in a model. These can be altered through the Edit Primitive command in the Modeling menu.



Edit Primitive

Object type: Camera

Camera name: Camera #1

Focal length: 0.75 in   Scale: 1.0

	X	Y	Z
View from:	-2.0	2.2	32.8
View towards:	-3.0	2.4	31.9

□ Constrain camera view roll

As indicated, the Edit Primitive dialog is different for each object type. The Edit Primitive dialog box contains the fields to editn a camera object. They are discussed on the next page.

- **Camera name.** When a camera object is inserted in a model, it is given a default name, which you may use or change to suit your purposes. If the name is changed, the new name also appears in the Camera Objects sub-menu in the Windows menu.
- **Focal length.** This field contains the distance at which the camera object is focused. You may adjust the focal length of a camera object using this field or interactively in its camera window using the focal length slider. The focal length determines the amount of perspective apparent in the camera window. A high value here translates into a telephoto/low perspective view of the model in the window. Conversely, a low value indicates a wide angle/strong perspective view.
- **Scale.** This field sets the internal scale of the camera object compared to other objects in the model. The scale is expressed in the same units chosen in the Set Rulers command. The smaller the scale, the more dramatic the perspective and the less likely the camera will clip polygons as it moves through the model in an animation.

  Too small a scale can make objects appear extremely large and distorted in perspective, much like a small insect would see the human-sized world. Conversely, too large a camera scale can diminish perspective, especially as the camera becomes larger than the objects it is viewing, much like a human looking at miniature items in a doll house.

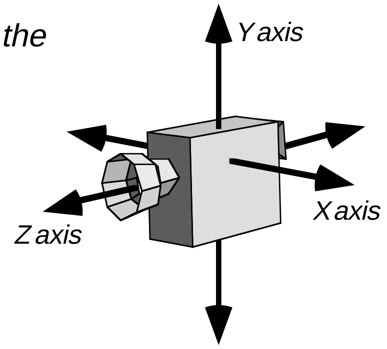
  Again, the size of the camera object icon in the modeling windows has no to effect on this field or the focal length of the view from its window.
- **View from.** This field indicates the camera object's position in the model. The values displayed on the three axes represent its location in absolute model coordinates. Adjusting these values moves the object to that location in the model, just as the Transform command can do.
- **View towards.** This field indicates a coordinate in the 3-D model space at which the camera is to point. The values are expressed in absolute model coordinates. Adjusting these values can turn the camera to point at the location specified.

  While this field identifies a specific point where the camera object is to be focused, it does not create a depth of field effect. Vision3d's snapshot camera always has infinite depth of field.

  NOTE: *Another mechanism — the Point At animation extension — is also provided to facilitate pointing a camera object at an object. It may be used to have a camera object follow another object in an animation or just point at a stationary object in a still model. The Point At command is found in the Path Filters sub-menu in the Animation menu.*

- **Constrain camera view roll.** This last check box allows you enable a steadying constraint on the view in the camera window, much like some consumer video cameras have a "SteadyCam" feature to help the moving images from the camera appear more stable and smooth in their movement. In this case, the steadying feature prevents the view from rolling onto its side as the camera object is moved at oblique angles through the model.

Relative axes for the camera object



The camera icon may be rotated (around the object's relative Z axis) at any angle, but "up" in the view in its window will always be aligned on the model's absolute Y axis. Rotation around the X and Y axes is not constrained in the view by this checkbox. Of course, the camera can still be moved and rotated anywhere you wish.

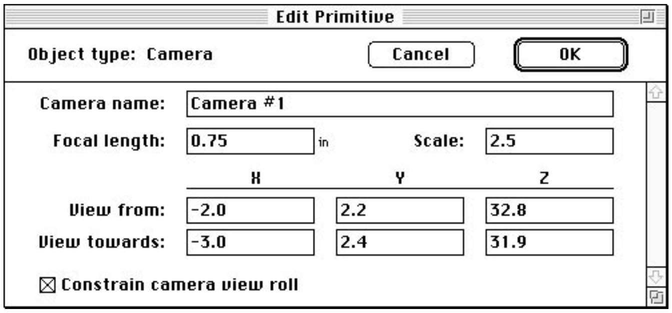
If your model is set up with a different Y axis (relative to the objects), you may wish to disable this feature so you can manually align the camera's view on the same relative axes as the objects.

Adjusting a camera object's frame size.

The Scale field in a Camera Object's Edit Primitive dialog box is discussed briefly in the description of the dialog. This refers to changing the "internal" size of a camera object as a way to adjust perspective and alter object-clipping behavior. This is a further explanation of that feature.

Consider a modeling project in which you want to move through the scene and examine some items. Most of the objects will be seen from a normal distance, but some you want to see in close detail. Or perhaps you have built a city and want to view it as if you were in an airplane flying through it.

Both of these projects are possible in Vision3d, using the camera object, but it would require reducing the camera's internal scale to avoid visible clipping of the object’s polygons as you pass closer than usual to them. This isn’t a simply making the camera object’s icon so it is smaller in the model window. It involves changing a scale setting in the camera object’s Edit Primitive dialog.



Object type: Camera

Camera name: Camera #1

Focal length: 0.75 in Scale: 2.5

	X	Y	Z
View from:	-2.0	2.2	32.8
View towards:	-3.0	2.4	31.9

☒ Constrain camera view roll

Note that this feature is meant to be used only in situations needing unusual perspectives or close-up camera positions. While most situations would not require changing a camera object’s logical size, there are some that would benefit by a smaller camera object:

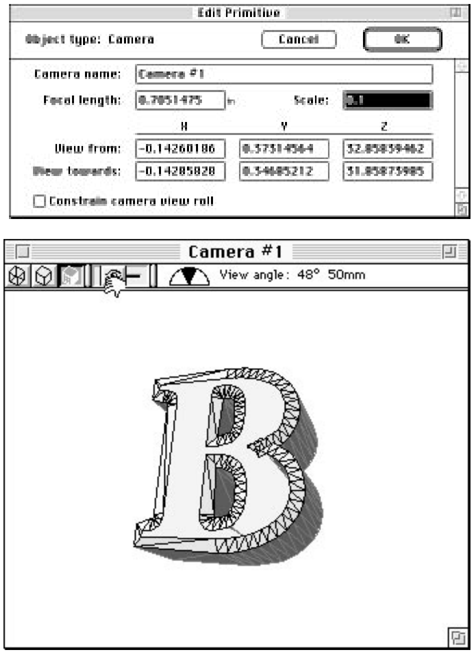
1. **Animations in which a camera object “flies through” a scene.** The smaller a camera’s logical size, the less likely it will clip through the surface of objects as it passes. If it does, the clipping can usually be made to occur behind or outside the view plane.
2. **Extreme close-ups and extreme perspectives.** The smaller the camera object, the closer you can get to an object before surface clipping occurs, thereby allowing you to obtain more extreme perspectives.

Because changing a camera object’s size radically alters the way that camera handles perspective in a scene, the control to do this is available ONLY through the Edit Primitive dialog. It is meant to be a specific, conscious change of camera scale, not an accidental modification that suddenly gives different results than you normally expect.

As indicated earlier in this chapter, scaling a camera object’s “physical” size by manipulating its icon in a modeling window has no effect on that camera’s logical size. Scaling a camera object’s logical size, however, in effect changes the camera’s film size.

For example, in the real world a 35mm camera uses a 50mm lens as its “normal” focal length; a medium format camera (with a 2.25” by 2.25” film frame) uses a 75mm lens for normal. Large format cameras (4” by 5” film frame) require all the way up to 180mm to produce a normal perspective. Smaller camera formats, like Kodak’s 110 and 126 cameras, use lens in the 25mm to 40mm range for normal perspective. This correlation between the film size and shift in a lens' focal length is same thing that happens when changing Vision3d's logical camera size.

Changing perspective by adjusting a camera object's "logical" size:



Object type: Camera

Camera name: Camera #1

Focal length: 0.7051475 in Scale: 0.1

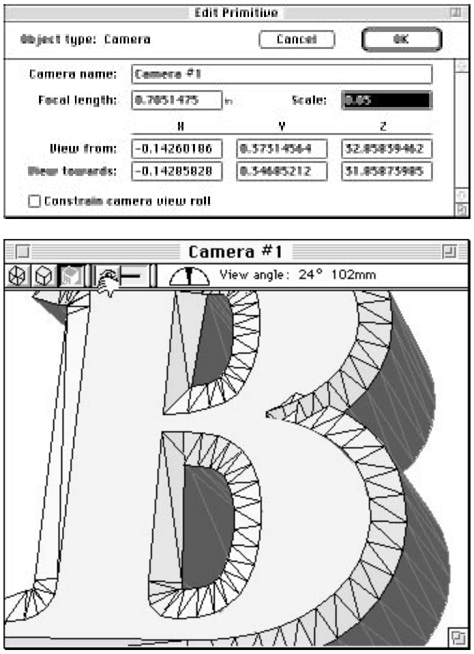
	X	Y	Z
View from:	-0.14260186	0.57514564	32.85859462
View towards:	-0.14285828	0.54685212	31.85875985

☐ Constrain camera view roll

Camera #1

View angle: 48° 50mm

This illustration shows the perspective with a standard-sized camera object using a normal (50mm) focal length.



Object type: Camera

Camera name: Camera #1

Focal length: 0.7051475 in Scale: 0.05

	X	Y	Z
View from:	-0.14260186	0.57514564	32.85859462
View towards:	-0.14285828	0.54685212	31.85875985

☐ Constrain camera view roll

Camera #1

View angle: 24° 102mm

Reducing the camera object's logical size by one-half has the effect of doubling its focal length.

In Vision3d, changing the logical size of a camera object without changing the focal length by a corresponding amount can produce some interesting results. It is usually a good idea to change the camera size and focal length by the same factor. This initially keeps the perspective in the camera window the same as before. Then you can adjust the focal length with the focal length control in the camera window for the desired effects.

Mastery of this feature will be more understandable with some experimentation. Try this with simple models first, before applying the technique to complex projects. You should be careful to write down the original scale and focal length values before attempting to change the camera size. That way, if you get lost (because the results are not as expected), you can just re-enter those values to return to a known scale and try again.

It’s all in the Camera!

Same object used in the previous example appears here as a rendered image

…Before rendering, the half-size camera object was moved closer and its focal length adjusted for the desired effect.

Note how the letter appears deeper and the 3-D perspective is more pronounced, even though no dimensional changes were made to the object.

Opening the camera window (revisited)

Obviously, the purpose of a camera object is to look at the model through its viewfinder. The window for a camera object can be opened in two ways:

1. **Double click on the camera object.**
2. **Select the desired camera object by name from a Camera Objects sub-menu in the Windows menu.**

Each camera has one window available to it, which is used to display a view of the model from the position of the camera. If the window for a specific camera is already open when an attempt is made to open it, that window is brought forward and becomes the active window in the model.

Modifier keys that apply to inserting Camera objects:

A. Shift key — Constrains the tool to draw camera objects with the proper dimensions on all axes. Without the Shift key, the icons for camera objects can be distorted.

B. Option key — Draws an object from its center, instead of the corner.

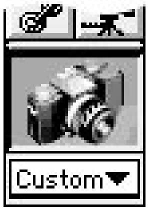
Snapshot tool

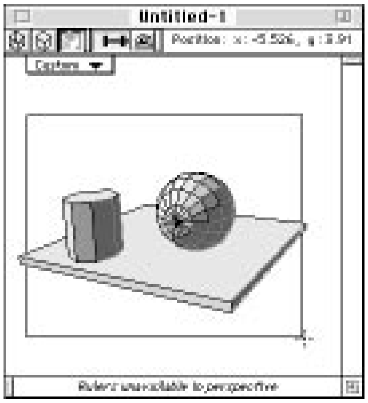
The camera tool is used to take snapshots of the model displayed in the various views of the model. Snapshots taken with the camera tool become separate rendering windows, in which the images “develop” via the chosen rendering method. Snapshots may be taken from views in either a modeling window or a camera window.

Snapshots may be initiated in one of three ways:

1. **Single click.** The size and proportion of the rendering, as well as the viewing position of the snapshot, is determined by the size of the view where the camera cursor was clicked.

2. **Cursor drag.** The size and proportion of the snapshot is determined by the marquee visible in the window from the cursor drag. The viewing position of the rendering is based on the view where the cursor was dragged.

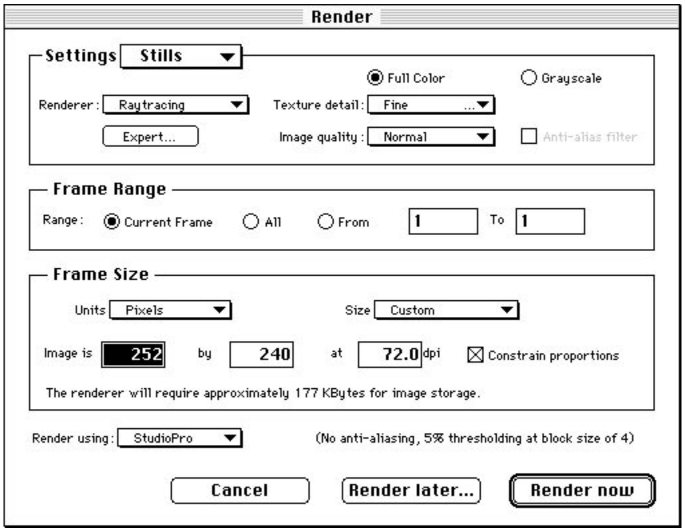



Untitled-1
Position: x: -5.526, y: 3.91
Custom
Rulers unavailable in perspective

3. **Use the Render command.** Calling this command in the Rendering menu opens a dialog that allows you to specify the complete set of rendering parameters, including image size, resolution and animation frames, if applicable, before the rendering starts.

This dialog opens when the Shift key is held down while using the camera tool

…or may be called directly using the Render command in the Rendering menu.



Render

Settings: Stills

● Full Color ○ Grayscale

Renderer: Raytracing — Texture detail: Fine …

Expert…

Image quality: Normal — □ Anti-alias filter

Frame Range

Range: ● Current Frame ○ All ○ From 1 To 1

Frame Size

Units: Pixels — Size: Custom

Image is 252 by 240 at 72.0 dpi ☒ Constrain proportions

The renderer will require approximately 177 KBytes for image storage.

Render using: StudioPro (No anti-aliasing, 5% thresholding at block size of 4)

This dialog may also be opened by holding the Shift key down while clicking or dragging the tool in a view. The viewing position for the rendering is taken from the currently active view.

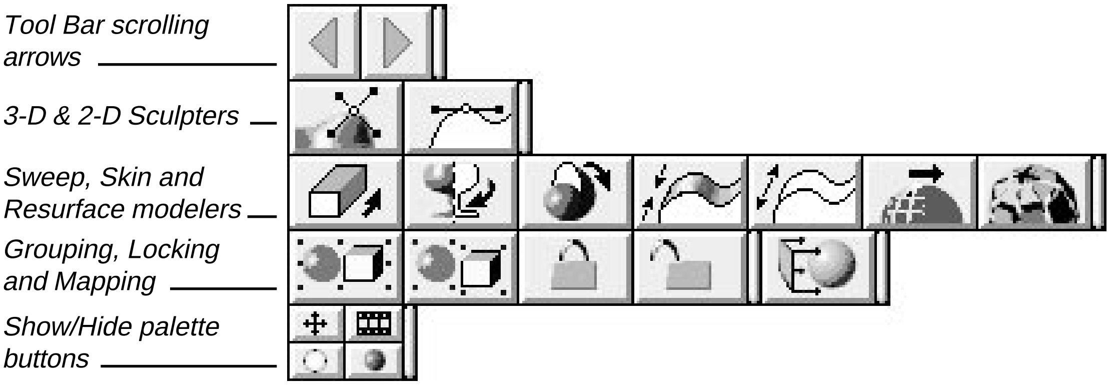
Modifier keys that apply to initiating renderings using the Camera tool:

A. **Shift key** — Opens the Render dialog box as part of the snapshot process to set or adjust additional rendering parameters. Without the Shift key, clicking or dragging the camera tool immediately begins a rendering.

B. **Option key** — Draws the camera's cursor drag marquee from its center, instead of the corner.

Button bar

Vision3d initially displays a Button bar that provides buttons for the most often used operations and modelers. You can display or hide the Button bar using the Show/Hide Button bar command found in the Palettes sub-menu, in the Windows menu. If the Button bar is hidden when you quit Vision3d, it remains hidden the next time the application is launched.


Tool Bar scrolling arrows
3-D & 2-D Sculpters
Sweep, Skin and Resurface modelers
Grouping, Locking and Mapping
Show/Hide palette buttons

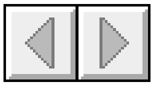
Buttons on the Button bar appear from extensions installed in the Extensions folder. These extensions may be modelers, animation enhancements, or simple additions that call one or more commands from the menus. Thus, you will see buttons on the Button bar not only for modelers like extruding and lathing, but also for simple commands like Group and Ungroup.

NOTE: *Each of these extensions also appear as commands in the menus. You may find the Button bar to be more convenient, but you do not have to use the Button bar only. If the Button bar is hidden, you can still access all of the features through the standard menu commands.*

The Button bar is neither moveable nor customizable at the user level. No mechanism is provided for you to create your own macros and attach them to a button. The button resource must be included by developers of the various extensions at the time they are written.

Multiple rows of buttons

As mentioned, the number of buttons on the Button bar depend on the modeling and animation extensions currently installed in the Vision3d Extensions folder. If more buttons are present than will fit in the width of your screen, the horizontal arrow buttons cycle the buttons present to allow you to view more than one row of buttons on the bar.



Left and Right scrolling arrows

Operation

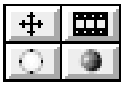
You must use the mouse cursor to engage a Button bar button. There is no other mechanism for “clicking” the Button bar buttons. When a button is clicked on, it depresses to provide a visual feedback. This indicates that the mouse click is accepted and the operation connected to that button is executed.

Buttons may be active or dimmed just like commands in menus, depending on the current situation of the active model. If an extension represented by a button is usable, the button will be active. If that extension cannot be used, the button will be dimmed.

Show/Hide palettes buttons

In addition to the tool buttons themselves, four special buttons are visible on the extreme right of the Button bar. They are quarter-size to take up little room and are provided as a convenient way to quickly show or hide the other palettes.

Show/Hide toggle buttons



Tool palette

Lighting palette

Animation palette

Texture/Shapes palette

These special buttons perform the same function as the other Show/Hide palette commands found in the same Palettes sub-menu as the Show/Hide Button bar command.

Modeling Extensions

Installed modeling extensions are accessed by commands that appear at the bottom of the Modeling menu, as well as buttons on the

Button bar. Vision3d includes several extensions in the package as part of the standard product. As other extensions become available and are installed in the Extensions folder, they will appear as commands in the Modeling menu and display buttons on the Button bar.

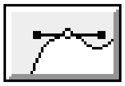
Extensions provide their own dialogs and menu bar allowing the construction of new objects, for converting existing objects to the type supported by the modeler, or for further editing by the same modeler. All necessary tools required to perform the work for which the modeler is designed are also provided in the dialogs. When editing the object is complete, the extension modeler inserts the object into the model space. You may then move, rotate or scale the object just like any other object in the model using the object manipulation tools provided on the tool palette.

As discussed earlier in this chapter, double-clicking on an object calls the Modify command, which uses the object's native modeler to edit its geometry. Whenever an object is edited with an extension modeler, it is tagged so Vision3d can identify the modeler that was used to create or last modify it. Double-clicking on an object to activate its native modeler provides a consistent interface response for all objects, even though they were created using different methods.

The standard modeling extensions are discussed in this section of the chapter. Other after-market or third-party extensions include their own documentation explaining how to use the modeling techniques provided in that extension.

2d Sculpter

The 2d Sculpter modeler is an extension provided to allow reshaping 2-D objects after they are drawn in the model. This extension works with only one 2-D object or 2-D group at a time.

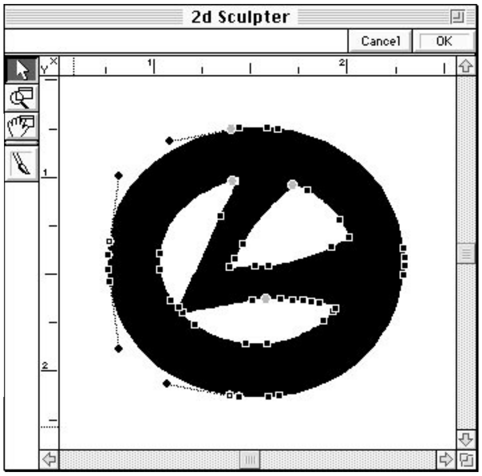


When present in the Vision3d Extensions folder, a 2d Sculpter command to use this extension appears in the Modeling menu. Additionally, the above button appears on the Button bar.

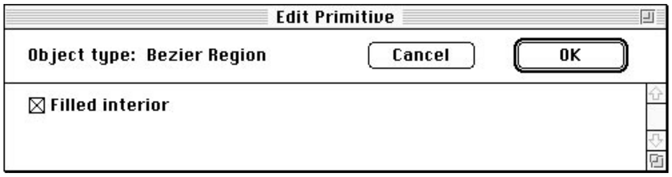
Description

As the name implies, 2d Sculpter allows you to adjust the shape of existing 2-D objects. You may want to use this modeler to prepare an object for use in another modeler, such as the Lathe or Extrude extensions, or for fine-tuning the ribs of a FlexObject for skinning. Both polygonal and bézier type objects can be edited in this extension.

The modeler allows you to convert an object's type from polygonal to bézier splines, and vice-versa. Usually, it is more desirable to work with a bézier description of an object, since that object type requires less memory and offers more accurate control of the geometry during editing.



If the 2d Sculpter command or button is dimmed, the selected object does not qualify for editing with this extension. The reshaping is available only when a single 2-D object or a single object in a 2-D group is selected. Typically, you would use the reshape extension on one object at a time, but you can edit multiple 2-D objects in certain cases. 2-D groups are special cases. A 2-D group is created when two or more 2-D objects ON THE SAME PLANE are grouped together. You can check the type of a selected object by choosing the Edit Primitive command.



Object type: Bezier Region

☒ Filled interior

Currently, this type of group occurs only during importing an Illustrator or PICT image with discrete multiple objects in the files. You cannot create a 2-D group by manually selecting and grouping 2-D objects. If multiple 2-D objects are grouped by the user, they are always considered a standard group. A standard group of objects cannot be edited in this extension, even though they may be all 2-D in type. Ungrouping the objects and selecting the individual 2-D objects within the group will allow them to be edited. This is the recommended technique.

Sometimes a single object can be grouped by itself, making it unavailable for editing with this extension. In that case, you can use the Ungroup command to remove the grouping shell from the object and it will then be recognized by the 2d Sculpter extension.

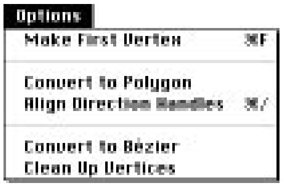
Opening a 2d Sculpter window

Calling the 2d Sculpter extension opens a separate window containing the selected object. By isolating the object in this manner, it can be presented in a larger format and made easier to edit.

Different menu bar

While the reshape window is open, Vision3d's standard menu bar is replaced by one that supports the functions of the extension. The extension's menu bar contains only four menus.

The first two menus in this menu bar are dimmed because no commands are active in them while in this extension. The Edit menu contains several commands usable in the 2d Sculpter window: Undo, Clear, and Select All. Note that the Cut, Copy and Paste commands are NOT available while in this extension.



The Options menu is specific to the 2d Sculpter extension. It is divided into three sections: 1) a command that applies to all object types, 2) commands that apply only to bézier objects, and 3) commands that apply only to polygonal objects. The commands and their uses are presented later in this modeler discussion.

Special "Fit to View" feature

There is a special feature that requires special mention here. It is a "Fit to View" operation that can facilitate handling objects in the 2d Sculpter. Note that this feature is not represented by a command in the menu bar. **IT CAN BE EXECUTED BY PUSHING THE COMMAND-EQUAL (⌘=) KEY COMBINATION ONLY.** You may recognize this as the same keystroke shortcut for the "Fit Views to All" command in Vision3d's Windows menu in the main menu bar. This functionality is implemented here in the 2d Sculpter extension as well.

As you edit an object, you may zoom in to work on a specific area, then want to see the entire object in the view again. This is a quick way to accomplish the task, rather than use the zoom out option on the magnify tool.

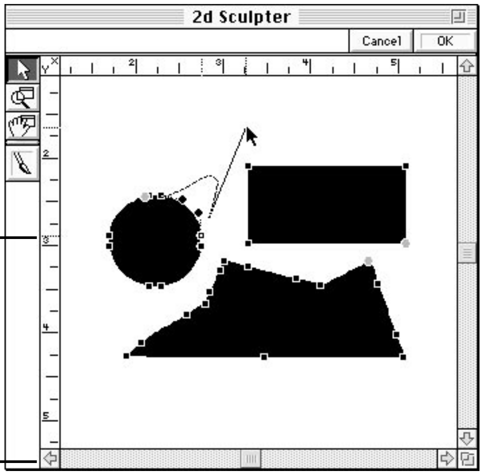
Using this feature has two additional benefits: It automatically re-centers the object on the workspace and expands the workspace's dimensions to fit the window, if necessary. This is especially useful if you have dragged some of the vertices outward to the edge of the workspace, this provides more room to move them further if desired, because the workspace will be made larger automatically to accommodate the larger object.

How to use

The selected object appears in the 2d Sculpter window centered on the workspace and scaled so it is completely visible in the extension's window. The workspace in this window is set up to be approximately four times larger than the object. Several controls and tools are provided in the window frame.

If the object being edited is described as a polygonal object, handles appear on each vertex point of the object upon opening the 2d Sculpter window. You can reshape the geometry of any object by simply clicking on a vertex point and dragging it to the desired position.

If the object is bézier, there are usually fewer points. These points can be moved just as points on a polygonal object, but each point also has direction handles attached which allow you to bend the edges of the object if it is closed or the line segments between points if it is an open polyline.



Pointer tool
Magnify tool
Grabber tool
Sculpt tool
Rulers
Scroll bars

An explanation of the controls and tools follows:

- **Rulers** in this window use the same units as the model from which the object came and indicate its actual size regardless of the view scale. The zero point for the rulers is in the upper left corner. It is not movable.
- **Scroll bars** are also provided to move around in the workspace, or you may use the Grabber tool to shift the position of the object in the view.
- **Pointer tool** allows you to select one or more vertex points and drag them on the workspace.
- **Magnify tool** allows you to zoom in or out. This means you can move in close for delicate work on a specific area of the object or move back out to look at the entire object. The magnify tool works similar to the same tool in the main modeling windows. Clicking in the view zooms in on the area where you click. Holding down the Option key zooms out when the tool is clicked. It does not have a variable zoom, however.
- **Grabber tool** allows you to "grab" the workspace and move it so a different area is visible within the view. This is the same as using the scroll bars, but allows you to move the workspace diagonally in one step.
- **Sculpt tool** allows you to add to or remove from the surface area of an object. This is the method you would use to punch a hole through an object.

  Choosing the Sculpt tool opens a secondary window in which you are able to perform the desired "sculpting" operations on a pixel-level representation of the object. After finishing there, you are returned to the first window for any further adjustments via pushing and pulling vertices. The operation of the tools in the secondary window are discussed later in this section, after the vertex-level editing techniques are presented.

Vertex editing

More than one vertex point may be selected and moved at one time. Multiple vertices can be selected either by dragging a marquee around them or by clicking on the points one at a time while holding the Shift key down. Subsequent dragging of one of the selected vertices moves all selected points in unison. If all vertices are selected, moving one vertex point moves the entire object.

Additionally, an object can be moved (without having to select all vertices first) by clicking and dragging anywhere on the object.

To de-select all vertices, click in the window anywhere away from the object.

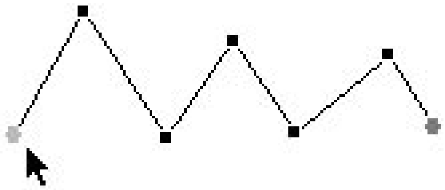
Note that when an object appears in the reshape window, it is stripped of rotational transformations. In other words, if the object has been rotated in the model space prior to being editing with the 2d Sculpter extension, the rotation is temporarily removed from the object. The object then can be edited in its "native" state. Its transformations are re-applied when the object is returned to the model, after your work in the reshape window is done.

While transformations are time-dependent, i.e. — they can be applied over time, vertex positions on objects are not. An object may be scaled or rotated as part of an animation (see *Chapter 7: Scripting Animations*), but reshaping is fundamental to the basic geometry of an object.

Typically, you would use the 2d Sculpter extension BEFORE doing any animation-related scene assembly.

Changing the beginning vertex point

To change the beginning vertex point to a different location on the object, just select the desired vertex point and choose the "Make Beginning Vertex" command from the Options menu.



Beginning and ending points are indicated by round handles — green for beginning and red for ending.

All 2-D objects have a beginning and ending vertex point. The location of the beginning point on an object is usually determined by the sequence in which the vertex points are entered into the object as it was drawn in the modeling window. Some tools just arbitrarily pick a point.

The beginning and ending vertex points appear in the 2d Sculpter window as round, colored handles on the object; all other vertices are square and black. The green handle is the beginning point; the red handle is the ending point.

While you usually do not care which vertex is the beginning point, there are some situations in which you may need to shift the beginning point to align it with beginning points on other 2-D objects. The Skin modeling extension is one situation where the position of the beginning vertex point is very important to the result.

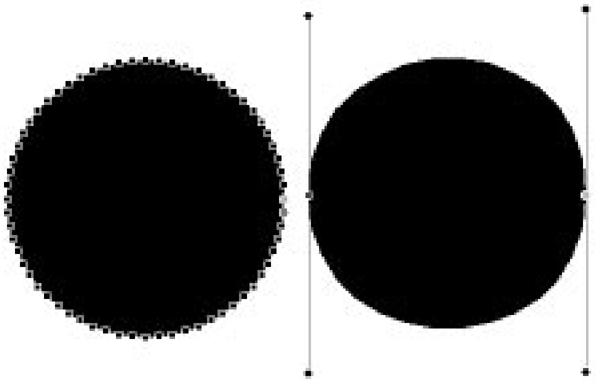
Converting a polygonal object to bézier splines

Some objects are created as bézier splines when they are drawn in the modeling window. The Text and Pen drawing tools are examples that create bézier spline objects.

Other objects are created as polygonal descriptions. The Rectangle and Irregular Polygon tools are examples that create this type of object. While polygons can be edited in the 2d Sculpter extensions just as easily, most objects work better as a bézier description. Polygonal objects must be converted to bézier before you can edit them as splines.

Example of the same object in polygonal form (left) and bézier form (right).

Not how bézier form requires only two vertex points to describe the circle, while the polygonal equivalent needs 64.



To convert a polygonal object to a bézier object, select one or more vertices on the object and choose the "Convert to Bézier" command from the Options menu. The command is dimmed if there is no polygonal object with selected vertices.

Note that the entire object is converted to bézier, not just those vertex points that are selected. An object must be entirely bézier or entirely polygonal.

Converting a bézier object to polygonal representation

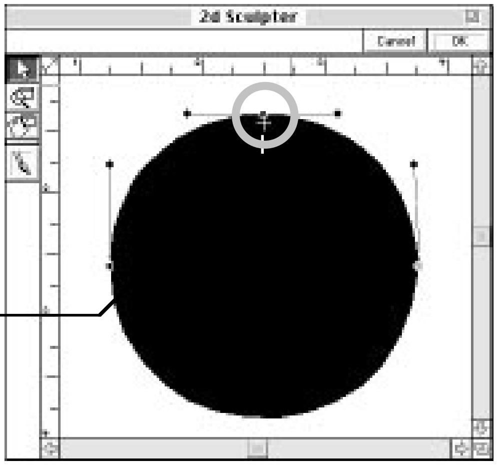
Occasionally, you may wish to convert a bézier object to polygonal form. To do this, simply select one or more of the object's vertex points and choose the "Convert to Polygon" command from the Options menu. The command is dimmed if there is no bézier object with selected vertices.

Removing and adding vertex points

As part of the reshaping process, in addition to moving vertices around you may need to remove some vertices or perhaps add more to accomplish the geometry changes desired. You may add or remove vertices from either type of object — polygonal or bézier — in the same manner

Note that you cannot delete all the vertices for an object while in the reshape extension. By definition, that would mean the object has been deleted from the model. The 2d Sculpter extension does not have the “authority” to take that step. While in the reshaping extension, a minimum of two vertex points are required for either a line or a closed bézier object; three points are required for a closed polygonal object. To delete the entire object, you must return to the model and remove it from the model at that level.

Removing vertices is straightforward. Simply select the points you wish to remove and push the Delete key. The object will be re-drawn to show its new form. Of course, the Undo command is enabled during any operation, so if the resulting form is not as expected you can simply undo the deletion and try something different.



Holding down the Option key allows you to add vertex points to a line or object edge.

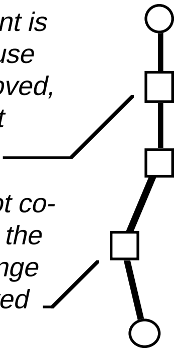
Note how the cursor changes shape when it intersects a line or edge while the Option key is held down.

To add a vertex point, hold down the Option key while positioning the cursor on an edge or line between two other points. When the cursor changes to a plus (+) shape as it intersects an edge or line, simply click the mouse button to add a vertex point. Unless the cursor changes to a plus, there is no edge detected and clicking the mouse button will not add a vertex point.

As an added convenience, you can add a vertex point and begin dragging it immediately. You do not have to release the mouse button between adding the point and dragging. When the mouse button is released, the window re-draws to show the new point and any changes in the geometry.

Removing co-linear points

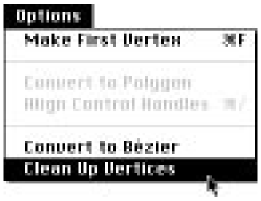
Occasionally, when a polygonal object is extensively edited in the 2d Sculpter extension — i.e., you have added several vertex points and/or radically changed the geometry — you may find that some of the vertices have become co-linear, and therefore redundant. Co-linear points are those points that lay on a line or object edge that is already defined by two other points. They are considered redundant because their removal would not change the form or geometry of the object in any way.



This vertex point is co-linear because when it is removed, the line doesn't change shape

This point is not co-linear because the line would change shape if removed

The reshape extension has a provision to examine and automatically remove any co-linear points on polygonal objects. This is initiated by choosing the "Clean Up Vertices" command from the Options menu. If it finds any co-linear vertex points, they are deleted.

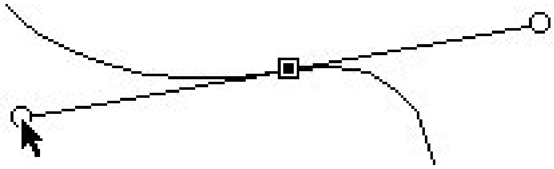


Note also that when you click on the "OK" button to complete the editing of a 2-D object, and the object is in polygonal form, this same process occurs and any co-linear vertices are removed before the object is returned to the model. This is done to save memory and to insure an object is described as efficiently as possible.

Obviously, bézier objects can also have co-linear points in its geometry. However, bézier lines or object edges are described in a completely different way than their polygonal counterparts. The Clean Up Vertices command is unavailable when the object being edited in 2d Sculpter is bézier. That means if you want an object to retain all the vertices you have configured for it, you should work in bézier form.

Working with bézier splines

The interface developed in Vision3d's 2d Sculpter extension for editing bézier splines is consistent with those used in other stand-alone drawing programs, such as Adobe Illustrator®, Aldus® FreeHand™, or Deneba Systems' Canvas®.



As with those applications, the vertex point determines the absolute position of the line path or object edge at it passes through that point. Further, each vertex point has two direction handles that control or influence the curve of the line segment between it and the next point on either side. The farther a direction handle is pulled out from its vertex point, the more force it applies to its line segment to bend or curve it. Direction handles are moved by clicking and dragging them.

Dragging out zero-force direction handles

If either one of the direction handles for a particular vertex point is not visible, it is because it has a zero force and is co-incident (on top of) the vertex point. No bending force is being applied to the line segment on that side of the point. Sometimes that is the desired appearance, but there may be times when you want to reshape that part of the line or object.

To pull a direction handle off of a point, you must hold down the Control key while clicking on and dragging that point. With the Control key down, the vertex point remains stationary and the direction handle is selected and dragged instead. If both direction handles are co-incident with the vertex point, the topmost handle is dragged out first. To get the other handle, you must repeat the process. There is no discernible pattern as to which direction handle is topmost when both are co-incident. It varies with the way this object is stored internally.

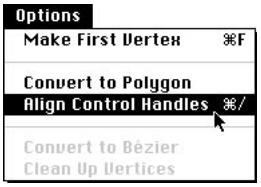
Once a direction handle is away from a control point, it can be clicked on and dragged without using the Control key. The Control key modifier can also be used if a direction handle from one vertex point happens to fall on top of another point. Without the Control key, clicking on the direction handle will always select the point instead, since points are given precedence to direction handles.

Aligning direction handles

Vertices usually have direction handles that are aligned or tangent to the line passing through that point. When the direction handles are aligned, they are hooked together. Adjusting one handle also moves the one on the other side.

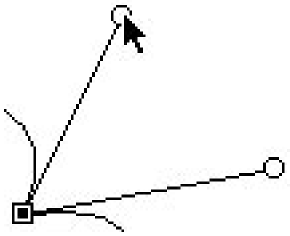
If a vertex point is "hinged" (described later), you can bring the direction handles back into alignment in one of two ways:

1. **Move one of the handles until it is visually aligned with the one on the other side of the point.** When you release the mouse button, the direction handles become linked together. If you move one of the direction handles again, the other follows.
2. **Select the point and choose the "Align Control Handles" command from the Options menu.** The positions of both handles are averaged and they are adjusted to a common tangent where they can align with each other. This method may alter the curve of the line on both sides of the vertex point.



Hinged direction handles

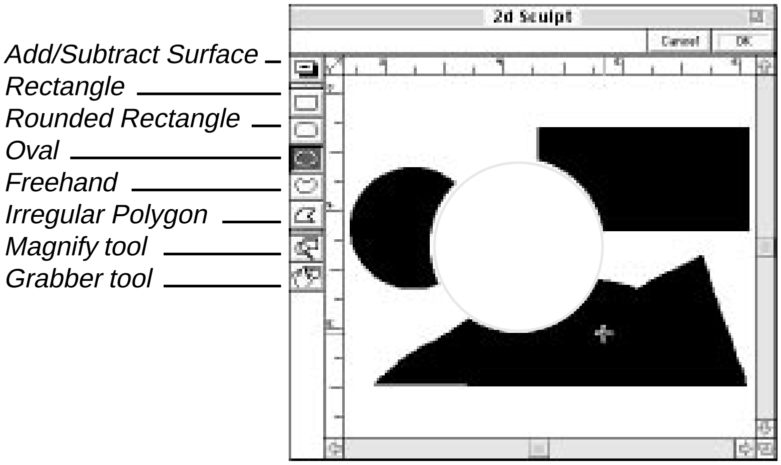
Vertex points can be hinged so that the line or object edge is free to make sharp angles at the point. When a vertex is hinged, its direction handles can be moved independently to adjust the curve of line segments separately on either side of the point. Moving one direction handle has no effect on the other handle.



To hinge a vertex point, hold down the Option key while dragging one of the direction handles. You will be able to drag it without the opposite handle moving. Once the alignment between direction handles is broken for a particular point, either handle can be subsequently moved freely without using the Option key again.

Sculpt tool operation

It is important to understand the process of switching between the vertex-level editing view and the sculpting view. Upon choosing the Sculpt tool, the entire object is temporarily converted to a bitmap representation, similar to a pixel image in a paint program. The tools in the secondary Sculpting tool window are designed to operate on the pixels rather than the vertices of the object.



Note that while the Sculpt tool window is secondary to the main 2d Sculpter window, it has its own title bar and may be moved independently on the desktop. However, it is a modal window, just like the main window. You cannot perform any other operations outside this window while it is open.

Most of the controls provided in the secondary Sculpt tool window are identical to the main window, i.e. — the rulers, scroll bars, magnify tool, and Grabber tool operate the same in both windows. All of the drawing tools in this window create closed shapes. Those special tools available in this view to add or subtract surface to an object are as follows:

- **Add/Subtract Surface.** Click on this tool to toggle the behavior of the other tools. When the plus symbol (+) is visible, the tools below it add surface to the object. This can be used to fill a hole within the boundaries of object or add to the edge of the object. When the minus symbol (-) is visible, the tools remove surface from the object. This can be used to make holes within the object or "shave" areas from the edge of the object.

- **Rectangle.** Use to add or subtract surface in a rectangular pattern. The operation is performed by clicking and dragging diagonally. A marquee appears to indicate the size and proportion of the rectangle. Holding down the Shift key constrains the marquee to a perfect square. Releasing the mouse button completes the drawing process.
- **Rounded Rectangle.** Use to add or subtract surface in a rounded rectangular pattern. This tool operates exactly the same as the square cornered rectangle tool.
- **Oval.** Use to add or subtract surface in an oval pattern. The operation is the same as the previous two tools, except that holding down the Shift key constrains the marquee to a perfect circle.
- **Freehand.** Use to add or subtract surface in a freehand pattern. This tool behaves differently than the previous three tools. The operation is performed by clicking and dragging along the edge of the shape you wish to draw. A line appears to indicate the shape as it is drawn. Releasing the mouse button completes the drawing process. If you do not return to the beginning point of the object being drawn, a straight line is added to the form you draw to close the shape.
- **Irregular Polygon.** Use to add or subtract surface in an irregular pattern whose edges are straight lines. This tool behaves differently than any of the previous tools. The operation is performed by clicking once at each point where you wish to change direction. A line appears to indicate the shape as it is drawn. Holding down the Shift key constrains each line segment's angle to 45° increments. Double-clicking the mouse button on the last point completes the drawing process. If you do not return to the beginning point of the object being drawn, a straight line is added to the form you draw to close the shape.

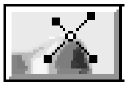
Clicking the OK button in the Sculpt tool window converts the black pixels of the image back into an object surface; white pixels become holes if inside the object or empty space if outside the object.

NOTE: *Converting an object's geometry to pixels, then back into an object again may result in a slight shift in the geometry. However, any discrepancies can usually be easily adjusted when back in the main 2d Sculpter window. If you anticipate using the Sculpt tool during a 2d Sculpter session, you should use it first, then fine-tune the object geometry after. If you work on the geometry first, then use the Sculpt tool and have to redo your work after it comes back, you can often end up wasting time.*

3d Sculpter

The 3d Sculpter modeling extension is provided to edit 3-D objects at the vertex level. It accepts both bézier spline and polygonal mesh 3-D objects, although there is a difference in the way the two different types of objects can be edited. More than one object can be edited at a time, but multiple objects must be put in a group first.

A group can be any combination of objects in any spatial arrangement. However, the more objects in the group brought into the modeler at one time, the more memory required to edit them. Too many objects can be self defeating, i.e. — it becomes more difficult to navigate in the window and slower to work. It is usually best to work on just one object at a time.



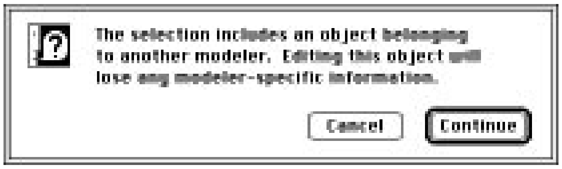
When this extension is present in the Vision3d Extensions folder, a 3d Sculpter command appears in the Modeling menu. The above button also appears on the Button bar.

Description

3d Sculpter allows you to modify the geometry of 3-D objects after they have been drawn (as a primitive) or constructed with another modeler. It provides a three-dimensional workspace where you can push and pull vertices on any axis to mold an object's geometry in complex, organic ways.

It is designed to work with the surface of objects. To that end, only objects with a surface can be brought into this modeler. So, while an open 2-D polyline cannot be edited in this modeler, the 3-D surface created by extruding or lathing that polyline can.

Filled 2-D objects can also be edited in this modeler, allowing you to take a flat object and warp or bend them in ways not available with other extensions.



The selection includes an object belonging to another modeler. Editing this object will lose any modeler-specific information.

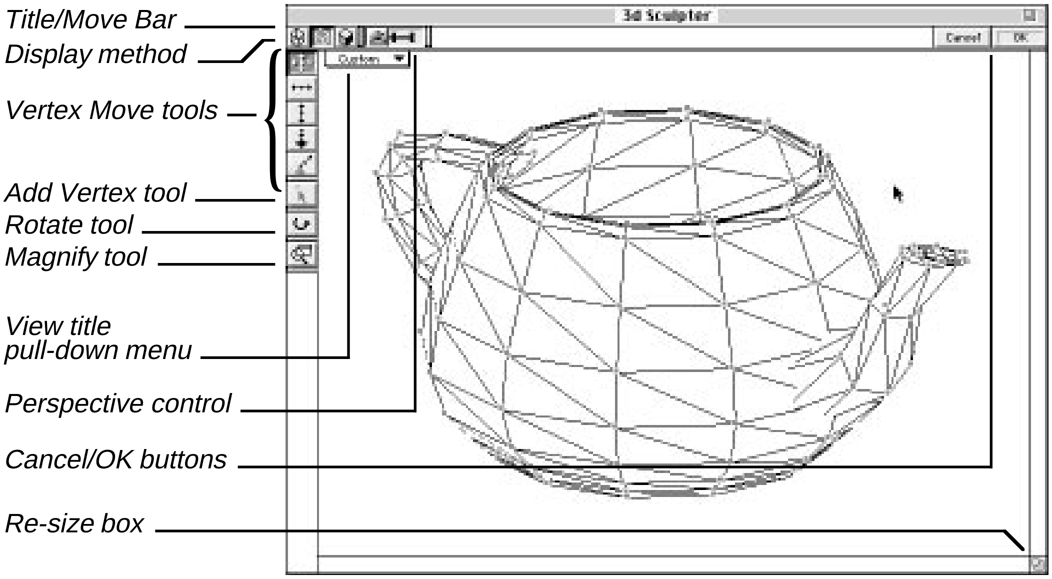
When an object constructed in another modeler, i.e. — extrude, lathe, etc., is selected and this modeling extension is called, an alert appears warning you that the link to the object's original modeler will be broken. That means, an extruded object cannot be re-extruded once you edit it with this extension. Attempting to edit an instance of a shape also presents this alert.

This alert gives you the option to continue or quit. If you continue, the link to its original creator will be broken and the object is presented in a 3d Sculpter window for editing. However, the change in modeler linkage, as well as any geometry changes you make in this modeler do not take effect until you click on the OK button. If you click on the Cancel button to exit this modeler, the object is left untouched.

Of course, choosing to edit a primitive in this modeler does not display the alert, since primitives do not have any linkage to a previous modeler.

Window controls and modeler tools

Upon opening the 3d Sculpter modeler, the selected object is presented for alterations. A single view is contained in the window. You may switch the view orientation through the view title pull-down menu, a mechanism similar to the views found in modeling and shape windows. This is very useful as you work with an object — selecting specific vertex points, establishing drag planes for the vertices, and of course checking on overall geometry. The Isometric view is usually favored for the checking phase.



If the object is bézier in type, you will also see the vertices currently present on the surface of the object. Polygonal objects also have vertices, but only show when selected. Vertices are used to manipulate the object's surface.

The modeler's window contains several controls and tools in its frame to facilitate this editing process. Many of these controls used in the 3d Sculpter window are common to other windows in other areas of the software and should already be familiar to you. A description of the window controls follows:

- **Title/move bar.** Allows movement of the 3d Sculpter window on the desktop.
- **Resize box.** Allows re-sizing of the window as desired. The view within the window re-draws to center the object in the view, even if the window is too small to show the entire object. Re-sizing the window will not change the scale of the view.
- **Display method.** Allows you to choose from wireframe, hidden-line removed, or quick shading to display the object. The three buttons in the top left corner of the window operate exactly as they do in the main modeling window, as discussed in chapter 1 under the "Modeling window and its controls" topic.

  Choosing wireframe shows also surfaces of the objects, even those that are on the backside. It also shows all vertices, including those that would normally be hidden.

  Choosing the hidden-line removed method shows only the front surfaces. Additionally, only the front vertices are visible and selectable.

  Choosing the quick-shaded method provides additional lighting and shading information to help you better understand how the object is oriented and how its surface is smoothed. This is especially useful with bézier objects.
- **Perspective.** Allows you to choose from orthographic, normal, or wide angle perspective. The three-position perspective control also operates in the same manner as it does in the modeling window. Please refer to chapter 1, also in the same topic as the display method, if you need to refresh your memory.
- **View title.** Allows you to choose from seven home positions listed in the pull-down menu: Front, Back, Top, Bottom, Left, Right, and Isometric. This control, too, works the same way as the view title control in modeling window view(s).

The 3d Sculpter window also contains vertex move tools that are unique to this modeling extension. Each performs a specific function. A description of these vertex-editing tools follows:

- **General Vertex Move tool.** Use to move one or more vertex points in any direction. As the tool is dragged, all currently-selected vertices move in the same direction. Holding down the Shift key constrains the movement horizontally or vertically in the view. While this tool does not move on the Z axis, choosing a view perpendicular the current one allows you to effectively move vertices on that axis.
- **Horizontal Move tool.** Use to move one or more vertex points horizontally in the view. All currently-selected points move in the same direction. No modifier key is required, since the tool is self-constrained.
- **Vertical Move tool.** Use to move one or more vertices vertically in the view. This tool works the same as the previous tool, except it is self-constrained to move only vertically, relative to the view.
- **In/Out Move tool.** Use this tool to move the selected vertex point(s) in or out relative to the view. While the three previous tools work only on a plane parallel to the view, this tool works only perpendicular to the view.
- **Surface Normal Move tool.** Use this tool to move the selected vertex point(s) in a direction dictated by the surface normal for each point. All previous tools worked relative to the view. This tool works relative to the object's surface.

  If more than one vertex point is selected, depending on the surface geometry each point could move in a different direction. However, they will all move the same distance.
- **Add Vertex Point tool.** Use to add another vertex point on the surface of the object. Adding a point causes the surface patch on which the point is added to sub-divide into more (and smaller) patches. Depending on where the point is added, up to four additional patches may be created. This process is explained later in the discussion of this modeler.

  This tool is only functional for bézier type objects. Polygonal objects cannot add or delete points from their surface.
- **Rotate tool.** Use to rotate the object in the view. Rotation is relative to the view direction. You may rotate an object two ways:

  1. By clicking and dragging on one of the rotation handles that appear around the object(s) in the view (when the Rotate tool is selected). These handles operate exactly the same as object rotation handles in modeling windows. This method automatically constrains rotation to only one axis at a time, depending on the handle you choose to drag.

2. By clicking and dragging on the surface of the object(s). This effects a trackball-style rotation, which allows rotation on multiple axes at one time in the direction and amount the cursor is dragged. This type of rotation is unconstrained, i.e. — no modifier keys are used to limit the amount or direction of rotation.

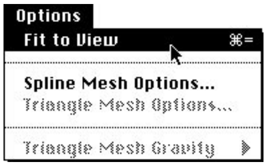
When an object is rotated, the view title changes to show “Custom.” Again, this is similar to the behavior of view titles in the standard modeling windows.

To return the view to one of the seven home positions, after a rotation operation, click on the view title and choose one of the orientations.

- **Magnify tool.** Use this tool to zoom in or out in the view. This allows you to look at portions of an object’s surface close up for more delicate work, then move back out to look at the whole object again.

  This magnify tool works exactly as the Magnify tool on the Tool palette does in the model windows. Click once with the plus symbol (+) showing to magnify two times (2X), Click once with the minus symbol showing to reduce by one-half (1/2X). The plus symbol is the normal setting; the minus symbol (-) appears by holding down the Option key. Holding down the Shift key allows variable zooms.

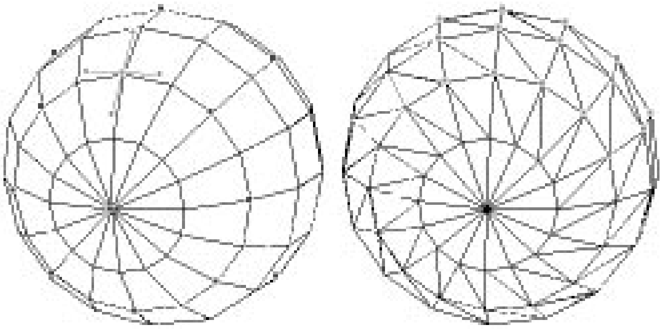
  The point at which you click the tool in the view is the area magnified. Note that there are no scroll bars. When you zoom in on a particular area of an object, if you wish to work in another area, if it is not visible you must zoom out again until it is visible before you can choose that area.



  To facilitate the need to zoom out, a “Fit to View” command is provided in the 3d Sculpter’s Options menu. Choosing this command resets the view scale to show all of the object(s). You may access this command from the menu directly or use its keystroke equivalent of Command-Equal (⌘=). You may recognize this command key sequence as the same one used for the “Fit Views to All” command in Vision3d’s main menus.

Bézier surface patches vs. Polygonal meshes

As mentioned, bézier objects are handled differently than polygonal objects when brought into 3d Sculpter. The illustration on the next page shows the obvious differences in appearance.



Bézier Object:
- *Rectangular surface*
- *Vertices always show*
- *Direction handles on selected vertices*

Polygonal Object:
- *Triangulated surface*
- *Vertices show only when selected*

The first thing you notice is that bézier objects always show the vertices available to edit their surfaces, while polygonal objects do not have any vertices showing at all. Of course, polygonal objects have vertices as well; they are at the intersection of the triangulated polygons. But, because there are usually so many, to avoid confusion they only show when selected.

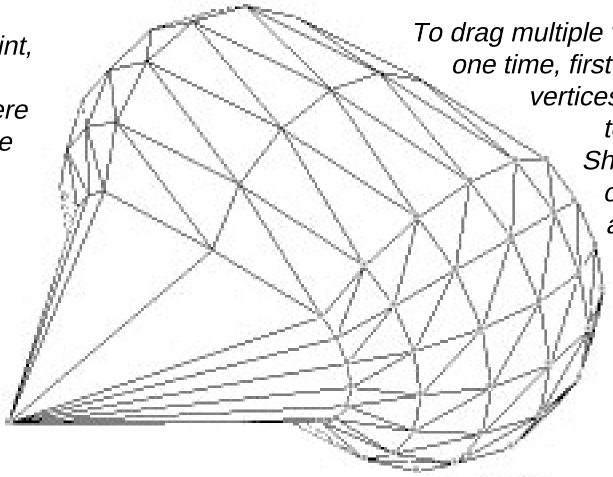
The bézier vertices are blue normally, and turn red when selected. Polygonal vertices appear red when selected. Bézier vertices have direction handles attached, which are used to bend or warp the surface patch(es) controlled by the selected vertex points. Polygonal objects do not have direction handles. When a bézier vertex is moved the adjacent surface patches follow in a spline-like fashion. Moving polygonal vertices simply stretches or contracts adjacent polygons in the mesh surface.

The most important difference in the handling of bézier or polygonal objects is that you can add and remove vertex points to a bézier object. However, vertex points cannot be added or removed on a polygonal object; you have to work with the mesh as it exists.

Modifying polygonal objects

Altering the geometry of polygonal mesh surfaces is quite straightforward. Simply click on one or more vertex points and drag any direction. The affected polygons are recalculated and the surface is re-drawn to show the changes.

The Undo command is available to return the object to its previous state, but it must be used immediately after the vertices are dragged. Of course, if you make a serious error and cannot correct it, you can always click on the Cancel button to close the 3d Sculpter window and return to the model; leaving the object untouched.

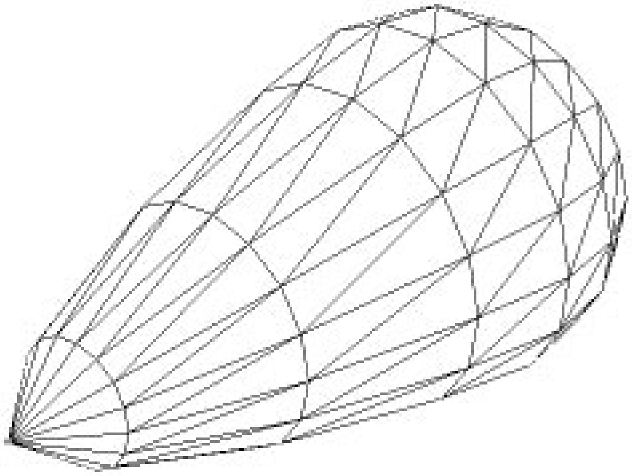


You can pull more than one vertex point, like stretching this polygonalized sphere horizontally to make an ovoid.

Dragging a single vertex without gravity produces sharp spikes

To drag multiple vertices at one time, first select the vertices you wish to move by Shift-clicking or by using a selection marquee

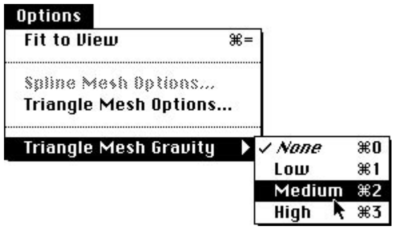
…then, drag them in unison.



But with gravity turned on, the results are more gradual.

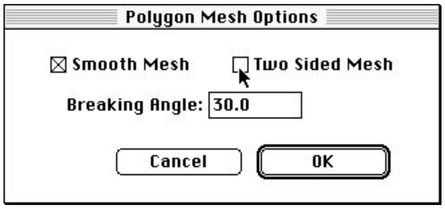
Note how the distance other vertices move lessens with their distance from the vertex being dragged

Of course, pulling a single vertex often results in sharp spikes coming out of the object. If you want a more rounded effect on polygonal objects, you can turn on gravity. With gravity enabled, vertex points closest to the point being dragged move a greater distance than those farther away.



Gravity is enabled by choosing “low,” “medium,” or “high” in the Triangle Mesh Options sub-menu, in the 3d Sculpter’s Options menu. Gravity is turned off by choosing “none” in that same sub-menu. Note the keystroke equivalents also assigned to these options.

Additionally, you have the option of changing the polygons used to make a polygonal object from single-sided to double-sided, and vice-versa. This is available through the Triangle Mesh Options command, also in the Options menu.



☒ Smooth Mesh ☐ Two Sided Mesh

Breaking Angle: 30.0

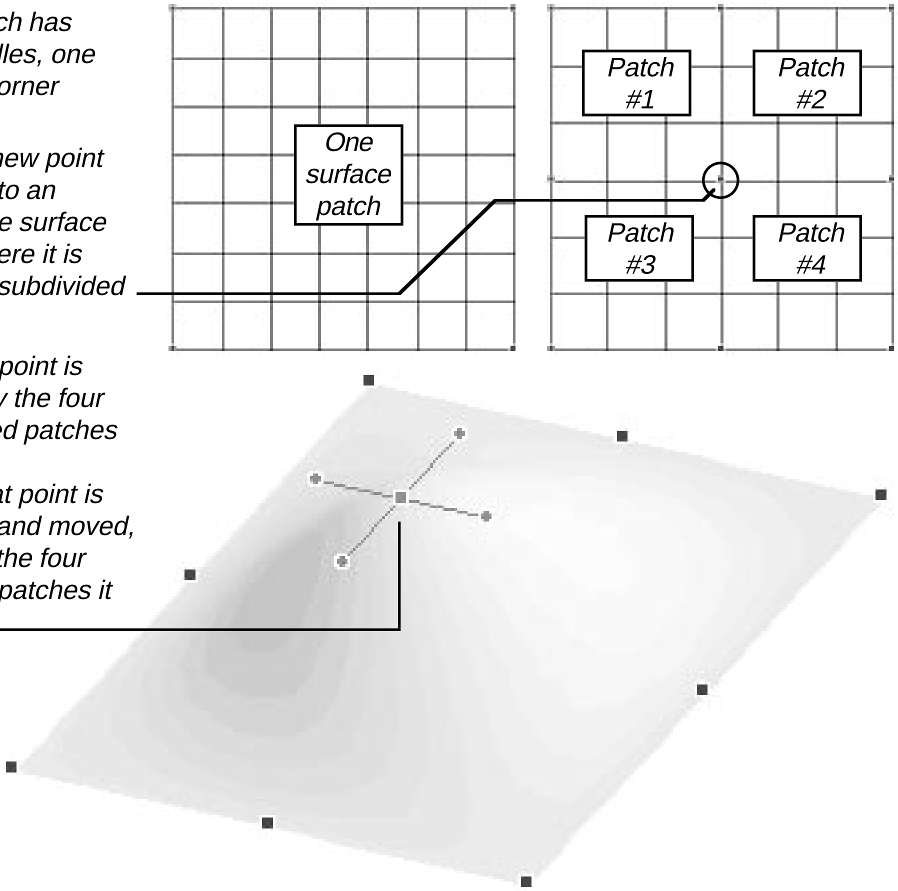
The dialog box that opens shows the current status of the polygons — single- or double-side — and allows you to click the appropriate radio button. Single-sided polygons provide for optical properties when a transparent texture is applied to the object. This is discussed further in *Chapter 5: Applying Textures*.

While in this dialog, you can also make the surface smooth or faceted. If smooth, you can enter the breaking angle. These fields are the same you will see when you use the Smooth modeling extension. For a complete description of these fields, please refer to the Smooth modeler later in this chapter.

Modifying bézier objects

To understand how to work with bézier objects, you must first understand how surface patches are constructed and manipulated. This is best accomplished by starting simple and observing how the surface behaves as you manipulate the vertices.

The following illustration describes the basic structure. Think of this simple rectangle, with its single surface patch, as a sub-set of more complex objects with multiple surface patches; the behavior is the same.



Each patch has four handles, one at each corner

When a new point is added to an object, the surface patch where it is added is subdivided

The new point is shared by the four subdivided patches

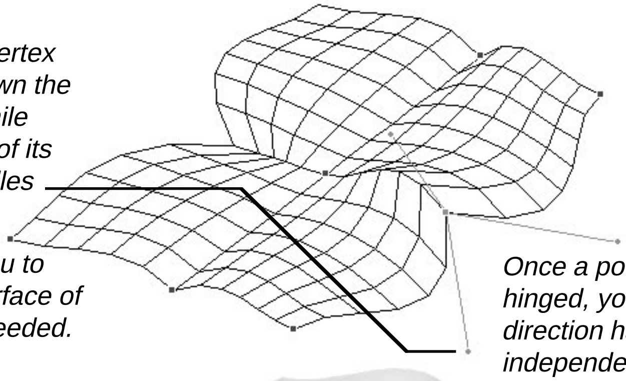
When that point is selected and moved, it affects the four adjacent patches it controls

As long as the vertex point you are moving is shared by adjacent patches, the surface responds in a smooth, predicable way. Most vertices are like that.

Making direction handles hinged

To hinge a vertex point, so that its direction handles can be moved independently, hold down the Option key while dragging the direction handle you wish to hinge.

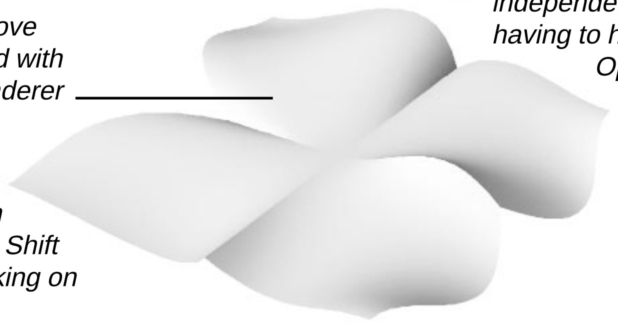
To align the direction handles of a vertex point that has been previously hinged (or was set that way when the object was brought into 3d Sculpter), hold down both the Shift and Option keys at the same time while clicking one of the hinged direction handles.



To “hinge” a vertex point, hold down the Option key while dragging one of its direction handles

This allows you to crease the surface of an object, if needed.

Once a point has been hinged, you can drag its direction handles independently without having to hold down the Option key each time.



Example of above object rendered with the Phong Renderer

To align the handles again, hold down both the Option and Shift keys while clicking on the handle.

Hinged vertices allow you to crease the surface. Aligning them makes the surface smooth at that point.

Note that if the vertex point has four direction handles attached to it, only one set of handles are hinged when you use the Option key. The other handles that pass through the point are not affected unless you also drag one of those handles with the Option key. When a set of direction handles has been hinged once, they can be dragged in future adjustments without holding down the Option key again.

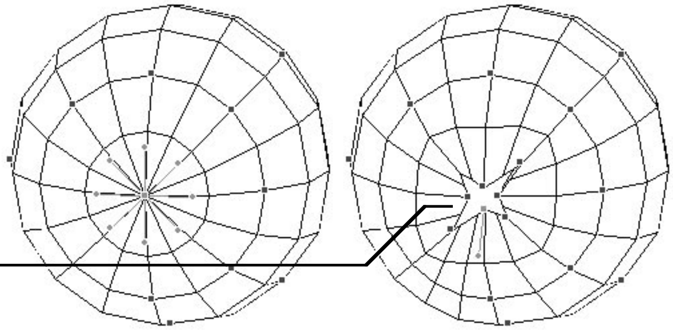
Working with co-incident vertices

The difficulty usually comes when you want to manipulate the surface using vertices that are not shared by adjacent patches. These are usually points that are coincident with one or more other vertex points that control their individual patches. Co-incident vertices occur when the corners of surface patches have been brought together to close up the surface of an object.

If you simply click on the blue dot showing in the window, you will select only the topmost vertex point of those that are co-incident. Then, if you drag that point, the others are left behind and the surface of the object suddenly has a “tear” in it! An example of this is shown in the following illustration. This is how a sphere is described in bézier form.

The poles on a sphere are actually made from 8 co-incident vertices

If the individual vertex points are moved outward, the surface opens or tears

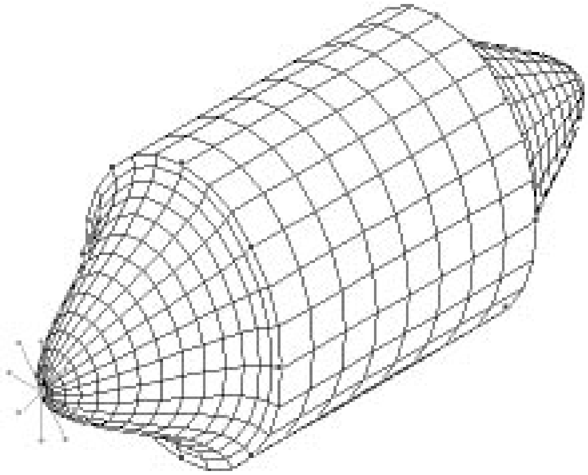


Note how each point has just one direction handle for the single surface patch it controls. This is the main clue that a vertex point is not shared by adjacent patches

Although this can be a useful technique in some projects, you usually do not want to open up the surface of an object. The better technique is to first drag a selection marquee around the blue dot representing the co-incident points. That selects all the points at once, then you can drag them as one and retain their co-incident relationship.

When working with co-incident vertices, it is best to select them all at once by dragging a marquee around the dot, then dragging the points as one.

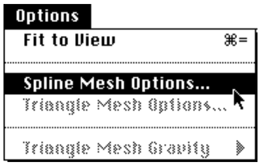
That way the surface behaves with predicable smoothness



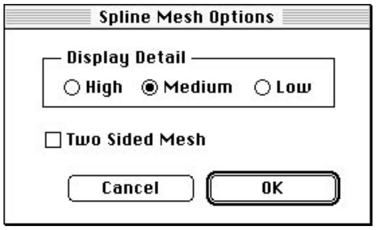
Once a single co-incident point is moved, it is hard to return it to its previous position with the other points, so you will want to be sure of this step if you choose to do it. Of course, the Undo command is available for inadvertent vertex moves, if used immediately afterward.

Spline mesh options

Sometimes, you may need to perform delicate adjustments on an object's surface. If you zoom in using the magnify tool, you may find that the surface representation is too coarse. The detail with which a bézier object is displayed on the screen can be adjusted through the Spline Mesh Options command in the Options menu.



This command opens a dialog box in which you can designate the bézier spline mesh detail. There are three choices: Low, Medium, and High. Just like the Triangulate Mesh Options dialog. Again, the significance of this feature is explained in *Chapter 5: Applying Textures.*


Spline Mesh Options

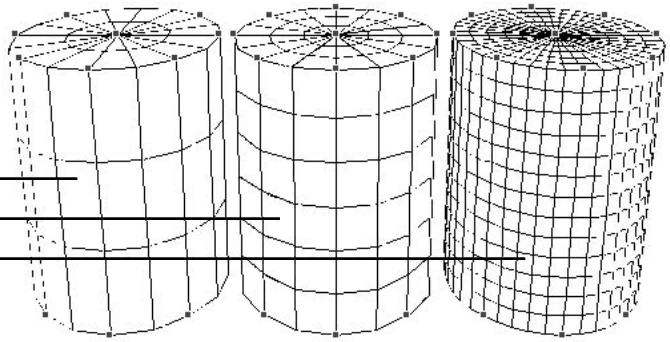
Display Detail

○ High ● Medium ○ Low

□ Two Sided Mesh

The display detail defaults to medium. This is the setting every time the 3d Sculpt window is opened for editing an object. Choosing either of the other settings and clicking on the OK button causes the window to re-draw at the desired detail level. This feature does not affect how the object is returned to the model when the session is ended, but it can help your work go easier while in the 3d Sculpt modeler.

Spline Mesh Detail Options



Examples of:

Low

Medium

High

The illustration on the previous page shows the difference between display detail levels. With cylinders, it is obvious. Other objects may not show a noticeable difference between low and medium.

Caveats

The 3d Sculpter modeling extension is unique among the various modelers included with Vision3d. It takes existing objects, often constructed with other modelers, and modifies their surface geometry. It does not appreciably add to the data required to describe an object, just changes it.

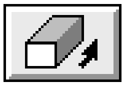
Therefore the usual warnings about being frugal with the number of polygons that may be generated do not directly apply in this case. However, there are some areas you may want to watch.

- Editing primitives in 3d Sculpter changes these objects from a true primitive to a bézier description of that object's geometry. Thus, a sphere in bézier form does take a bit more memory and does render a little slower. You should be aware of this.
- Adding a point to a bézier surface patch subdivides the surface patch. This takes some additional memory to store the object when returned to the model. While adding one point is not significant, it does add up. You should avoid indiscriminately adding more points than you need. The goal is always to describe any object as efficiently as possible, whatever its geometry.

Extrude

The Extrude model extension constructs 3-D objects using 2-D templates as the front and back faces, with a smoothed, triangulated mesh polygonal surface between them. It accepts any 2-D object, either bézier or polygonal as the template. Multiple objects can be extruded at one time, such the 2-D objects generated by the Text tool from font characters.

Multiple objects can be grouped or ungrouped. However, the resulting extrusions will be grouped as a single entity. While these may be ungrouped afterward, this is not recommended, since the modeler link will be broken by the ungrouping. If you wish to manipulate extruded objects separately, it is better to extrude them separately.



When this extension is present in the Vision3d Extensions folder, an Extrude command appears in the Modeling menu. The above button also appears on the Button bar.

Description

Extrusion is the process of “pushing” an object through a mold or template to define a surface along its longitudinal (depth) axis. In Vision3d, a 2-D polygon is used as the template. You can then specify how far you want to push the template through space to create the geometry for a 3-D object.

This version of the Extrude extension also allows you to specify a profile for the sides of an extruded object. The chosen profile is concatenated into the resulting extruded object to make a single polygonal entity. Using this feature, you can achieve simple straight extrusions, rounding or beveling of object edges, or radiusing to achieve complex geometry effects.

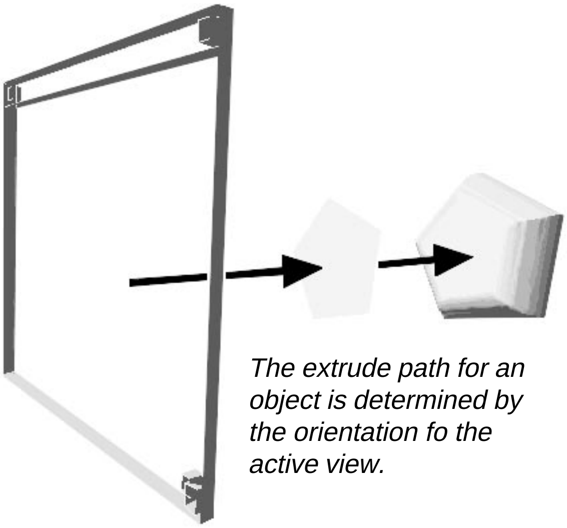
NOTE: *The Extrude extension smoothes the extruded surface as part of its operation. The breaking angle is fixed at 30°. If needed, you may re-calculate the smoothing with the Smooth modeling extension. The Smooth modeler allows you to set a higher or lower breaking angle.*

Several profiles are built into the Extrude extension. Additionally, you can create and import your own custom profiles. The making of custom profiles is discussed later in this topic.

How to use

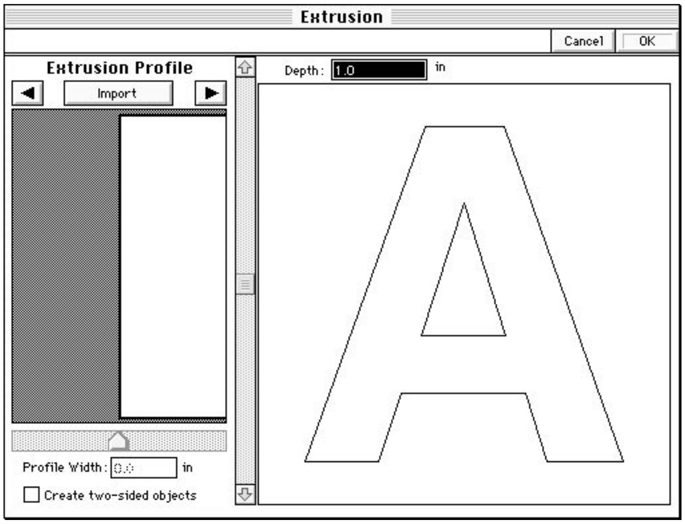
The Extrude modeler is an extension to the core Vision3d application. The command that calls an extrude operation is found in the last section of the Modeling menu under the Swept Surfaces sub-menu.

Extrusion occurs along a straight path in the direction of the active view’s relative Z axis. Because of this feature, it is possible to extrude an object at an angle other than perpendicular to the 2-D polygon.



The extrude path for an object is determined by the orientation fo the active view.

When the Extrude command is called, a separate dialog box opens. The left side of the dialog contains controls for graphically adjusting the extrude parameters. The right side shows the selected 2-D object to be used as the extrusion template.

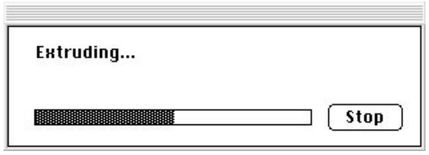


Depth: 1.0 in

Profile Width: 0.0 in

□ Create two-sided objects

Two numeric fields are present on this dialog for setting the extrusion depth and profile width. These are linked to the graphic controls for those same parameters; allowing you to make an adjustment using either method and obtain immediate feedback from the other control.



Extruding...

When you are satisfied with the parameters in the dialog box, click on the “OK” button to execute an extrude operation. A dialog appears to inform you of the progress of the extrude. Provision is made to terminate the process at any time. Clicking on the “Stop” button in the progress dialog returns you to the model, not the Extrude modeler.

Only 2-D objects can be extruded. If the Extrude command is not active, a selected object does not qualify for extrusion. If multiple objects are selected and only one of the objects cannot be extruded, the command is disabled.

More than one 2-D object may be selected and extruded together. A grouped set of 2-D objects can also be extruded. The resulting extrusion will be grouped as a single unit. They retain their separate identity within the grouping and can be ungrouped for access to the individual 3-D objects. However, ungrouping a set of objects that have been extruded together breaks that group’s connection to the Extrude modeler as its “native” modeler. In such situations, it is better to extrude each object separately then group them afterward. This maintains the modeler identity for the objects in case you need to re-extrude the objects.

You can re-extrude an object by either double-clicking on it or selecting it first, then calling the Re-extrude command from the Modeling menu. This can be done from any view without altering the original extrude direction.

Refractive textures on extruded objects

If you are planning to apply a texture with transparency and refraction on an extruded object, you will want to also give it correct optical properties.

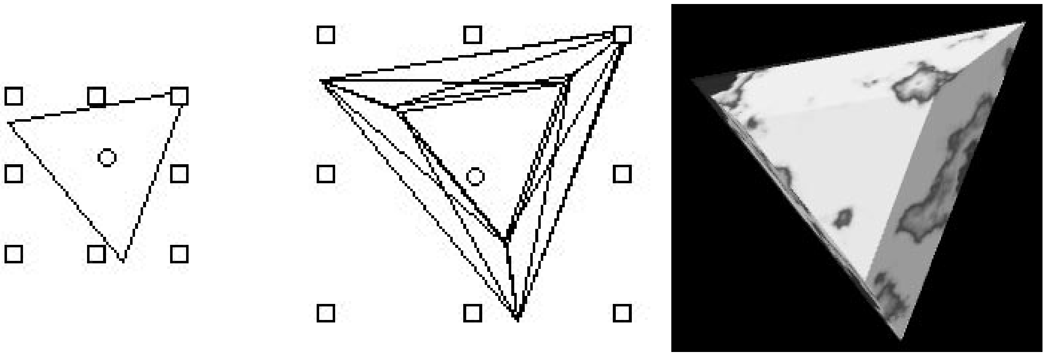
☐ Create two-sided objects

You have the option of making the polygons created with the Extrude modeler either single-sided or double-sided. This is set by checking a field in the lower right corner of the Extrude dialog box. Single-side polygons are only visible to the rendering algorithms from one side; double-sided are visible from either side. The effect of this difference is that objects made of single-side polygons appear solid when using a transparent texture. Objects with double-sided polygons appear hollow.

Thus, if there is an index of refraction in the texture definition, it will render differently depending on the types of polygons used. Of course, refraction only appears inimages rendered with Raytracing, RayPainting, or Raydiosity.

Using profiles

The square profile is the default profile. It is used for extrusions with straight sides. The width of this profile has no effect on the extrusion, since the extrusion produces only simple geometry. Thus, the width controls are disabled when this profile is chosen.



When other profiles are used for an extrude operation, the geometry and width of that profile becomes a component of the resulting extrusion. During an extrusion, profiles are added to the outside extents of the original 2-D template.

Making custom profiles

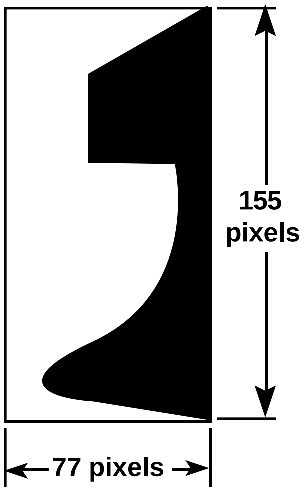
You can define profiles of almost any geometric complexity. The profile must be a 1-bit (B&W) PICT image of the dimensions indicated in the illustration on the next page.

It is recommended that you use a drawing program that saves a draw-PICT file for creating profiles. Properly constructed, it generates fewer vertices on the extruded object, and as the profile's width is adjusted in the Extrude dialog box, a draw-type profile maintains resolution better. A pixel-type PICT will also work, just not quite as well.

Make the profile document this size.

The profile geometry must be filled with black

The profile area must be continuous from the right edge of the document.



Note that the area inside the profile itself must be completely black all the way to the right edge of the document. THIS IS IMPORTANT. Otherwise the extrude operation will give unexpected results.

Caveats

- As will any modeling operation, the economical use of polygons is preferred. Profiles that generate excessive polygons should be avoided whenever possible. Those with curved surfaces are especially prone to this problem; often adding as many as 10,000 polygons to a model with one extruded object.
- If you expect to see the inside surface of an extruded object — because it has a transparent texture, the viewing position will be inside the object, or the extrusion is hollow — you must use double-sided polygons. Otherwise, the inside surface will not be visible to the renderer(s).
- You can re-extrude an object as long as it is identified internally as an extruded object. Ungrouping an extruded object permanently removes that identification. Grouping with other objects will not remove the connection, but the object is not separately available unless ungrouped again.
- If the 2-D object being extruded is a bézier object, it will be converted to a polygonal representation as part of the extrusion. By definition, the edges of bézier are very smooth, especially on curves. The smoothness of the conversion from bézier to polygonal depends on the apparent size of the object in the view. To make the profile edge of the extruded object smoother, simply make it appear bigger in the view

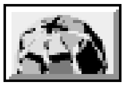
Since this issue comes up most often with text-shaped bézier objects, this feature is presented under the Text tool discussion earlier in this chapter. However, it applies to any bézier object being extruded.

Do not confuse apparent size with actual size. You do not need to scale up the object; all that is needed is to zoom in on the object(s) before extruding. This mechanism allows you to control the amount of polygons generated in the extrusion process. The larger the object appears in the view, the finer the conversion and the more polygons generated.

Obviously, if the extruded object(s) is the focus of the project you will want to invest extra polygons into their creation. If other objects have more visibility and the extruded object(s) has a background role, you will want to moderate the amount of polygons generated by making the 2-D bézier template smaller before extruding.

Facet

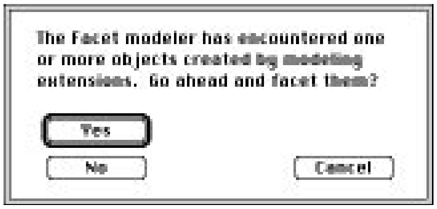
The Facet modeler is an extension provided to break objects down into individual polygons. This extension works with any type of object — primitive, polygonal mesh, or bézier spline mesh — and any number of objects at one time. The more objects faceted at one time, however, the more memory required to perform the triangulation and vertex point calculations.



When present in the Vision3d Extensions folder, a Facet command to use this extension appears in the Modeling menu, under the Resurfacing sub-menu. Additionally, the above button appears on the Button bar.

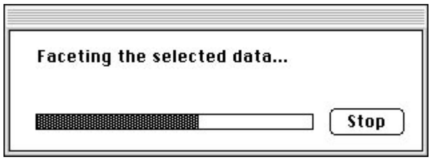
How to use

This modeling extension does not have any type of interface or settings that must be made in connection with its operation. To facet an object, simply select it and choose the Facet command (or click on its button showing on the Button bar). Depending on the type of object and its complexity, a faceting operation can be instantaneous or take several minutes.



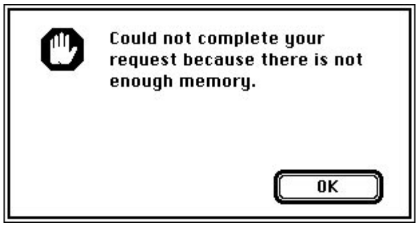
The Facet modeler has encountered one or more objects created by modeling extensions. Go ahead and facet them?

If the object chosen to be faceted was created with another modeler, an alert appears notifying you that a link exists and will be broken if you continue. You are given the option to cancel or continue the faceting operation. If you cancel, the object is untouched.



Faceting the selected data...

If you choose to continue, a progress bar appears to keep you informed of the time required. If there is not enough memory available to complete the operation, an alert appears to that effect and the object(s) is left untouched.



Could not complete your request because there is not enough memory.

Sometimes the out of memory message can appear even after the operation appears to have gone quite far along the progress bar. If more than one object is being faceted at a time, you may be successful by doing them one at a time. If there is still not enough memory, then you must either reduce the amount of memory being taken by your model or increase the RAM partition allocated to Vision3d.

The Undo command is available to revert to the original object, if used immediately after a faceting operation.

NOTE: *If the object being faceted has a single-sided surface, such as primitives do by default, the polygons that result are automatically changed to be double-sided. This is done so that you will be able to see the inner surfaces, if you remove part of the surface and look inside.*

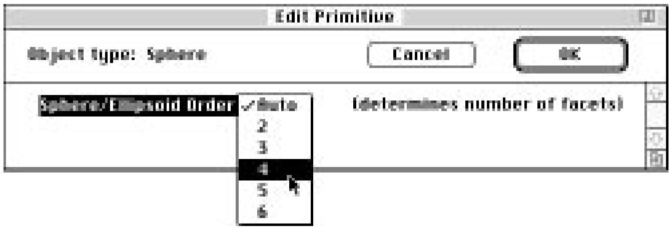
Controlling Faceting density

Some objects give you a measure of control over how finely the surface will be faceted. As mentioned at the beginning of this topic, the Facet extension sees objects as one of three basic types: Polygonal mesh, Bézier spline mesh, and Primitives.

If the object being faceted is a polygonal mesh, there is no control over how fine the faceting occurs. It will simply use the existing polygons that make up the mesh surface.

If the object is a bézier surface, the faceting varies with how large the object appears in the view. The larger its apparent size, the finer the faceting, and the more polygons generated. This behavior is identical to the bézier to polygon conversion process that occurs when a 2-D object is extruded, lathed or swept, only as applied to converting 3-D objects.

If the object is a primitive, the apparent size is also used for faceting density. However, there is also an override feature available through the Edit Primitive command in the Modeling menu. For primitive objects, a field is provided in that command's dialog where you can set the surface order detail of the objects and override the automatic setting. This command is fully discussed later in this chapter.

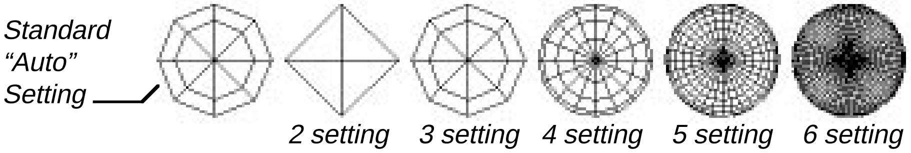


Object type: Sphere

Sphere/Ellipsoid Order (determines number of facets)

Normally, primitives are drawn in the model views using an auto-surface order mechanism for quicker re-draws of the screen, because the smaller the object appears in the view, the less surface detail generated. Of course, these primitives appear fully formed when rendered, unless you use this dialog to force a surface detail.

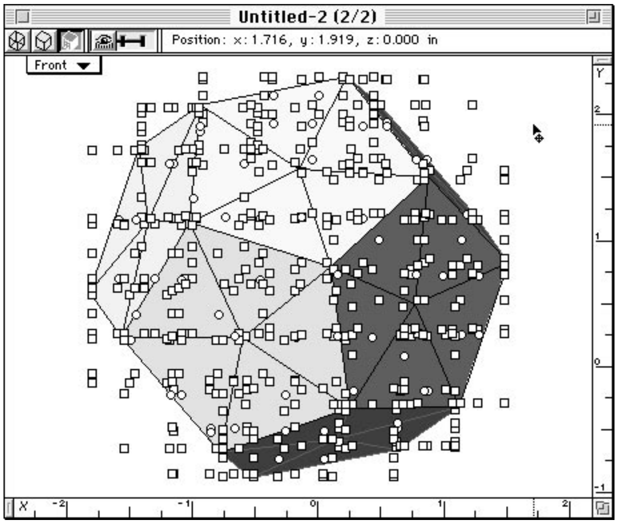
Variations in sphere with surface order settings in Edit Primitive Dialog



You can also use this option to control the faceting density of a primitive by first making the setting you want in the Edit Primitive dialog box, then using the Facet extension to turn the primitive into a bunch of polygons.

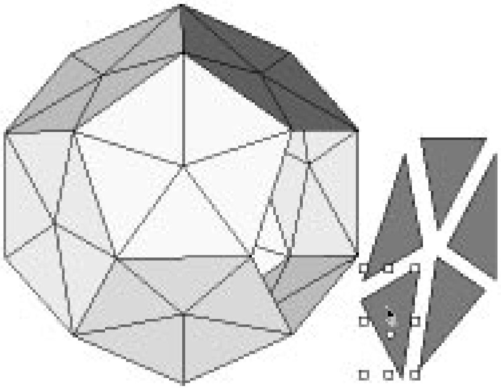
Handling faceted objects

When an object has been faceted, its surface becomes a collection of ungrouped polygons, each with their own vertices and individually selectable. The object is returned to the model with all the individual polygons selected.



Position: x: 1.716, y: 1.919, z: 0.000 in

You do not have to wait for all of the handles to draw on all the polygons; it is an interruptible event. You can move on to the next phase of modeling. Clicking anywhere away from the object unselects all the individual polygons and the object looks very much the same as it did before.



It is not the same, however. Because the polygons are now separate, there is no smoothing information connected with the surface. Even a sphere will render as a collection of flat polygons after faceting.

Why use faceting

Sometimes, your purpose for faceting an object is to dissect it or remove certain portions of the surface then work with the remainder. Another reason would be to allow the application of a different texture to each polygon or small groups of polygons. Yet another reason would be to re-smooth it with different parameters. This is especially useful with lathed, extruded, or swept objects. The options available for re-smoothing a faceted object are discussed in the Smooth modeling extension topic, next in this chapter.

There may be other uses. As with many other features in Vision3d, the possibilities are limited only by your imagination.

Grouping polygons

Vision3d supports multiple levels of grouping. For example, suppose you want to map a package design onto an object so that it looks like a standard “tuck box.” Vision3d does not have a mapping style that automatically wraps the package design as you wish, but this can be easily accomplished manually.

By faceting cube (drawn to make rectangular-shaped sides) into six separate objects, you can access the individual sides of the “box.” With properly prepared textures — containing the appropriate section of the package design as a color map — you can apply them to the various sides of the box.

Once the textures are applied, you can group the polygons for each box together. The group acts as a single object, just like it would before faceting, but now with the multiple textures as part of its surface description. You could then duplicate the box and turn it around to show the back side and group those two objects again so they stay together when you move them into the rest of the scene.

Smooth

The Smooth modeler is an extension provided to re-calculate the surfaces of polygonal objects after they are drawn or constructed in the model. This extension works with both 2-D or 3-D objects. While you can smooth multiple objects at one time, the resulting smoothed objects return as a single, un-groupable entity. If you want to have access to the individual objects after smoothing, you should perform this operation on them one at a time.

When present in the Vision3d Extensions folder, a Smooth command to use this extension appears under the Modeling menu, in the Resurfacing sub-menu. Additionally, the above button appears on the Button bar.

When to use

Use the Smooth extension whenever you want to recalculate the surface of a polygonal object so that it will render with a smooth surface.

Smoothing is the process of taking a group of selected and contiguous polygons, purging redundant co-incident vertices (if necessary), and replacing with a triangulated polygonal mesh; attaching normals to the vertex points in the process. Vertex normals are required for the renderers in Vision3d to produce a smooth surface.

The concept of smoothing a surface is presented as part of the general discussion on polygonal objects at the beginning of this chapter. You should read it to better understand what the smoothing process entails.

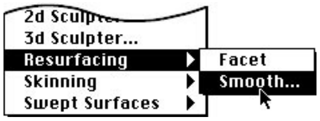
How to use

Some objects can be smoothed directly. Others must be faceted before this modeler can be used. Generally, all polygonal objects can be smoothed without having to go through the Facet modeler first.

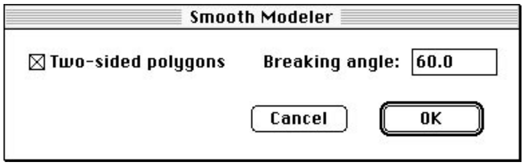
Obviously, primitives and bézier objects do not need to be smoothed, since they are already that way. However, there may be occasions when it will be necessary to use this extension on those object types. If you want to use the Smooth modeler on primitives and bézier objects, they must be faceted, i.e. — converted to polygonal form and broken down into the individual polygons that describe their geometry.

To smooth polygonal objects, use the following steps:

1. **Select the polygons to be smoothed.** Only selected polygons are smoothed. If an object has been faceted prior to using the Smooth extension, you must make sure all the polygons from the original object are selected that you wish to be included in this smoothing operation. Sometimes, it is most convenient to simply group the individual polygons immediately after they are faceted so you wont lose any of them. In other situations where you need to perform “surgery” on the object to remove or delete portions or it, you may still want to group the remaining polygons after the dissecting is complete.



2. **Choose the Smooth command.** This may be done either by clicking on the Smooth button on the Button bar or using the command in the Modeling menu directly.

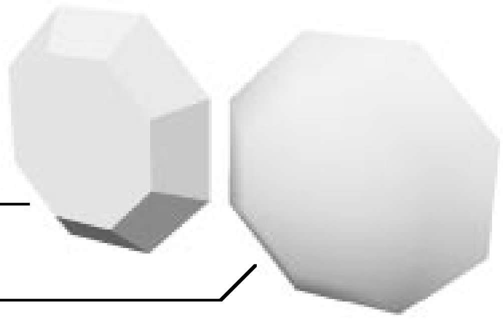


☒ Two-sided polygons Breaking angle: 60.0

3. **The Smooth dialog box appears.** This dialog allows you to set two parameters that affect how the smoothing operation is performed:

   A) *Two-sided Polygons.* This checkbox enables the generation of two-sided polygons. When disabled, one-sided polygons are created instead. Two-sided polygons can be seen from both sides when rendered. That implies that single-sided polygons are visible from only one side, which is true. While one-sided polygons do take a bit less memory, their significance is that they offer specific optical properties with transparent textures. Please refer to *Chapter 5: Applying Textures* for further explanation of this feature.

   B) *Breaking angle.* This field sets the maximum angle between adjacent polygons in a surface that will be smoothed. If the angle of incidence is greater, the surface will appear creased at their junction. The range of this field is from 0° to 180°. Generally the higher the value, the smoother an object appears. Lower values result in a more angular appearance of the surface.



The effects of changing the breaking angle on identical objects

Default 30° setting when extruded

Higher 89° setting from Smooth exten.

Changing the breaking angle affects the surface appearance ONLY, through manipulating the normals attached to that surface. This is not the same as the geometry. Depending on other factors in an object's construction, it can have a smooth surface, but still show a visibly-faceted profile or edge.

4. **Click on the OK button.** A progress bar appears to provide feedback during the smoothing process. The process can take a few seconds or several minutes, depending on the number and type of polygons being smoothed.

Upon completion, the smoothed polygons are returned to the model as a single entity or object. When viewed in wireframe, the smoothed surface is characterized by a distinctly triangulated appearance.

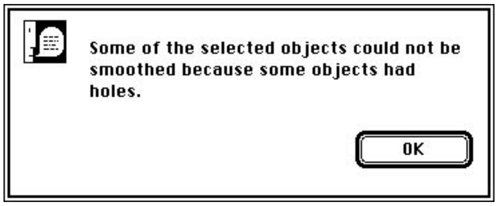
The lowest level renderer that will render objects with smooth surfaces is Gouraud. With Smooth Shading enabled in the Rendering Effects Dialog, polygonal objects that have been smoothed render with a smooth surface.

Limitations

The Smooth extension is a tool that can perform many useful functions. It cannot do everything, however, and sometimes it makes assumptions that don't agree with what you might want. There are two areas where this modeler may need special attention to create the desired result.

Handling un-smoothable polygons

If the Smooth modeler finds polygons among those that cannot be smoothed, an alert appears to inform you of the presence of these polygons. Typically, there are two reasons a polygon cannot be smoothed: 1) it has a hole in it or 2) it has more than one concave vertex point in its geometry. The alert will tell you which of these situations exist in the object you are attempting to smooth. If both conditions are present, the alert indicates the first one found.



Some of the selected objects could not be smoothed because some objects had holes.

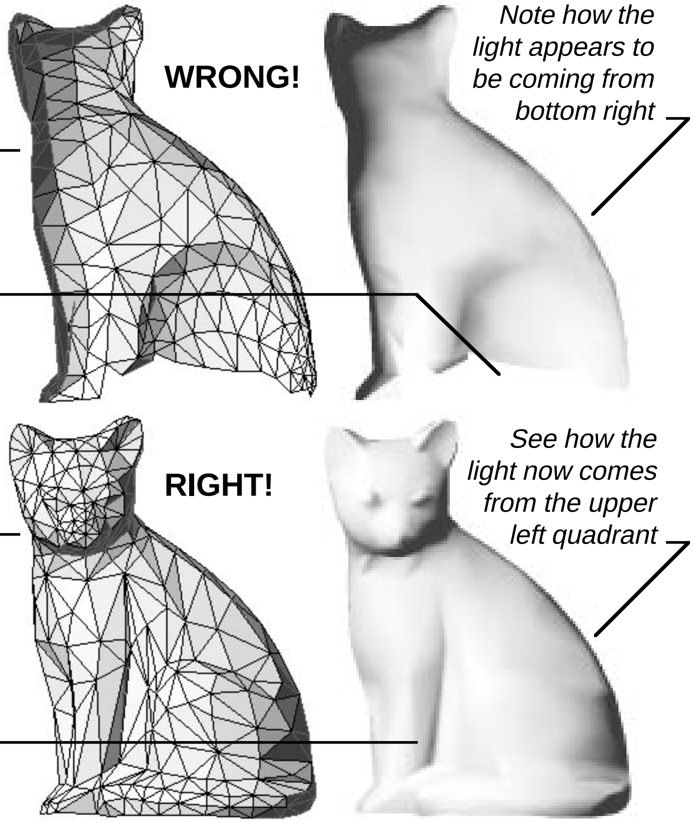
Un-smoothable polygons are not included in the mesh grouping that comes back. You are then free to work with the un-smoothable polygons further. These types of polygons can occur either as part of an object's construction or among those created with the Facet extension.

NOTE: *The 3d Sculpter modeling extension uses a different triangulation algorithm than the Smooth extension. You may find opening the object in 3d Sculpt will successfully triangulate the problem polygons where Smooth cannot. If needed, you can then use the Smooth modeler further to a different breaking angle (from the 30° default used by the other modelers).*

Flipping faces

When re-calculating a polygonal object's surface that is made of single-sided polygons, it is sometimes possible for the Smooth modeler to point some or all of the surface in the wrong direction.

Your first clue that this might be the case is that the lighting seems to be coming from the wrong direction. Additionally, in quick shaded display (and when rendering), some of the surface appears to be missing.



A cat, imported as a DXF file, smoothed with a 60° breaking angle and single-sided polygons

The object displays backwards and renders with parts missing

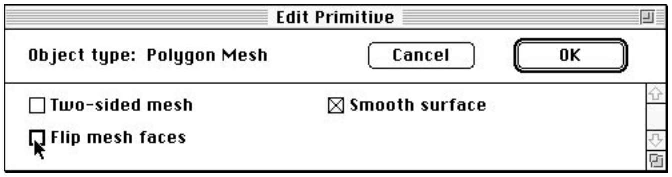
Note how the light appears to be coming from bottom right

Using Edit Primitive command, the faces on the polygonal surface were flipped

The object now displays correctly and renders with all of its surface visible

See how the light now comes from the upper left quadrant

This is most often seen on complex objects that come from other software, usually through the DXF import extension. This may also happen when converting bézier objects into smoothed polygonal objects.



Edit Primitive

Object type: Polygon Mesh

☐ Two-sided mesh

☒ Smooth surface

☐ Flip mesh faces

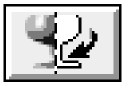
The solution is quite simple. All you need to do is select the object and choose the Edit Primitive command and click on the "Flip mesh faces" checkbox. Upon clicking the OK button, the faces of the single-sided polygons are flipped 180° to face EXACTLY the other direction.

Of course, you can also choose to make the surface double-sided in this same dialog. If you do not plan to apply a transparent texture to the object, this is an equally valid option, because then it won't matter which way the polygons face, since they are double-sided.

Lathe

The Lathe modeler is an extension provided to allow revolving 2-D objects through space to create 3-D objects. The resulting object is constructed of polygonal mesh, with smoothed surfaces.

Multiple objects can be selected and lathed at one time, if desired. however, only 2-D objects — polygonal or bézier — can be used as a lathe template. If the template is of bézier type, it is converted to a polygon as part of the lathe operation.

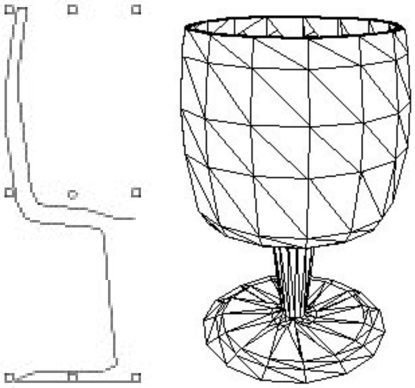


When present in the Vision3d Extensions folder, a Lathe command to use this extension appears under the Modeling menu, in the Swept Surfaces sub-menu. Additionally, the above button appears on the Button bar.

NOTE: *The Lathe extension smoothes the lathed surface as part of its operation. The breaking angle is fixed at 30°. If needed, you may re-calculate the smoothing with the Smooth modeling extension. The Smooth modeler allows you to set a higher or lower breaking angle.*

Description

Lathing creates symmetrical, 3-D objects by revolving 2-D templates or profiles around a designated axis. Lathing is a surfacing operation; creating the surface geometry as it revolves the 2-D profile through space.



Only 2-D objects can be lathed. If the Lathe command is not active, the selected object cannot be lathed. The object may be an open 2-D line or a closed object. If it is a closed object, its interior can be either empty or filled.

Lathing is a view-relative operation. That is, the active view determines the 2-D object's orientation in the Lathe dialog. The lathe direction is ALWAYS clockwise around the axis. For full 360° lathes, this may not be significant. But for partial lathes, it may be a factor in choosing the appropriate active view from which to lathe, so that the selected object will be constructed in the desired direction and orientation.

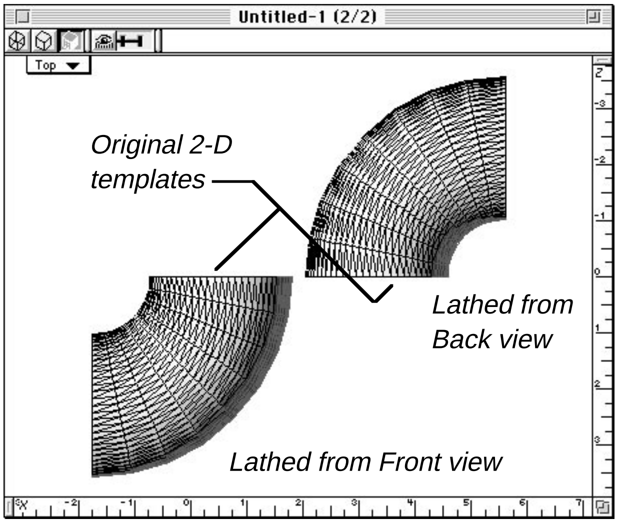
While multiple objects can be lathed at one time, there are several considerations when choosing to do that:

1. **There is one lathe axis.** Multiple objects will all be lathed or revolved around a common axis.
2. **The result is returned as a single grouped entity.** Attempting to ungroup the entity to gain access to individual items breaks their link with the Lathe modeler.
3. **Ungrouping will not break down simultaneously lathed objects as expected.** The original 2-D template(s) will be in one sub group, along with their mirrored "back faces," and each revolved surface will be in another. In a complicated project, this can cause some

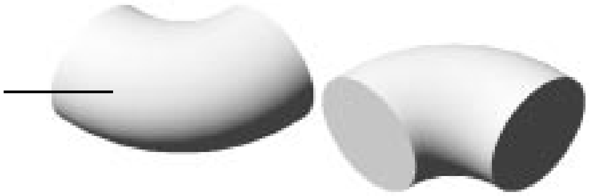
handling difficulties, especially if you want to apply different textures to the objects, faces included. You will have to ungroup several levels, then select all the elements for each object and group them back together.

The active view determines lathe direction:

Two identical circles were lathed 90° from the Front and Back views. The lathe axis was set to the left side of the objects.



Rendered images, from Isometric view



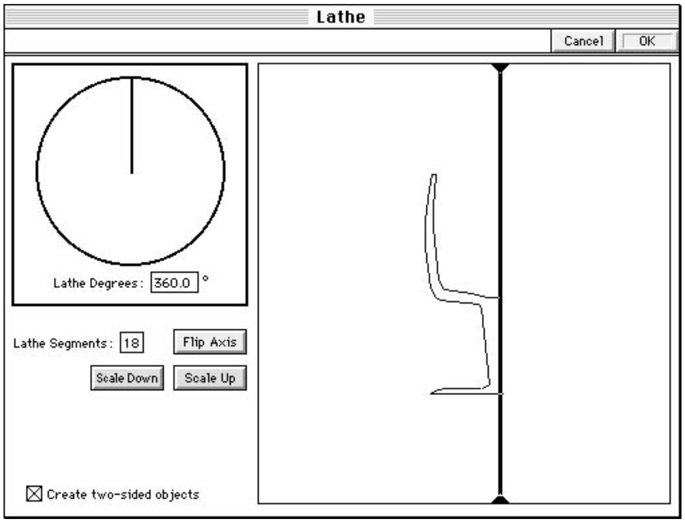
Given these issues, it is generally recommended that you lathe each object separately.

How to use

To call the Lathe extension:

- **Draw or import a 2-D template in any view.**
- **Select the 2-D template (or multiple 2-D objects, if required)**
- **Choose the Lathe command from the Modeling menu.**

The Lathe dialog window appears in which you set the lathe parameters and execute the procedure.



Lathe Degrees : 360.0 °

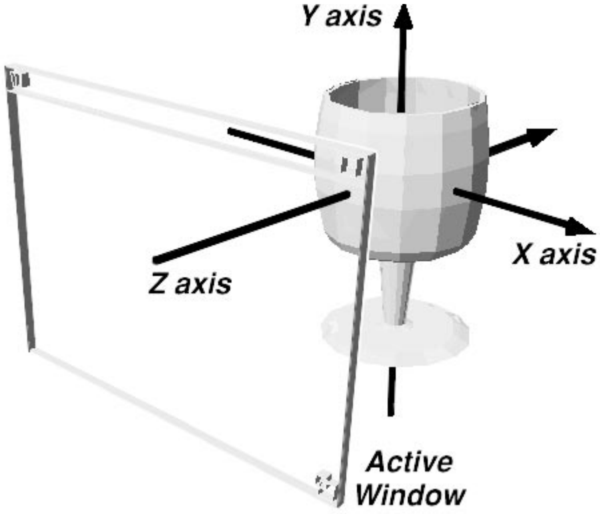
Lathe Segments : 18

☒ Create two-sided objects

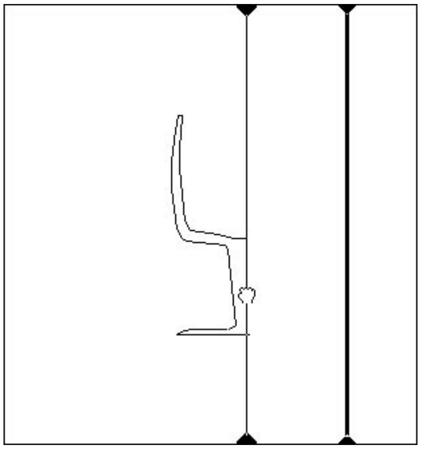
The Lathe dialog window contains all the required controls for setting the lathe parameters, i.e. — lathe axis, degrees of rotation, angle of tilt (or skew) for the lathe axis, and the number of segments. The left side of the dialog presents the lathe parameters. The right side displays the object to be lathed, along with a control to position the lathe axis.

Lathe axis

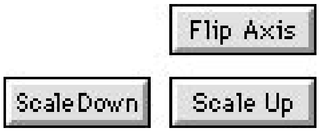
The lathe operation's axes are initially determined by the orientation of the active view. It will be the same as the view's X, Y and Z axes at the time the extension is called.



Once in the Lathe dialog, you can further adjust the axis control on the X and/or Y axes. This is accomplished by clicking on the axis control and dragging it within the display area of the dialog.



A “Flip Axis” button is also provided to rotate the lathe axis 90° from its current setting. Each time the button is clicked, the axis rotates 90° counter-clockwise in the display area.

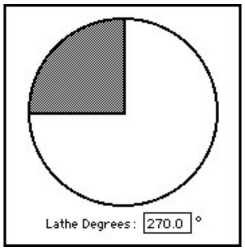


Occasionally, you may need to move the lathe axis farther from the object than the display area allows. To accommodate this requirement, the object may be scaled smaller in the display area; allowing a greater range of distance for the lathe axis control. Each click of the scale button reduces by 50% or enlarges by 200%. An object cannot be enlarged greater than its actual size, however.

Note that this scale change is relative to the lathe axis within the Lathe dialog only. It does not affect the actual scale of the completed lathed object when returned to the model.

Degrees of rotation

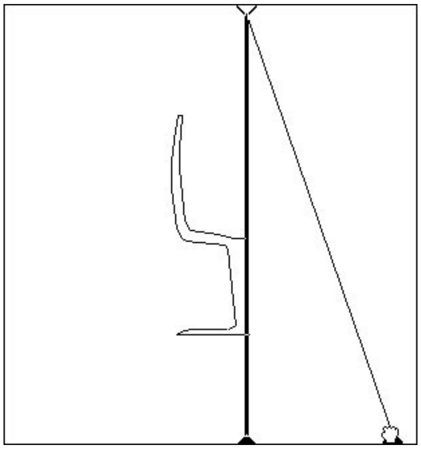
This control allows you to graphically define how much of a full circle (360°) is to be lathed. By moving the Rotation Circle control, the degree of rotation value is changed to the arc defined.



An editable numeric field is also provided to enter the lathe degrees directly, if desired.

Angle

The lathe axis control can be skewed from straight vertical or horizontal. In effect, you can tilt the rotation of the template. This can produce some very useful results.

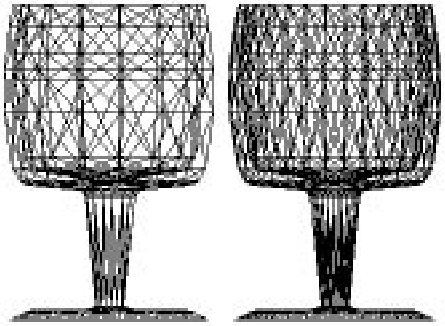


Number of segments

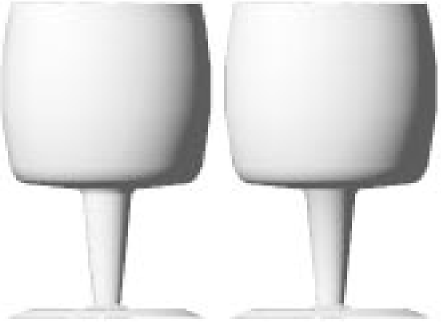
The number of segments may be specified. Note that the number of segments is generated within the degrees of rotation. For example, if 15 segments are specified for a 360° lathe, a segment will be generated every 24°. If the lathe amount is only 120°, segments are generated every 8°.

The minimum number of segments allowed is one per 90° of lathe. For best results, it is recommended that the minimum number of segments be set to generate a segment every 30°. This also corresponds to the breaking angle used by the Lathe modeler. If segments on a lathed object are greater than 30°, the surface will not appear smooth when rendered.

The goblet on the right was lathed with 30 segments, while the one on the left was lathed with only 15 segments…



…Note how the surfaces of the rendered goblets are equally smooth.



Because the lathe operation produces a triangulated mesh surface, increasing the number of the segments, i.e. — making the facets finer, is not necessary to create a smoother surface on rendered objects.

Refractive textures on extruded objects

If you are planning to apply a texture with transparency and refraction on a lathed object, you will want to also give it correct optical properties.

□ Create two-sided objects

You have the option of making the polygons created with the Lathe modeler either single-sided or double-sided. This is set by a checkbox in the lower right corner of the Lathe dialog box. Single-side polygons are only visible to the rendering algorithms from one side; double-sided are visible from either side. The effect is that objects made of single-sided polygons appear solid when using a transparent texture. Objects with double-sided polygons appear hollow.

Thus, if there is an index of refraction in the texture definition, it will render differently depending on the types of polygons used. Of course, refraction is only available in raytraced images, as discussed in *Chapter 8: Rendering*.

Re-lathe option

Vision3d identifies each object’s native modeler in the event that you wish to modify an object later. Through this link, you are able to simply double-click on a lathed object to bring up the Lathe dialog with that object’s current parameters. You can also re-lathe an object by selecting it and using the Modify/Re-lathe command from the Modeling menu.

The link can be broken by using the Ungroup command or by using another modeler, such as the Facet or Smooth extensions, to change the object in other ways. Using a different modeler then links the object to that extension as its native modeler.

Caveats

- Lathing with an excessive number of segments is not necessary for surface smoothness. You should use the lowest number of segments that can adequately describe the surface.
- If you expect to see the inside of a lathed object, either because it is not a closed object or the viewing position is inside the object, you must use double-sided polygons. Otherwise, the inner surface will not be visible to the renderer(s).

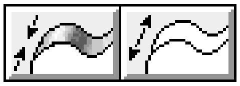
Skin/Unskin

The Skin and Unskin modelers are two separate extensions, but discussed together, since they are so closely related. The Skin modeler is provided to loft a surface over two or more “ribs.” The resulting object is called a FlexObject™. The Unskin modeler is provided to remove the skin of a FlexObject so you can gain access to the ribs again for further editing. The Unskin command works only on previously-skinned FlexObjects.

The Skin modeler works with any type of object as ribs — 2-D primitive, 2-D polygonal, or 2-D bézier — and any number of objects at one time, as long as they are 2-D objects. However, the ribs are automatically converted to bézier upon skinning, since the surface created is bézier. Sometimes, the conversion can cause a change of rib geometry. It is strongly recommended that you use bézier tools in the construction of the ribs, or convert any polygonal ribs to bézier with the 2d Sculpter modeler before skinning. That way, you can be guaranteed of geometry integrity during the skinning process.

NOTE: *The Skin modeler must work with actual 2-D objects. Instances of shapes cannot be skinned. If you do find an instance has been used for an intended rib, you can simply ungroup it to make it a regular 2-D object and break its link to the shape. The handling of shapes is discussed in* Chapter 4: Using Shapes*.*

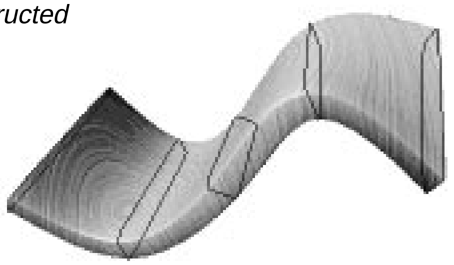
The number of ribs required depends on the extent of the object and the geometry you wish to construct. As usual, the more ribs used and the more complex the surface, the more data generated in the skinning process and the more memory required to store and render the object.



When present in the Vision3d Extensions folder, Skin and Unskin commands appear on the Modeling menu, in the Skinning sub-menu. Additionally, the above buttons appear on the Button bar.

Description

The Skin modeler is a powerful extension that adds a versatile type of object to Vision3d's modeling repertoire.



FlexObjects™ are constructed by skinning 2-D ribs positioned as needed to achieve the desired contours.

In this illustration, ribs are superimposed to show how they easily create complex, organic, and asymmetrical geometry.

FlexObjects are created using a three-step process: 1) position the 2-D "ribs" as needed to define an object's contours and cross section, 2) link the ribs together in the desired sequence (using the Link tool on the Tool palette), then 3) construct a bézier surface over those ribs with the Skin modeler.

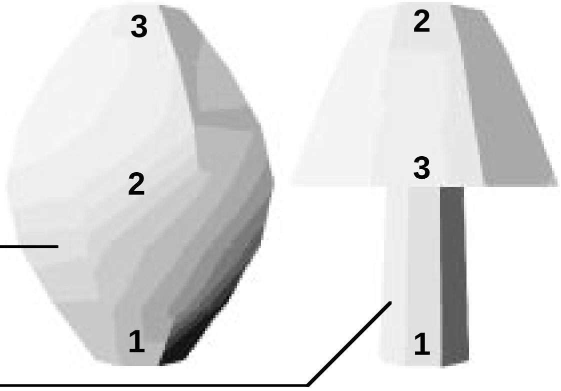
Ribs are simply 2-D objects of any desired geometry. The ribs may be of different form and may be scaled larger or smaller to describe a desired contour. They do not have to contain the same number of edges or vertices. One rib can be a rectangle and another rib a circle…Vision3d's Skin modeler will automatically resolve the differences in the number of vertices or sides to the ribs. For an in-depth discussion of vertices, polygons, and bézier splines, please refer to the beginning of this chapter.

There is no limit to the number of ribs that can be used to define an object's geometry, depending on the available memory. However, the sequence in which the ribs are linked also determines the final 3-D geometry or shape.

Example of how a FlexObject's geometry may be changed with the linking sequence.

Object #1 was linked in a normal sequence …bottom, center, top

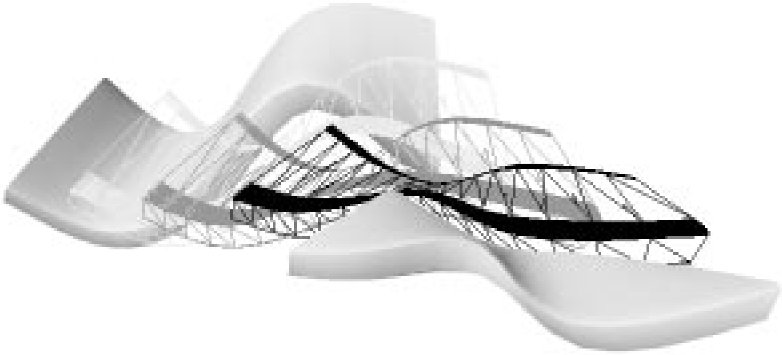
Object #2 was linked in a "folded" sequence …bottom, top, center



Additionally, FlexObjects are animate-able in ways that can take advantage of their unique construction. If any or all of the ribs in a FlexObject are given time-dependent actions prior to skinning — i.e., movement, scale changes, rotation — the resulting FlexObject's surface follows the prescribed animation sequence.

Illustration of animated FlexObject

By giving the ribs motion BEFORE skinning, this object can be made to "flex" or change shape during an animation.

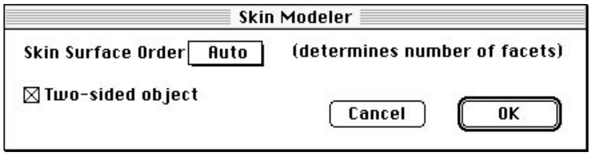


Each rib can be given independent motion; hence, the name, "FlexObject." Please refer to *Chapter 7: Scripting Animations* for more information on defining animation actions, entering event markers on the Animation palette and generally setting up animation sequences for objects.

Using the Skin Modeler

As indicated, the sequence in which the ribs of a FlexObject are linked determines the skinning sequence and resulting geometry. Additionally, before performing the actual skinning process, you must select all the ribs to be included in the constructed object. If a rib has been linked previously, but is subsequently not selected for skinning, it will be ignored at construction time. In such a case, you must unskin the FlexObject, re-select all the ribs, and skin it again to fix the skinning error.

Upon initiation of the skinning operation, a dialog box appears in which you set the parameters for skinning.



Skin Modeler

Skin Surface Order: Auto (determines number of facets)

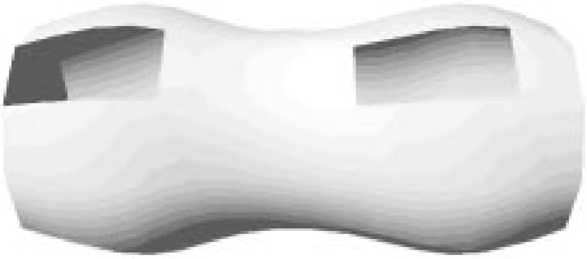
☒ Two-sided object

There is two settings to make in this dialog box. The Two-sided surface checkbox allows you to designate whether the object will have a single- or double-sided bézier surface. Single-sided objects are only visible to the rendering algorithms from one side; double-sided objects are visible from either side.

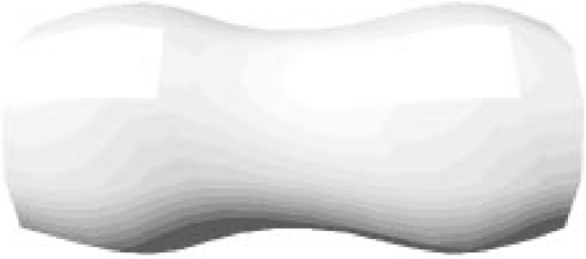
The Skin Surface Order field allows you to control how finely the surface is represented by polygons in the modeling window and when rendered. Skin objects are bézier in construction, and as such will always render with a smooth surface. But there may be a situation when you want a more faceted appearance to the surface. This lower the value in this field, the coarser the faceting level. Normally, this is left on the “automatic” setting so its representation can be as fine as required, either in the modeling windows or rendered images.

Transparent and refractive objects are usually made of single-sided surfaces so they will appear solid and refract light coming through them. Objects made of double-sided surfaces appear as a hollow or empty shell when rendered with a transparent texture applied, but they cannot appear refractive (even if the texture specifies it). Refraction is a rendering option available with the Raytracing and higher algorithms.

FlexObjects with surface openings that allow you to see the inside can ONLY be constructed with two-sided surfaces.



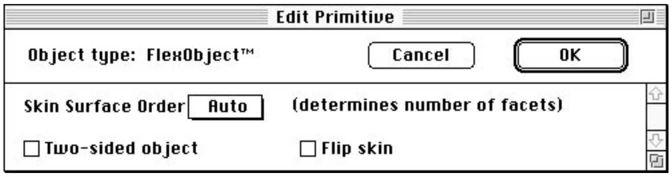
…Otherwise, the single-sided object disappears when viewed or rendered from its back side or inside.



NOTE: *If either end cap of a FlexObject is hollow or unfilled, or if any of its ribs are not closed — i.e., the beginning and ending points of the object do not meet, the Skin modeler FORCES the object to be built with a two-sided surface…there is no option. Open ends or open sides allow you to see inside the FlexObject. Thus, open objects must always be two-sided for their insides to be visible when rendered.*

Adjusting a FlexObject surface after skinning

You can adjust an existing FlexObject's skinning parameters without having to unskin and re-skin the object again. This is accomplished by choosing the Edit Primitive command while the object is selected. A dialog box appears that is similar to the Skin Modeler's original dialog box when the object was constructed.



Edit Primitive

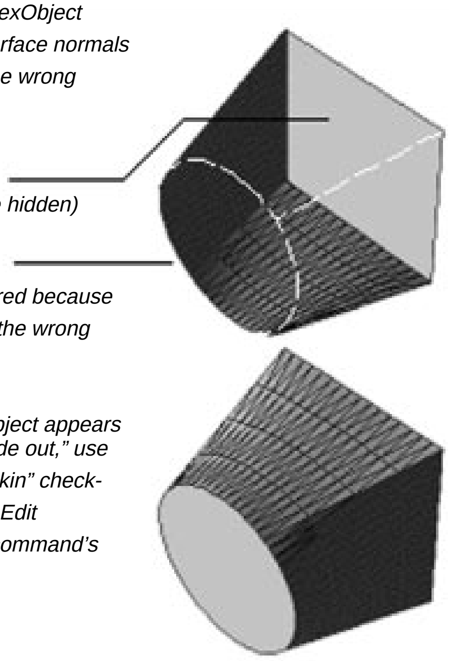
Object type: FlexObject™

Skin Surface Order Auto (determines number of facets)

□ Two-sided object □ Flip skin

One of the options available in the Edit Primitive dialog that is not in the Skin modeler's dialog is to flip the faces of the skin object. This is provided for those rare times when the Skin modeler may create the surface of a FlexObject with its normals pointing the wrong direction.

The most obvious indicator that a FlexObject's surface is facing the wrong direction is that the lighting appears to be coming from the wrong direction. Its geometry also appears inconsistent, as shown in the following illustration. Of course, this would only be apparent if the surface is single-sided. With a double-sided surface, the normals face both directions and will always appear correct.



The top FlexObject has the surface normals pointing the wrong direction

Back face (should be hidden)

Front face (disappeared because it's facing the wrong direction)

If a skin object appears to be "Inside out," use the "Flip Skin" check-box in the Edit Primitive command's dialog box

Changing the fields in the Edit Primitive dialog box, then clicking on the OK button causes the FlexObject's surface to be recalculated. Depending on the direction of the surface normals, its appearance could change radically.

Using the Unskin Modeler

While the Edit Primitive command allows you to perform minor adjustments on the surface of a FlexObject, there may be occasions when you wish to alter the object in more fundamental ways. Reverting to the original ribs from a constructed FlexObject is a straight forward process. Simply select the FlexObject in question, then choose the Unskin command from the Modeling menu. Alternately, you can click on the Unskin button found on the Button bar to initiate the procedure.

The ribs are restored in their exact state at the time the FlexObject was constructed. This also includes any animation event markers defined for the ribs. You may further edit the exposed ribs at this point — for geometry, position, scale, rotation, etc. — then, select them again and initiate a skinning operation of the "new" FlexObject.

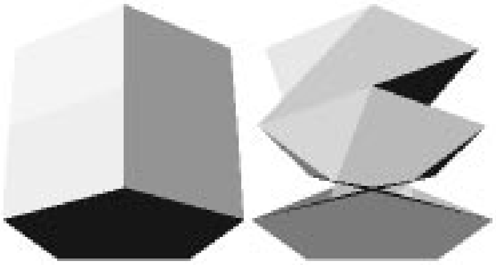
Note: *Editing a rib's form in any of the other modelers (such as 2d Sculpter) causes established linking to that rib to be severed. In that case, you must re-establish the linking between ribs before skinning.*

Caveats

- Avoid over-modeling. As with any object, the memory needed and the rendering time is directly related to the complexity of the surface created to describe a FlexObject. The number of ribs used to define an object also determines how many surface patches are created for the FlexObject. In general, it is preferable to use as few ribs as required to describe the basic contours of the object and rely on the object's natural smoothing ability to achieve the surface appearance desired.
- If the surface of a FlexObject unexpectedly deforms, it is probably because the ribs have become mis-aligned. Mis-aligned ribs cause the skinned surface to appear "twisted." Do not confuse this type of twisting with the natural flexing of an object. This problem causes the surface to break up and its shading to become inconsistent.

  Fortunately, the beginning vertex point on existing ribs can be changed and the problem easily fixed through the "2d Sculpter" modeling extension. Changing the location of the beginning vertex points is discussed in the documentation for that extension.

Mis-aligned ribs can cause the Skin modeler to twist the surface during construction. This is fixed by aligning the beginning vertex point on each rib.



All polygons have a beginning and ending vertex point. Usually, you do not care where these points are on a polygon. But with FlexObjects, it can become important, as shown in the above illustration. The two objects here were made from the same ribs, except that the one on the right was purposely mis-aligned prior to skinning.

Twisting can happen for a couple of reasons:

1) If a rib is rotated too far in relation to those next to it, the resulting surface "twists" because the corresponding vertices are too far away and the angles are too extreme to shade correctly.

2) It is also possible to assemble a set of ribs for a FlexObject that are mis-aligned during creation through the use of different tools — i.e., using the rectangle tool for some ribs, then making other ribs with the regular polygon tool where the beginning vertex point may be positioned in a different location on the rib.

- Whenever possible, it is preferable to use ribs with the same number of vertex points. Each rib may have a different geometry, but the number of vertex points can often be the same. If so, the result is a more efficient construction of the FlexObject.

  While the Skin modeler does not require the number of vertices to be the same on each rib, when they are it performs its calculations much faster and the resulting FlexObject takes less memory. When the vertex points on all ribs are exactly the same number, the Skin modeler can use the standard smoothing construction that produces a simpler surface. If differences are detected, another type of algorithm must be used, which takes longer to construct and requires more memory to support.
- If you are planning to apply a texture with transparency and refraction to a FlexObject, you will want to also give it correct optical properties. That means making the object with a single-sided surface so it appears solid and can properly refract light rays reflected from objects behind it.

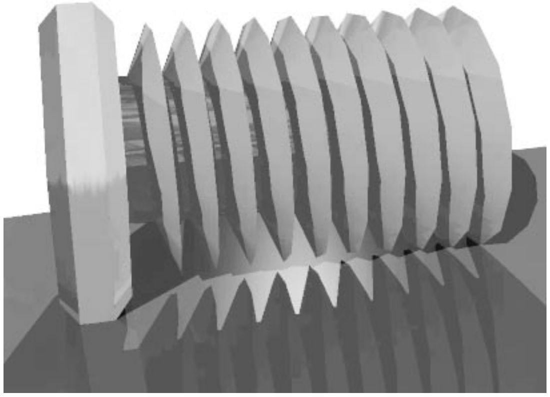
Sweep

The Sweep modeling extension performs a combination of operations on a selected object to produce complex geometric shapes. This extension works with only one 2-D object or 2-D Group at a time.

When present in the Vision3d Extensions folder, a Sweep command to use this extension appears on the Modeling menu, in the Swept Surfaces sub-menu. Additionally, the above button appears on the Button bar.

Description

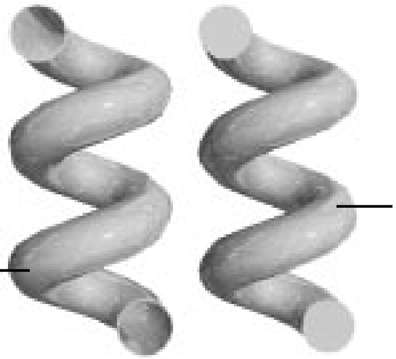
Through a combination of both lathe and extrude transformations, this modeler uses a 2-D object as a template and revolves it around a dynamic axis, i.e. — the axis can be moved between segments or steps. This is done through parameters set in the Sweep dialog box.



Sweep objects are constructed as polygonal mesh . Only 2-D objects can be swept, but they may be either polygonal or bézier in nature. If the Sweep command is dimmed when the Modeling menu is pulled down, or the Sweep button is grayed out, the selected object cannot be swept.

Additionally, the object may be an open or a closed object. If closed, it can be either a hollow or filled object. Hollow templates sweep as hollow 3-D objects, without solid end caps; filled polygons have end caps which give the appearance of a solid object.

Example of sweeping frame and filled 2-D circles



Frame (empty) circle produces a hollow 3-D Sweep object

Filled (surface) circle produces a solid 3-D Sweep object

Multiple objects may be selected and swept as one entity. They are swept around a common axis. The result is returned to the model as a group of 3-D objects. The group may be ungrouped to access the individual objects, if needed. However, this will permanently remove their connection to the Sweep modeler for possible re-sweeping in the future. You may also find they do not ungroup as expected, and can be difficult putting together into coherent objects. Generally, it is recommended that objects should be swept individually then assembled together.

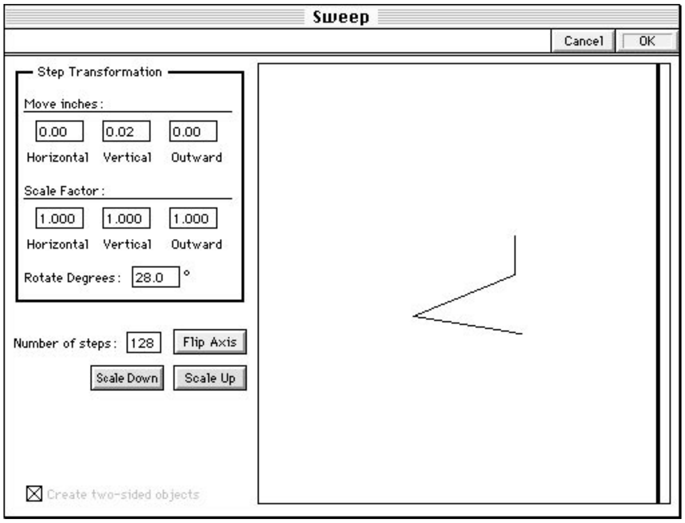
IMPORTANT: *The sweep direction is ALWAYS clockwise around the axis. This may be a factor in choosing the active view from which to initiate a Sweep operation.*

How to use

To call the Sweep modeler:

- **Draw or import a 2-D template in any view.** Make a point of knowing the actual size of the object. This may be important as you set the amount of axis offset and scale changes between steps.
- **Select the 2-D object.**
- Choose the Sweep command. This can be done from the Swept Surfaces sub-menu in the Modeling menu or by clicking on the Sweep button on the Button bar.

The Sweep dialog box appears. The left side of the dialog presents the sweep parameters. The right side displays the object to be swept, along with a control to position the sweep axis.



Step Transformation

Move inches:

Horizontal	Vertical	Outward
0.00	0.02	0.00

Scale Factor:

Horizontal	Vertical	Outward
1.000	1.000	1.000

Rotate Degrees: 28.0 °

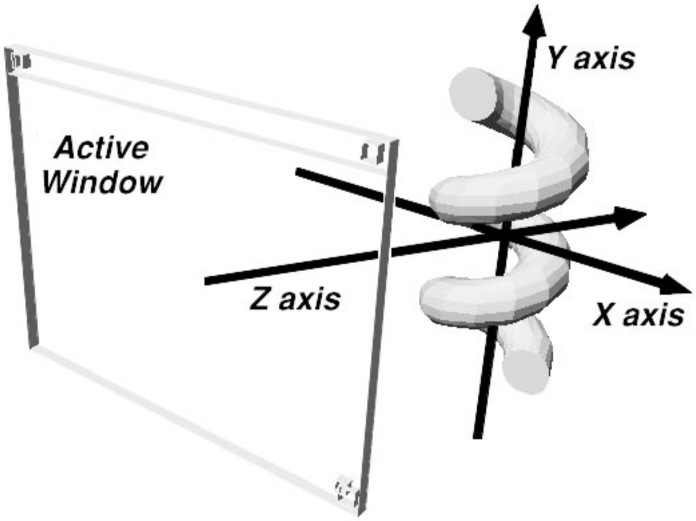
Number of steps: 128

☒ Create two-sided objects

Sweeping an objects requires that you first set how a specific segment or step will be constructed, then set how many steps or repetitions the sweep process generates. The degrees of rotation field applies only to individual steps, not the total amount rotation for the finished object. THIS IS AN IMPORTANT DISTINCTION!

Sweep axis

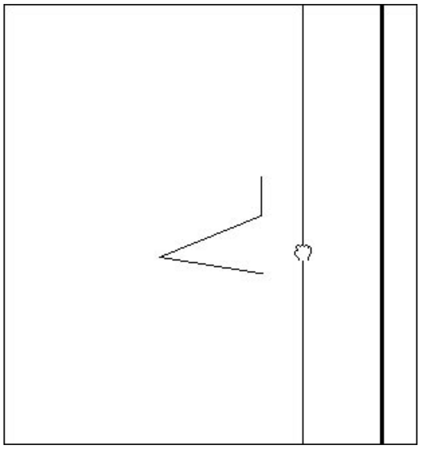
The Sweep operation's axes are initially set the same as those of the active view at the time the extension is called.



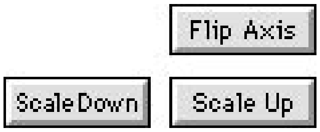
Once in the Sweep dialog, you can further adjust the axis control on the X and/or Y axes, but not the Z axis. This is accomplished in two ways:

- **By clicking on the axis control and dragging it within the display area of the dialog.**
- **By clicking on the "Flip Axis" button.**

The axis control can be dragged side-to-side or top-to bottom by clicking and dragging anywhere along its length. This is the means of positioning the axis relative to the template being swept.



A "Flip Axis" button is provided to rotate the sweep axis 90° from its current setting. Each time the button is clicked, the axis rotates 90° in the display area.

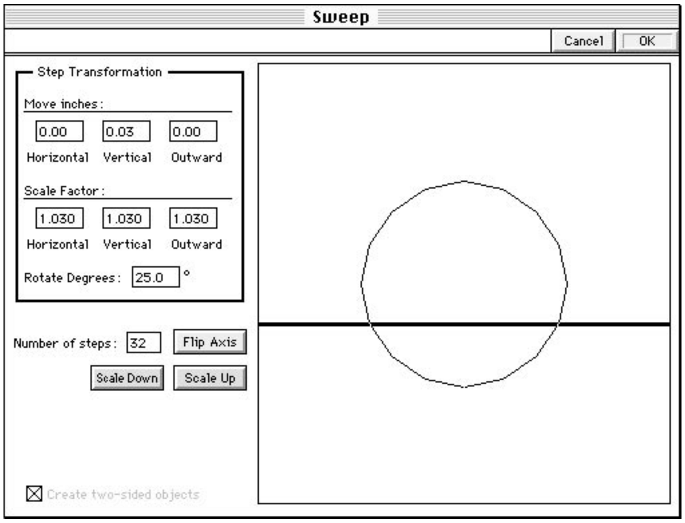


Occasionally, you may also need to move the sweep axis farther from the object than the dialog's display area allows. To accommodate this requirement, the object may be scaled smaller in the display area; allowing a greater range of distance for the sweep axis control. Each click on one of the scale buttons reduces by 50% or enlarges by 200%. An object cannot be enlarged greater than its actual size, however.

Note that this scale change does not affect the actual scale of the completed sweep object when returned to the model.

Step Transformations

Each step in the sweep object is specified in this portion of the Sweep dialog box. You have the option to dynamically shift the axis and alter the size for each segment, as indicated in these fields.



Step Transformation

Move inches:

0.00	0.03	0.00
Horizontal	Vertical	Outward

Scale Factor:

1.030	1.030	1.030
Horizontal	Vertical	Outward

Rotate Degrees: 25.0 °

Number of steps: 32

☒ Create two-sided objects

The units of measure for this dialog are the same as those specified for the active model through the Set rulers command.

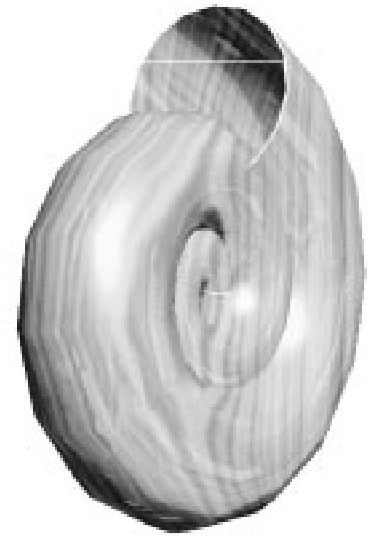
- **For the Move fields:** a zero value in any of these three fields means the sweep axis remains stationary in that direction. Entering a value causes rotations to be compounded based on the position of the sweep axis during the previous rotation instead of the starting point.

  Thus, if a value of 0.03 is entered in the Vertical Move field, the sweep axis is shifted .03 units upward (positive numbers move up, negative values move down) AFTER the first step, then another .03 units upward following the second step, then another .03 units after the third step, and so on until the specified number of steps is completed.

- **For the Size fields:** a value of one (meaning 100% of previous step) in a field keeps the size of the segment the same size in that direction throughout the sweep operation. A value greater than one (1.000) progressively enlarges the object for each step; a value less than one reduces the object for each step.

  Thus, if a value of 1.030 is entered in all the Size fields, the size of the original circle is enlarged 103% AFTER the first step, then another 103% following the second step, then another 103% after the third step, etc.

The result of the values in the above illustration will be an outward spiral from the beginning sweep axis, as in the nautilus shell shown in the illustration shown on the next page.



This sweep object started as a 2-D circle with no surface. The sweep operation produced a hollow object. A procedural wood (solid texture) was applied and rendered using the Phong algorithm.

Number of steps

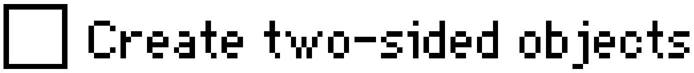
This field determines the number of times the Step Transformations parameters are repeated on the sweep template. To determine the total degrees of rotation for the entire object, you must multiply the rotation setting of each step by the number of steps.

It's important to realize that the multiplication of these two fields determines the overall length of the finished sweep object. Once the Step Transformations are established for a sweep operation, to make the sweep object longer or shorter, you need only increase or decrease the number of steps to be repeated.

For best results, you should keep the rotation for each step just under thirty degrees (30°). Remember, thrity degrees is the default breaking angle used by the Extrude, Lathe, and Sweep modelers. Because the Sweep modeler produces a smoothed surface of triangulated polygonal mesh, it is not necessary to use an excessive number of small steps to obtain a smooth surface. The surface will render as a smooth surface regardless of the fineness of the polygons, as long as the angle between steps is less than thrity degrees.

Refractive textures on sweep objects

If you are planning to apply a texture with transparency and refraction on a sweep object, you will want to also give it correct optical properties.



□ Create two-sided objects

You have the option of making polygons created with the Sweep modeler either single-sided or double-sided. This is set by checking a field in the lower left corner of the Lathe dialog box. Single-sided polygons are only visible to the rendering algorithms from one side; double-sided are visible from either side.

Transparent objects made of single-sided polygons refract light coming through them as if the object were solid. Objects with double-sided polygons appear hollow…as if it is just a shell.

Refraction is a rendering effect available only in raytraced images.

Re-sweep option

Adjusting a sweep object is as simple as double-clicking on the object to re-sweep. Re-sweeping allows you to adjust the sweep settings and try again. This may be done as many times as needed to achieve the desired results.

Vision3d identifies each object's native modeler in the event that you wish to modify an object later. Through this link, you are able to bring up the Sweep dialog with an object's current parameters at any time and from any view. You do not have to be in the same view as the original sweep.

You can also re-sweep an object by first selecting it then choosing the Modify/Re-sweep command from the Modeling menu.

The link can be broken by using the Ungroup command or by using another modeler, such as the 3d Sculpter, Smooth, or Facet extensions, to modify the object in other ways. Using a different modeler then links the sweep object to that extension as its modeler.

Caveats

- Sweeping an excessive number of segments is not necessary for surface smoothness. You should use the lowest number of steps that can adequately describe the surface.
- If you expect to see the inside surface of a swept object, either because it is hollow or the viewing position is inside the object, you

must use double-sided polygons. Otherwise, the inside surface will not be visible to the renderer(s).

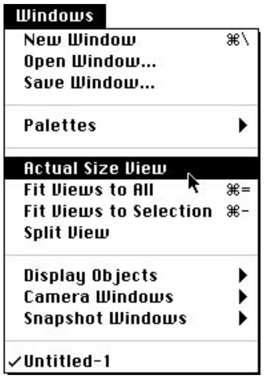
Commands and controls used in modeling operations

The following commands are considered pertinent to the modeling operations of the software. A detailed explanation is included with each command or control item listing.

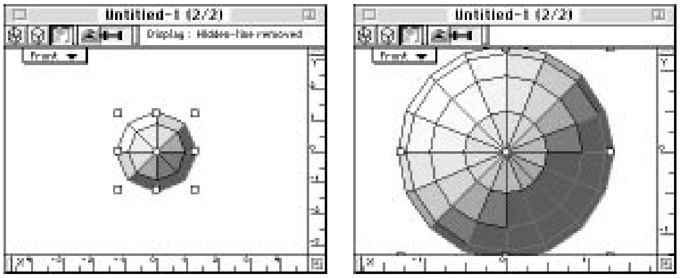
Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut to any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

Actual Size command (Windows menu)

Use this command to reset all views within the active window to display the actual size of objects in the model. This means all magnify or scale changes previously applied to the window (actually, views within the window) are removed.



It also means that the view set center for that window is moved to 0, 0, 0 in the model space and that the perspective setting is set to Orthographic so that the rulers can be made available to the view(s), if they are enabled.



Depending on the current position and magnification of the active window, use of this command can radically shift the position of the view set and change the way objects are displayed in the view(s).

It is useful for “starting over” when you aren’t sure where you are in the model space and at what scale the views are presenting the objects you’ve inserted or drawn.

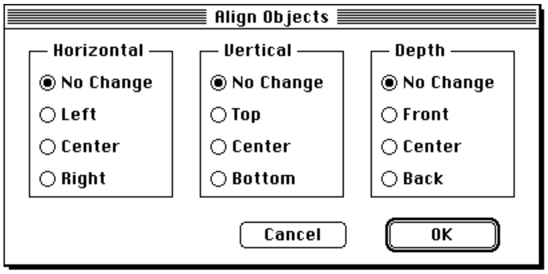
It’s also useful for viewing objects at their actual size…hence, the command’s name. Note that the rulers also correspond to real world scale — i.e., one inch on the rulers is actually one inch after using this command. Of course, this is all based on your monitor actually showing 72 dots per inch on the screen. If it is of a higher or lower resolution, the size displayed will not be exactly actual size.

Align command (Modeling menu)

The Align command allows selected objects to be aligned on the horizontal, vertical, or depth axis, either separately or any combination of the three. Align is available when two or more objects are selected. The alignment references in the dialog box (left, center, top, front, etc.) are relative to the active view.

The Align command is useful to line up objects exactly. Horizontal alignment can be by the left or right sides of the objects, or by their centers. The vertical alignment can be by the tops, bottoms or centers of objects. And depth alignments are by front, back or object centers.

Choosing Align in the Options menu opens a dialog. It allows you to specify which combination of alignments you want to effect. The dialog defaults to “No Change” on each axis when first open.



Align Objects

Horizontal
● No Change
○ Left
○ Center
○ Right

Vertical
● No Change
○ Top
○ Center
○ Bottom

Depth
● No Change
○ Front
○ Center
○ Back

If none of the selected objects is locked, the location of the outermost object on each axis determines the boundary for the edge alignment point. Objects are moved to the point of alignment on each axis chosen. If one of the selected objects is locked, the alignment keysto that object. If more than one selected object is locked, the command uses the outermost extents of the locked objects as the alignment boundaries.

Clicking on the OK button or pushing the Return key executes the alignments on the axes specified in the dialog.

Camera windows (Windows menu)

The Camera Windows command displays a list of the Camera objects inserted into the model. A Camera window is available for each Camera object present in the model.

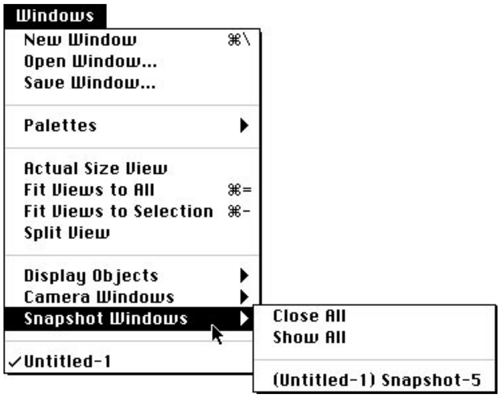
This command provides access to camera objects by name through a sub-menu. If no camera object has been inserted in the active model, the Camera Windows command is dimmed. Selecting one of the camera objects in the sub-menu opens the camera window for that object or makes it the active window, if already open. You may also open a camera window by double-clicking on the camera object directly in the active view of a modeling window.

Camera windows are a special type of window. Each one contains a view of the model from the position of a camera object inserted in the model. It is not attached to any view set, as are the views in modeling windows. Camera objects (and their windows) are free to move anywhere and have any orientation in the model.

Camera objects are inserted using the Camera Object tool on the Tool palette. One or more camera objects may be inserted in a model, each with their own camera window. Only one view is displayed in a camera window, from the orientation of the camera object attached to that window.

Snapshot Windows (Windows menu)

The Snapshot Windows command displays a sub-menu listing all open rendering windows. Snapshot Windows is available when one or more rendering windows are open. If no rendering window is open, the command is dimmed.



When a rendering window is the active window, a check mark appears in front of the window's name. Included in the sub-menu are two additional window-handling commands specifically for rendering windows. These commands bring all rendering windows forward (from behind modeling windows, if hidden) so they can be seen and close all open rendering windows.

If the Close All command is chosen, the rendering windows are made the active window one at a time and their saved status determined. In this manner, you can choose to discard a rendering or save those in progress as suspended renderings for future completion.

Duplicate command (Edit menu, ⌘D)

The Duplicate command creates a copy of the selection, offset by some distance in 3-D space. Duplicate allows specifying the distance and direction between duplicated objects.

To use this feature, select an object or group of objects and duplicate it. The newly-duplicated objects appear slightly above and to the right of the original in the active view. Drag the duplicate to the desired position relative to the original, using any view necessary. Then select Duplicate again. Vision3d will offset the second duplicate from its predecessor exactly as the first duplicate was offset from the original.

All properties of the original object(s) are copied in the duplicates, including geometry, textures, animation path, shapes hierarchy, etc.

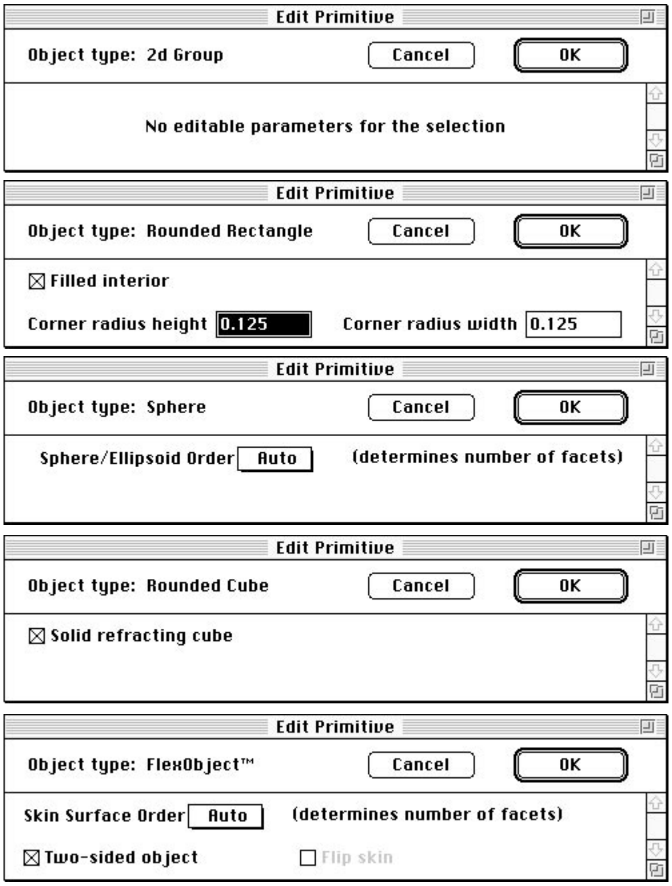
Edit Primitive command (Modeling menu)

The Edit Primitive command displays a dialog to allow fine-tuning of object parameters. The dialog's fields change depending on the type of object selected.

The Edit Primitive command is available whenever an object is selected. Multiple objects may be selected for adjustment at one time, but if they are different types of objects, the command is dimmed.

Similarly, if an object is selected for which there are no adjustable parameters, the command may still be available, but the dialog does not allow any alteration.

Here is a sample of the many different variations of the Edit Primitive dialog.



Edit Primitive

Object type: 2d Group

No editable parameters for the selection

Edit Primitive

Object type: Rounded Rectangle

☒ Filled interior

Corner radius height 0.125 Corner radius width 0.125

Edit Primitive

Object type: Sphere

Sphere/Ellipsoid Order Auto (determines number of facets)

Edit Primitive

Object type: Rounded Cube

☒ Solid refracting cube

Edit Primitive

Object type: FlexObject™

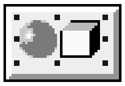
Skin Surface Order Auto (determines number of facets)

☒ Two-sided object □ Flip skin

Different object types have different parameters that appear in this dialog. They control the object's representation in the modeling windows and the way they render in scenes. Through this mechanism, each object, light source, or camera object may be customized to behave as needed in various situations. This command, with its various dialogs, can be a very powerful instrument in the modeling process.

Group command (Modeling menu, ⌘G)

The Group command combines two or more selected objects into one element so that it acts as a single entity. This command is available when two or more individual objects or other grouped entities are selected. There can be as many grouping levels as desired. The button shown in the illustration on the following page also appears on the Button bar to execute this command from there.



Any combination of objects and/or light sources can be grouped together. The same basic operations can be performed on a group that apply to a single item. A selected group is contained within one set of selection handles. A single bounding box appears whenever the group is moved, rotated or scaled. A group can contain other groups, i.e. — two or more groups can be grouped together.

Individual objects within a group cannot be accessed. However, shapes that are part of a group may still be accessed in their own workspaces using the shape-editing commands. Changes made to a shape in its workspace will appear in any instantiations of the shape, even if it is inside of a group at the model level.

Of course, if there's a Group command, there must be an Ungroup command. It is discussed later in this section of the chapter.

Hide Selected command (Selection menu, ⌘3)

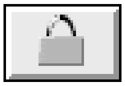
The Hide Selected command allows you to hide objects in the model work space. This feature is useful when trying to edit obscured interior or hidden objects. Or when rendering two versions of a model, with and without certain objects.

This command can also be used on light source icons and camera objects. Even though they are not actual objects, they respond to the Hide Selected command as if they were. However, Global lights (on the Lighting palette) cannot be hidden.

If a selected object is an instantiation of a shape, only the instance is hidden. Other instantiations are still visible. The shape itself is still accessible for editing in its own workspace. The Hide Shape command in the Shapes menu allows you to hide all instances of a shape, but only works with named shapes.

Lock command (Modeling menu)

The Lock command anchors the position, orientation, size, and applied textures of a selected object or group. This command is available whenever one or more unlocked object is selected. Additionally, the button shown in the following illustration allows you to execute this command from the Button bar.



Use this command to prevent accidental changes to an object. Any object, including cameras, shapes or light sources can be locked. When an object is locked, its selection handles change appearance to provide visual feedback.

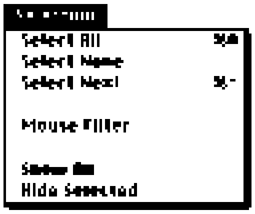
Attempting to manipulate a locked object will cause the cursor to change to a stop sign shape when positioned over a handle and provoke the System beep.

While a locked object's texture cannot be changed, the texture definition itself may be edited using the texture-editing commands found in the Textures menu and on the Textures palette.

Note that locked shape instances cannot be manipulated directly. However, the shape itself can still be accessed using the shape-editing commands.

Mouse Filter command (Selection menu)

In the course of editing models, you may come upon a situation where objects are inaccessible, either because they are inside another object or behind an object in the active view. It may be important to select the object for purposes of moving, rotating, scaling, or applying a texture.



To better deal with the three-dimensional world of your model, Vision3d provides a collection of selection commands and selection masks to determine how and which objects to select. These commands are found in the Selection menu (or the Selection sub-menu of the Edit menu on smaller monitors).

There are three ways you can select hard-to-reach objects with conventional techniques:

- **Move to another view.** If the object is behind another object, you can sometimes use another view that provides better access to the object.
- **Drag a selection marquee around the desired object.** If you can encompass the desired object without also fully encompassing the object in front, only the object behind or inside will be selected. If

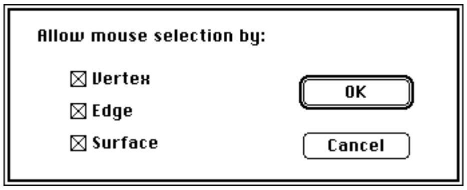
other objects are also selected, you can use the Shift key to un-select those objects while still retaining the selection on the desired object.

- **Temporarily hide the objects that are in the way.** After completing the work you wanted to perform on the hard to reach object, un-hide the other objects.

Often, a little inventiveness can accomplish what you need. Additionally, you can use two other ways provided in the Mouse Filter command to facilitate specific selection attempts:

- **Use the "Select Next" command.** Vision3d interrogates the point where the mouse was clicked to determine if any objects are behind the selected object along the depth axis. If a selectable object exists, the selection transfers or shifts to that object. If no object is found, the selection remains with the first object. Each time this command is called, the process is repeated. The selection rotates through the qualifying objects in sequence, repeating the cycle from first to last as often as the Select Next command is called.
- **Change the selection filters.** By setting the selection parameters to select objects only when the mouse is clicked on a vertex, for example, you can select the desired object through the surface of another object in front of it. The Mouse Filter command is provided to set the cursor's selection behavior.

  The Mouse Filter command displays a dialog with various options for selecting objects. These are check box items that may be enabled or disabled as needed to select surfaces only, vertexes only, or edges only. Or you may use any combination of the three. The check boxes defining the selection parameters in the dialog are retained between sessions or until changed.



Allow mouse selection by:

☒ Vertex
☒ Edge
☒ Surface

  Based on these mouse filter settings, each time the selection cursor is clicked in the active view, Vision3d interrogates the point of selection to see if it matches any of the enabled mouse filters. If one or more of the options qualifies, the object is selected. If no match is made, no selection event occurs.

NOTE: *If you turn off ALL three selection filters in this dialog, clicking on an object will never select it, since no condition exists to match the set-*

tings. There is no situation that recommends turning off all three filters. However, if that situation exists, you can still select objects by dragging a selection marquee around them.

Modify command (Modeling menu)

The Modify command allows more extensive modifications to objects than the Edit Primitive command. Modify is available when an object is selected. However, its response to a call depends on the type of object selected and whether a "native" modeler is present for that object.

If the selected object is a primitive drawn by one of the tools on the tool palette, modify will bring up the extension editor that lets you alter primitive objects, if available. If an object has been drawn with a different modeler (usually present as an extension) or has been converted to a different class of object, then calling the Modify command automatically brings up the modeler that last edited that object.

If a modeler cannot be identified for a selected object, the Modify command will open the Edit Primitive dialog box instead.

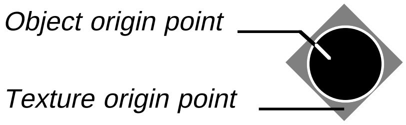
As noted in the various discussion on modeling extensions, you can manually select an appropriate modeling extension at any time to modify an object, which then becomes the native modeler for that object and will be automatically accessed through the Modify command in the future.

Origin points

This is not a command in a menu, but rather a feature of the handles and object manipulation mechanisms that become available when an object is selected.

Each object has both an object origin point and a texture origin point. The object origin point for an object is always visible when it is selected. The texture origin point is visible only if a texture is applied to the object AND the Show Texture Centers command is enabled in the Textures menu.

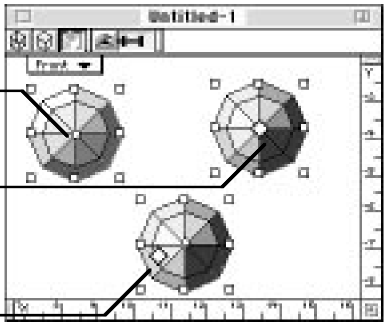
These points appear respectively as a small round dot and a small diamond whenever an object is selected. When these two origin points are both visible and are coincident (together), the object origin (dot) appears in front of the texture origin (diamond) in all views.



The texture origin point is visible only when a texture has been applied to the object AND when the Show Texture Center command is enabled in the Texture menu. If no texture is applied, no texture origin point is visible. Also, no texture origin point is visible when the selected object is a shape instantiation, since the texture has been applied in the shape's work space, not at the current level.

NOTE: *The Show Texture Center command is disabled whenever a new session begins, even if it was enabled in the previous session. Since it is rare to move the texture origin point, you must consciously decide to show texture origin points in order to manipulate them on the objects.*

Normally, both origin points are located in the geometric center of an object. However, they may be shifted in relation to the object, but ONLY IF the Show Animation Paths command is disabled in the Animation menu.



Only object origin point showing (no texture applied)

Object and texture origin points together

Texture origin point separated from object origin point

On new objects these two origin points are coincident. They can be separated, if needed, using the Control key. Either point may be repositioned independently in special circumstances. For most situations they usually remain together. If the two points are together, they remain together when dragging to a new position.

The object origin point is discussed here because it is closely related to modeling tasks and techniques. The texture origin point is also discussed further in *Chapter 5: Applying Textures*, since it is obviously related to applying textures on objects.

While the default location of the origin points in objects created by Vision3d is at the geometric center, objects imported from other applications may not locate their centers in the same manner. These can be adjusted, if necessary — the Re-Center command in the Modeling menu being the easiest method.

Object origin point

The position of the object origin point (●) determines how a selected object responds to certain operations using the object manipulation tools:

- **It sets the axis for object rotation.**
- **It is where a child object is linked to its parent.**
- **It locates the object on its animation path.**

Texture origin point

The position of the texture origin point (◆) determines how surface maps contained in applied textures are mapped to objects:

- **It is where the center of a surface map is placed on an object.**
- **It sets the axis for surface map rotation.**

Moving origin points

To move the origin points relative to the object, but maintain their spatial arrangement relative to each other:

1. **Select the pointer tool.** While any of the three object manipulation tools can select objects, only the Object Move tool (a.k.a Pointer tool) can move origin points.
2. **Position the cursor over the object origin point.** If the selected object has a texture applied, you will see two origin points, one behind the other. If no texture has been applied to the object, only the object origin point is visible.
3. **Hold down the Command key.** Without the Command key, dragging the origin point(s) also moves the object (even with the Show Animation Paths command disabled).
4. **Drag it to the new position.** The two origin points will move as one.

Separating origin points

To separate the two points, hold down the Control key while dragging either of them. If both origin points are visible, the object origin point is in front and may be moved by simply clicking and dragging the dot. The diamond shape of the texture origin point always appears behind the object origin point when they are together, but may be selected by clicking and dragging one of the visible diamond points. With the Control key down, the chosen origin point will be moved while leaving the other point, as well as the object itself, stationary.

Just as the object itself can be moved any direction in three-dimensional space, an origin point can be relocated in any direction, on any axis, and at any distance from the object's geometric center. They can only be moved horizontally or vertically in the plane of a particular view. To move the point on the depth axis, use a perpendicular view of the object.

Modifier keys that apply to positioning object and texture origin points:

A. **Shift key —** Constrains the cursor to drag the origin points horizontally or vertically only. Unconstrained, the cursor can move diagonally.

B. **Command key** — Moves the origin points (both object and texture, if visible) without moving the object. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

C. **Control key —** Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other as well. With the Control key down, either the object origin point or texture origin point (if visible) can be moved with moving the other point or the object itself.

Re-center command (Modeling menu)

Use this command to position an object's origin point(s) back at its geometric center. Re-centering the origin point is useful anytime it has been moved and you wish to reset it for further manipulation. This can occur for a number of reasons:

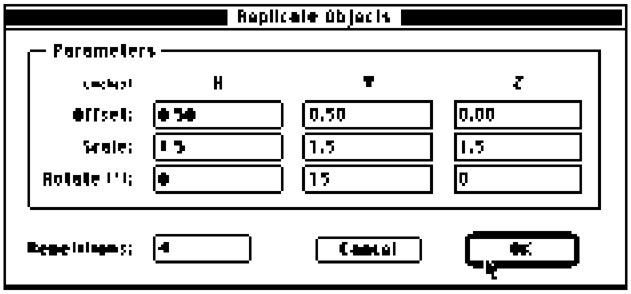
1. **After linking two objects,** where you want the child-object to rotate about its own geometric center rather than the parent's
2. **After manually moving the origin point for some operation.** This is often done for special rotation needs
3. **After you scale an object with the origin point outside the object.** If the origin point is outside the object when scaled, it does not move proportionally to the scale, as it does when inside the object.
4. **When you adjust an object's position relative to its animation path.** The origin point is that part of an object that actually follows an animation path…bringing the object along with it.

The Re-Center command is available whenever a selection exists. If more than one object or group is selected, each origin point for each entity is returned to their respective geometric centers.

Replicate command (Edit menu, ⌘R)

The Replicate command creates multiple copies of a selected object or group of objects, modifying position, orientation and size with each iteration.

Vision3d displays a dialog box allowing numeric entry for each replication parameter. The values are in the same units defined in the Set Rulers command.



(inches)	X	Y	Z
Offset:	0.50	0.50	0.00
Scale:	1.5	1.5	1.5
Rotate (°):	0	15	0

Repetitions: 4

The fields perform the following functions:

- **Offset.** These fields allow specifying the offset distances along one or more of the three axes for each successive copy. For your convenience, these values are always relative to the active view.
- **Scale.** These fields allow specifying how much larger or smaller each copy should be relative to its predecessor. A value of two would double the size of the object(s) with each repetition. A value of one, which is the default value, would make all copies the same size as the original. A value of 0.5 would halve the size with each repetition.
- **Rotate.** Specifies the amount (in degrees) on all three axes by which each repetition should be rotated from its predecessor.
- **Repetitions.** This field specifies how many times the Replicate command should copy the object(s) and apply the transformations specified in the replication parameters.

Select All command (Selection menu, ⌘A)

The Select All command selects all visible objects and light sources in the current work space. Select All is available when a model is active and there is at least one object in the model. The command affects only the active document.

Selected objects can be identified by selection handles surrounding them. Selected objects are affected by subsequent actions or commands. Un-selected objects are generally unaffected by commands.

Hidden objects or Shapes in the model space are not selected.

Holding down the Shift key while clicking on objects adds to or removes them from the group of selected objects. In this manner, a selected object may be de-selected without affecting the selection status of other objects.

Select Next command (Selection menu, ⌘~)

The Select Next command selects the next object after the one currently selected while in the active view. Select Next allows access to objects that may be inside or behind other objects.

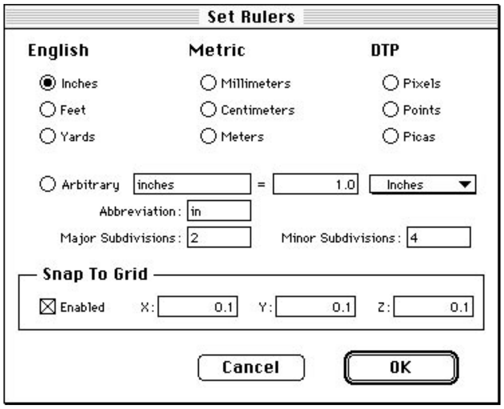
The selection process is applied at the spot where the cursor was clicked to make the selection. Each time Select Next is called, Vision3d looks along the active view’s Z or depth axis to determine if any other objects exist. If no object is behind the one selected (at the point of selection), the current object remains selected.

Select None command (Selection menu, ⌘1)

The Select None command de-selects all currently selected objects. The command affects only the active document. Using this command is the same as clicking the mouse button while the cursor is in the active view, but away from any objects.

Set Rulers command (Edit menu)

The measurement system defining the scale of objects, and that is displayed in the rulers, is set through this command. Set Ruler is available whenever a document is open. The defined unit of measure is retained as a default for all future sessions in Vision3d, or until changed by using this command again.



Set Rulers

English	Metric	DTP
● Inches	○ Millimeters	○ Pixels
○ Feet	○ Centimeters	○ Points
○ Yards	○ Meters	○ Picas

○ Arbitrary inches = 1.0 Inches

Abbreviation: in

Major Subdivisions: 2 Minor Subdivisions: 4

Snap To Grid

☒ Enabled X: 0.1 Y: 0.1 Z: 0.1

Set Rulers presents a dialog box to choose from pre-defined units of measure or to formulate your own. You may also set the major and minor graduations that appear on the rulers.

The Set Rulers command also includes a check box to enable a 3-D grid in the model space. You may set the detent spacing for the grid independently for each axis. The units for the grid are the same as those chosen for the rulers and scale in the views. The grid is not visible when enabled to avoid confusion with objects, but its effects are noticeable when dragging the cursor during object drawing operations or when adjusting object positions.

To enable the grid, turn on the check box. To turn off the grid, click on the check box again.

The measuring units may be changed as often as needed.

Both the ruler units and the grid settings are global in nature, meaning that they are used for all models. The chosen unit is retained between sessions. However, models do not save the units in which they were constructed. If you change measuring units, then open a previous model assembled using another scale, the current units and scale choice is adopted by the opened model.

Modifier keys that apply to moving objects with the grid enabled:

A. **Command key —** Temporarily disables the grid detents while the object is being moved. The key must be held down AFTER the move operation is begun to have the modifying effect described here. Holding down the Command key BEFORE the tool is clicked on the object to begin dragging causes the object origin point to remain stationary.

Show All command (Selection menu, ⌘4)

The Show All command removes the hidden designation from all objects in the active model. When an object or group of objects is hidden (using the Hide Selected command), they become invisible to the renderers in the modeling views as well as the higher-level rendering algorithms. They are also invisible to any selection method.

The only way to gain access to hidden objects is through the use of this command. This command has no effect on shape hidden with the Hide Shape command in the Shapes menu.

Show/Hide Tool Palette (Windows menu

The Show Tool Palette command displays the Tool palette. Show Tool Palette is available when the Tool palette is hidden. Hide Tool Palette is available when the Tool palette is showing. Choosing the Hide

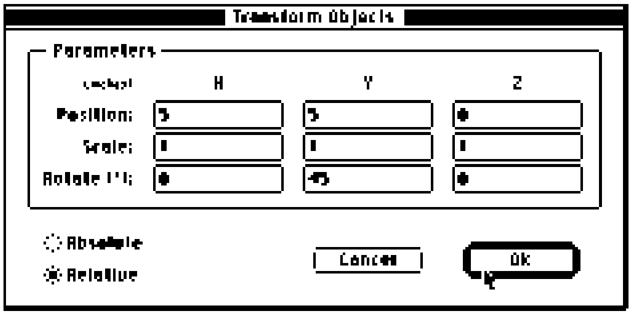
Tool Palette command is the same as clicking on the close box of the Tool palette or on the show/hide button on the Button bar for the Tool palette.

Vision3d displays only one Tool palette, even if multiple model documents are open. This and all the other palettes are always associated with the active document.

The Tool palette's show/hide status when a new session is started is determined by the previous session. If the palette was showing when the application was quit the last session, it will be showing when a model is opened in the current session. If it was hidden at quitting last time, it will be hidden on startup this time…the status is retained from session to session.

Transform command (Modeling menu, ⌘T)

The Transform command enables you to use numeric input for transformation operations, i.e. moving, rotation, and scaling of objects. Transform is available when any object or set of objects is selected.



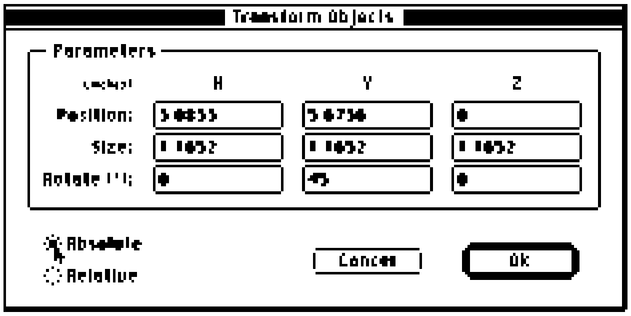
Transform Objects

Parameters

(inches)	X	Y	Z
Position:	5	5	0
Scale:	1	1	1
Rotate (°):	0	45	0

○ Absolute
● Relative

Calling the Transform command opens a dialog box where the transformation values may be entered or edited. Fields in the dialog allow the option of specifying relative offset from the selected object(s) current position on any or all of the three axes or entering absolute values for a new position, rotation or scale based on model coordinates.



Transform Objects

Parameters

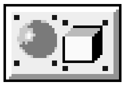
(inches)	X	Y	Z
Position:	5.0855	5.0734	0
Size:	1.1052	1.1052	1.1052
Rotate (°):	0	45	0

● Absolute
○ Relative

If the Absolute radio button is clicked, the current transformation information for the selected object is entered into the fields. You can then alter those values to re-position, re-size, and change the rotation as desired. Clicking the "OK" button or pushing the Return key accepts the values and performs the transformation.

Ungroup command (Modeling menu, ⌘U)

The Ungroup command splits a previously-grouped entity into separately-selectable elements. Ungrouping allows direct handling and editing of objects that were inside a group. This command is available when one or more ungroup-able objects are selected. Additionally, the button shown in the following illustration also appears on the Button bar to execute this command.



Use this command to break down complex groups into their individual components. When a group contains other groups, the first call of this command ungroups the outermost group only. Each call of this command ungroups the next level. Further ungrouping can sometimes even allow access to individual polygons making up objects.

Shape instantiations may be ungrouped at the model level, as well. While possible for special situations, it is not usually desirable. If a shape instantiation is ungrouped, its hierarchical connection to the shape is severed and it becomes a regular object. Breaking the shape connection also loses all the benefits of hierarchical modeling: 1) automatic updating of all instantiations in the model when a shape is changed, 2) memory efficiency, 3) carrying texture application at the shape level to the instantiation at the model level. Shape instantiations are edited most efficiently in the original shapes' work spaces.

Unlink command (Modeling menu)

Performs a global severing of any links to a selected object or group of objects. The command becomes active when an object with an existing link is selected.

Unlock command (Modeling menu)

The Unlock command removes the editing constraint on locked items. This command is available when one or more locked objects are selected. Additionally, the button shown in the illustration on the following page also appears on the Button bar to execute this command.

An object must be unlocked before changes to the object will be allowed using the object manipulation tools. Note that a locked shape instance is only locked at that level. It may still be altered at the previous level and any changes will be propagated to the locked instance(s) of that shape.

When an object is locked, its selection handles change appearance to provide visual feedback during locking and unlocking operations.

CHAPTER 4: USING SHAPES

Drawing one object, applying a texture, and rendering an image of it is a simple enough task. Even assembling several objects together into something actually recognizable probably won't tax your 3-D modeling skills too much. It's always a good idea to start with a modest project — one with which you can enjoy some degree of success — then move on to something more ambitious. But how about when you want to do something really meaningful?

Inevitably the time will come when you'll feel you're ready to take on THE *PROJECT*. You know, that's the one someone is willing to pay you to do, so you want it to be the best 3-D rendering ever and win all kinds of awards. It will have lots of objects. …And lots of detail. …And it must be assembled with a high precision. …And, of course, it must be rendered in high resolution.

When that day comes, will you know where to begin? Will you have the skills to navigate through a project that big? How will you keep all the various elements organized?

Fortunately, Vision3d offers a feature called Shapes to help in just this type of situation. Shapes allow you to break down big models into smaller "bite-sized" chunks. They allow you to work on just one component at a time, manipulating it in its own separate workspace — with its own set of windows, if necessary — then insert it into the model for assembly with other shapes.

But the use of shapes includes not only the ability to set up logical groupings of objects, it also allows you to impose a structure to a project. The bigger the project, the more helpful a shapes structure or hierarchy can be.

In addition to making big projects easier to construct, this feature also encourages experimentation because changes made to a shape can be propagated easily to each instance of that shape throughout the model.

How shapes work

Any object that is constructed as a shape may be made part of another shape. This is done by inserting one shape into another shape's workspace, using the Insert command in the Shapes menu. It is important to understand that the inserted shape is actually an instance (or, more formally, an instantiation) of the original object, and not the actual object. An instance has all of its shape's characteristics, including transformation (size, position, rotation), geometry, and texture.

As an instance, it is not directly editable, but can be given additional transformations and have a different texture applied. The transformations and textures applied to the instance level over-ride those applied at a shape level; the shape being considered a lower level than its instance. Multiple instances then may be used as building blocks of another shape, or several different shapes. There is no limitation on the different ways shapes and their instances may be combined to form the desired objects. Shapes may even be inserted into themselves — a condition called recursion — often with interesting results.

An added feature is that shapes can be referenced by name. If the names given the shapes are sufficiently descriptive, this also helps keep track of the organization of a project's structure and method of construction.

A complex shape assembled in this manner, using the instances of simpler shapes, is said to be hierarchical in nature. In a hierarchical model, the beginning object is called the root shape. You can have as many (or as few) root shapes as you need. These are usually a single object, but can have more than one component if that is deemed logical. Remember, you are the one that sets up the structure; Vision3d imposes no limits. A root shape can be used in other shapes (higher up the structure) to make more complex shapes…or inserted directly into the model.

Saving time with instances

As you assemble a model, you may change your mind about some of the objects — either the geometry, the type of object, or the texture used. You may even want to make a different object, constructed with a different modeler.

If the object in question was made as a shape, you can simply return to that shape's workspace (by choosing its name from the Edit Shape sub-menu in the Shapes menu), and make the desired adjustment(s) to the object. Those changes will immediately propagate through the model wherever instances of that shape are used.

Note that whatever you put in a shape's workspace becomes part of its instance(s) as well. If you replace an object in the shape with another one, the new object appears in the instances in place of the previous object. If you delete the object(s) inside a shape, the objects in the instances will disappear as well, but the instances are still where you put them, acting as a "place holder." If you later draw an object, or add an instance of another shape, into that shape workspace, it causes the instances to suddenly appear again with the new object.

If you completely delete a shape workspace from the model (through the Browse Shapes command in the Shapes menu), all instances disappear as well, but this does not leave any place holders.

Saving memory with instances

In addition to facilitating model construction, there is another distinct advantage to using shapes. Multiple insertions of a shape take much less memory than using copies of a regular object.

Inserting a shape instance adds about 20K to the model. Compare this to the memory required to completely duplicate an object, especially if the object is built as a polygonal mesh, with LOTS of vertices to store. This technique can be used to lessen the total number of objects in a large model, which can often make the difference between successfully completing the project or just plain running out of memory.

One caveat

A shape may be used as often as needed, and at multiple levels in the hierarchical structure of a model. However, because you have the freedom to build any structure you wish with these shapes, there is a tendency to not use them efficiently.

A hierarchical model should have a linear progression from the creation of root shape(s) to their ultimate presence in the model space. It is a good procedure to diagram the hierarchical structure on paper before you begin, so you won't get lost or forget an important element. Otherwise, you could end up using more memory than you should, and editing the shapes to make adjustments could be a lot more cumbersome.

An hierarchical example

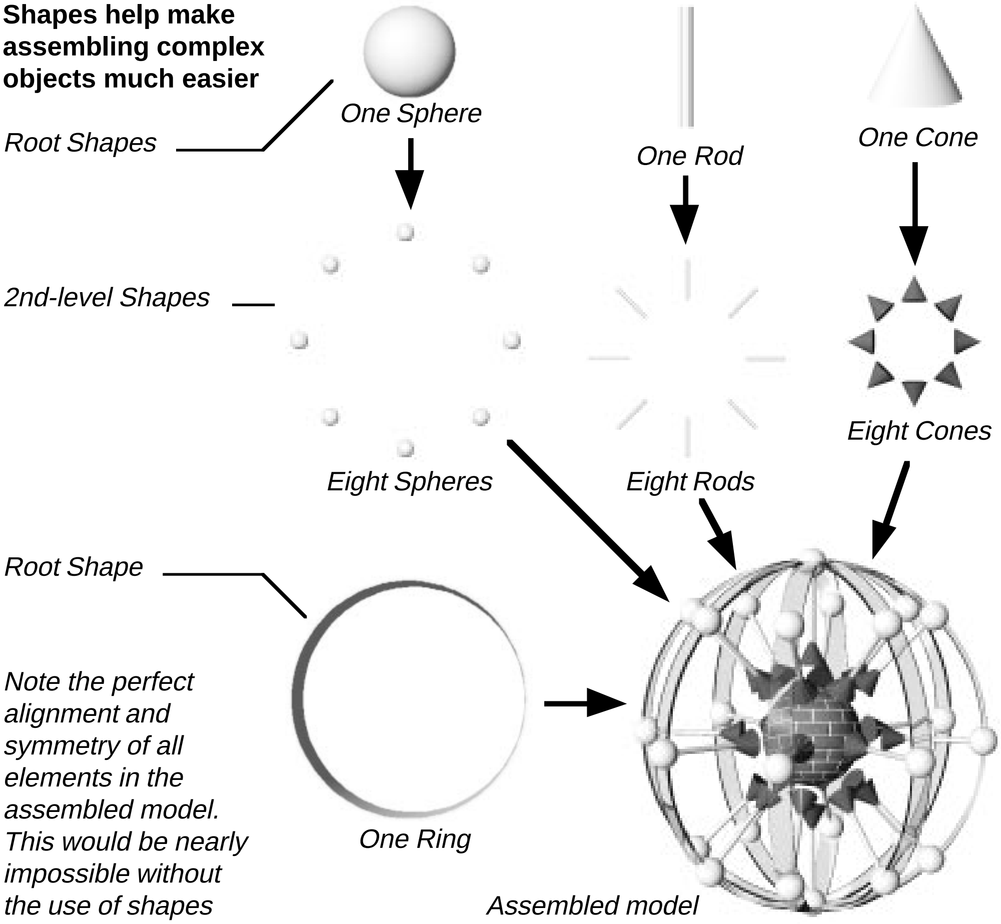
While shapes are useful when used to construct larger, more complex objects, they may also be used to assemble components requiring repetitive elements with exact spatial placement relative to each other.

Using a hierarchical approach in your projects gives you the option to isolate individual components in their own workspaces, which eases the modeling of otherwise difficult tasks.

Building the “Root” Shapes

Suppose you want to make an object that uses just a few objects over and over, but they have to be positioned exactly. In the following illustration, the root shapes were a single sphere, a single rod (actually a long, skinny cylinder), and a single cone. All these objects are 3-D primitives. These root shapes were named, “one sphere,” “one cone,” and “one rod.”

Shapes help make assembling complex objects much easier



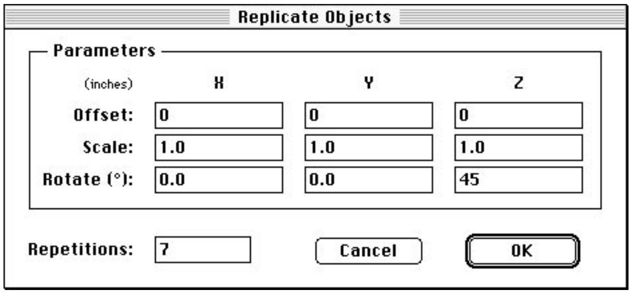
Note the perfect alignment and symmetry of all elements in the assembled model. This would be nearly impossible without the use of shapes

In their shape workspaces, the objects were drawn to the appropriate size using the rulers and the Transform command (to verify EXACT dimensions and position). While in the Transform command, each of these objects were also positioned at the center of their workspace, at coordinates 0, 0, 0 for the X, Y, and Z axes respectively. Textures were applied at this level, as well.

The next level

The three root shapes were inserted into three additional 2nd-level shape workspaces. The original scale was maintained at this level. The 2nd-level shapes were given the names, “8 spheres,” “8 cones,” and “8 rods.” In each new shape, the instance from the corresponding root shape were moved on the positive Y axis (that is, up in the Front view) an appropriate amount.

For the sphere instance, the coordinates were set at 0, 4, 0 (X, Y, Z axes). With the sphere instance still selected, its object origin point was dragged from its geometric center downward to the zero Y coordinate again. The Shift key was used to constrain the drag operation so the origin point was EXACTLY at 0, 0, 0, while the object was offset 4 units upward.



Replicate Objects

Parameters

(inches)	X	Y	Z
Offset:	0	0	0
Scale:	1.0	1.0	1.0
Rotate (°):	0.0	0.0	45

Repetitions: 7

Next, the Replicate command was chosen and values entered into its dialog to make 7 repetitions of the sphere instance, each with a 45° rotation on the Z axis from the previous one. There was no change in scale, meaning that they were all made the same size. Of course, these settings are in absolute coordinates, not view relative. Please refer to *Chapter 3: Modeling Objects* for an explanation of the Replicate command, if needed.

The result is a ring of eight spheres — the original, plus seven repetitions — on the same plane but rotated 45° apart, and all exactly the same distance from a common center located at 0, 0, 0 in the workspace.

Note: *Copying a shape instance is the same as inserting the shape again. Thus, in this case, the Replicated copies of the Sphere shape instance are instances as well.*

The same thing was done with the cone and rod shape, making a ring of each in their own 2nd-level shape. Each of these shapes were also positioned so that their ring of objects were centered on 0, 0, 0, but that is not absolutely essential. They could be positioned anywhere you want in their own workspace… and the instances can likewise be positioned anywhere you want in the next level of workspaces. There is total flexibility. In this example, each shape at each level was located at the center of their workspace for consistency.

Third level (the model)

Next, the three 2nd level shapes were instantiated into the model space. The insertion was done without a scale change. As each was inserted, they were positioned at 0, 0, 0. The view was switched to the Top view, and the Replicate command was again chosen. This time the repetition was set for just three times, with a rotation of 45° on the Y axis.

The resulting instances lined up the spheres, rods, and cones exactly.

Another root shape

After the three 2nd-level shapes instances were replicated and everything was in place, it was decided to add one more component, an extruded ring. So, a new shape was made and a hollow circle was drawn in the workspace. This ring was intended to intersect the spheres, so the circle was drawn with the appropriate diameter, then extruded one inch deep.

Unlike the other root shapes, the ring was inserted into the model directly, since nothing else is needed to be done to it. The ring instance was then replicated three times, just like the others. Of course, as expected, the rings lined up exactly.

However, it could have just as easily been inserted into a 2nd-level shape, replicated 3 times to form the multiple rings, then that shape inserted as a single unit.

Two slight adjustments

Actually, once the model was assembled two adjustments were made to obtain the final model shown in the illustration.

It was decided that the rings were too wide, they obscured too much of the other object. So, the ring shape was opened again for editing (through the Edit Shape command), and the extruded ring was scaled narrower using the Object Scale tool. This was done from the Top view while holding down the Option key. That way the scale operation was from the center of the object and the ring was kept in the same relative position in its workspace. This kept the instances from shifting sideways, as they would have if the scale was done from the edge. Of course, the instances of the ring in the model space instantly became thinner as well.

The second adjustment was to the rods. They were not long enough to join up with the spheres. There was a small gap. This was easily changed by opening the rod root shape again and making the rod taller. The Object Scale tool was used to drag the top center handle on the rod. The rod itself stayed in the same relative position, but the length was adjusted in the direction toward the spheres. This took two scale adjustments to get just right. But, by watching the model space as the rod instances were re-drawn longer, it was a simple 15- or 20-second operation to make the adjustment and return back to the model.

Just one regular object

To finish the project, a sphere was placed in the middle of the other constructed elements and given a brick texture. Obviously there was no need to use a shape hierarchy for this part of the model. Since everything had been constructed to center on the 0, 0, 0 coordinates, it was a simple matter to place the sphere at the same coordinates, where a big hole was left.

From any view position you choose, the various elements of this model are in perfect congruence. If a model of this type had been constructed without using the shape feature, it would have been very difficult to position each object in just the right spot and with just the right orientation.

This was just an exercise, so the model has no actual value or meaning. But it graphically demonstrates the power of using hierarchical structures for your own, more meaningful projects.

Summary

Constructing a model from shapes has two valuable benefits…and one minor inconvenience.

Benefits

First, all characteristics of the instantiation in the model are taken from the shape (its geometry, proportions, color, surface maps, etc.) Note, however, that scale, rotation and position may be set independently for each instantiation level.

The second, and main, reason for using a shape instantiation rather than repeating separate objects is efficiency. During the rendering phase, a renderer can potentially save a lot of memory and time because a hierarchical model is presented much more compactly. It allows the renderer to perform a certain amount of processing only once rather than many times. Actual savings, of course, depend on the nature and implementation of the shape.

Perhaps an equally important aspect of model construction via shapes is their use as a conceptual tool. The ability to insert a completed shape with one simple command, rather than requiring a long series of modeling steps each time, makes it easier to manipulate and use shapes.

Limitations

The only other consideration of using shape hierarchy in a model is the lack of access to the internal components of a shape instance while in the model space. Each instantiation comes into the next level as a grouped entity with ties back to the shape definition from which it is linked.

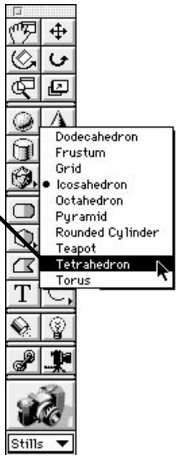
Of course, a shape instance can be ungrouped, if that becomes necessary. However, this is a sign of less-than-optimal hierarchical structure and should be considered carefully for its ramifications to the overall structure of a project.

If an instance is ungrouped, at whatever level that may occur, all ties to the shape are broken and it becomes a separate object or set of objects at the higher level.

While all modifications, including texture specifications, must be made at the shape level, this is also its greatest benefit, in that alterations automatically propagate to all instances in the model.

Making objects for the Pseudo-primitives tool

Several 3-D primitives are included in the product for use with the user-selectable feature of the pseudo-primitives tool. They are referred to as pseudo-primitives because these shapes are actually not primitives, but polygonally defined.



Holding down the mouse button on the Pseudo-primitives tool presents a sub-menu to select a shape for the tool to draw

The choice of which pseudo-primitive the tool will draw is made through a pull-down menu available by holding down the mouse button on the tool's icon. The pull-down menu displays a list of all pseudo-primitives found in the Primitives folder on startup.

The operation of this is discussed fully in Chapter 3. It is mentioned again here, because Shapes are one of the methods for creating your own pseudo-primitives, perhaps the best method. A specific shape can be saved from any model, no matter how complex, through the Browse Shapes command. The Shapes Browser dialog box allows you to select one or more shapes and save them as individual Shape files.

Objects qualify for pseudo-primitive use in four ways:

- **Saved as a Vision3d model**
- **Saved as a StrataVision 3d model**
- **Saved as a StrataType 3d model**
- **Saved as a Shape file**

Their complexity is limited only by your imagination and need, but you should realize that when a pseudo-primitive is drawn in a model, you will not have access to any of the shapes used to make the object. It becomes a "monolithic" object.

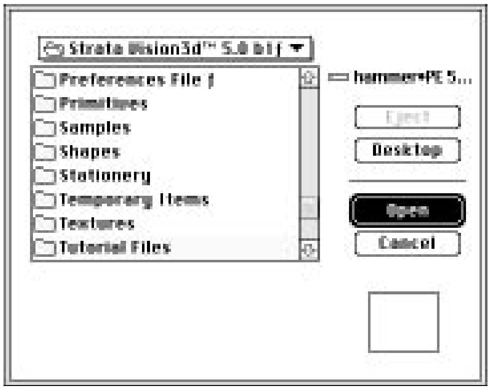
Once saved in one of these formats, you may then place the file in the Primitives folder and conveniently use that object through the Pseudo-primitives tool. Remember, while these objects are handled like primitives, they do not become primitives.

Quick Tutorial

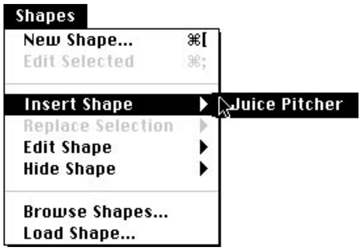
The object of this Quick Tutorial is to give you a basic understanding how shapes work in Vision3d. You will learn how to load a shape, how to create, copy and edit shapes and how shape hierarchy works. You'll also learn how to compose a model from shapes.

Loading the Juice Pitcher shape

1. **Open a new model.** As suggested for all modeling tasks, make sure the three-position lens control is set to Orthographic. Also check that the view orientation is set to Front to begin.
2. **Pull down the Shapes menu and choose the Load Shape command.** Vision3d responds by opening a directory dialog for loading a shape. It opens looking at the Shapes folder within the Vision3d application folder. You can use the controls that allow navigating, previewing, and loading Vision3d shapes.

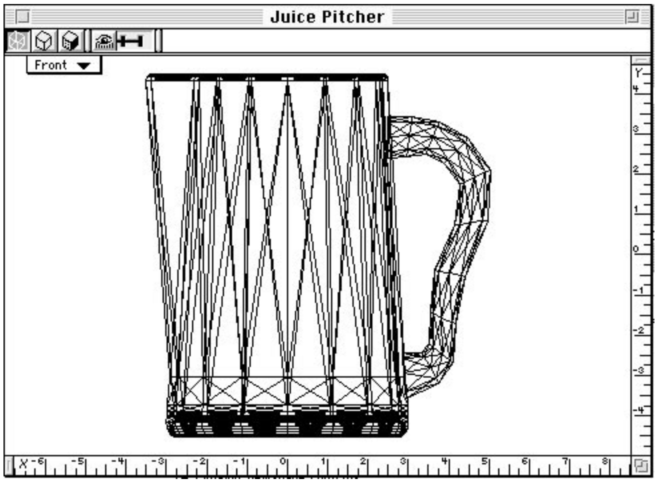


3. **Select a file called Juice Pitcher.** Navigate to the Tutorial Files folder within the Vision3d application folder. Notice Vision3d displays a preview image of the selected shape in a lower corner of the dialog window when a shape file is selected. Each shape file is given a preview image at the time it is saved.
4. **Click on the Load button.** A progress bar is displayed as the selected shape is loaded into the model. Note that the Juice Pitcher shape is loaded into the model, but does not appear in the model window.
5. **Insert the Juice Pitcher into the model space.** Pull down the Shapes menu and choose Juice Pitcher from the Insert Shapes sub-menu.



Vision3d responds by "loading" the cursor with the selected shape, denoting that it is ready to be inserted into the model space.

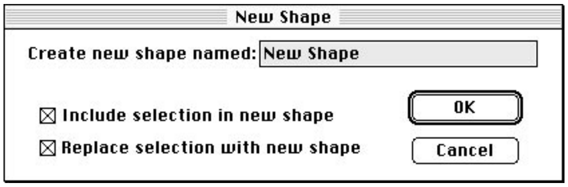
6. **Position the loaded cursor at the center of the model window and click once.** Vision3d responds by drawing an "instance" of the shape in the model window at full size.



In this case, the shape may be too large to fit completely in your model window. If so, use the “Fit Views to All” command found under the Windows menu.

Making a Juice glass

7. **Create a new shape workspace.** Pull down the Shapes menu and choose the “New Shape “ command. Vision3d responds by opening the New Shape dialog box.


New Shape

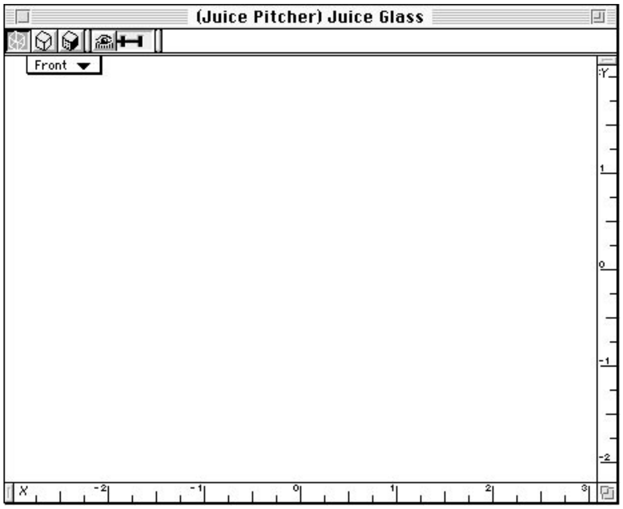
Create new shape named: New Shape

☒ Include selection in new shape

☒ Replace selection with new shape

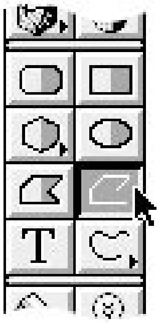
In this dialog, you can give the new shape a unique name. Moreover, two checkboxes allow you to include a selected object or instance in the new shape’s workspace, then replace it with an instance of this new shape. The contents of this dialog is explained further under a discussion of the New Shape command, later in this chapter.

8. **Name the new shape.** Type “Juice Glass” into the name field in the New Shape dialog window. Make sure both check boxes are empty.
9. **Click on the OK button.** A new shape workspace window opens. Note the title of this window (in the title bar).



Shape windows are just like model windows in their operation, but this is where you create and edit object(s) that you want to use as a shape instance in other workspaces. We will use this space to draw a cross-section, then revolve it into a juice glass with the Lathe modeler.

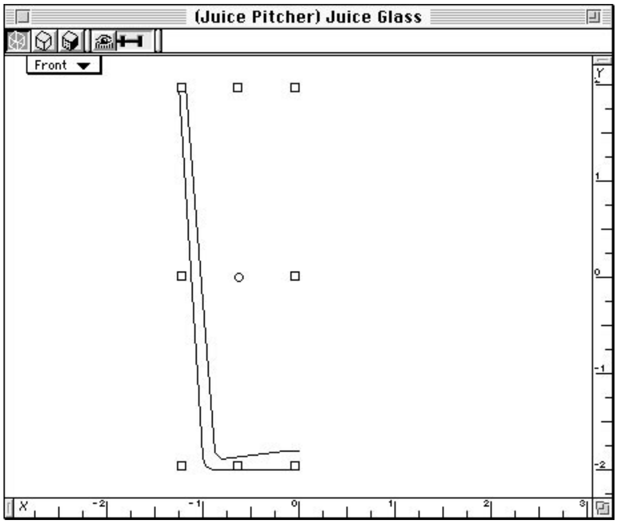
10. **Choose the Open Polyline tool from the Tool palette.** This is the tool that allows you to draw a series of joined straight lines (with vertices at each joint in the line).



Make sure you choose the open line version on the right side of the palette, not the tool that creates closed polygons on the left side.

11. **Draw a cross section of a juice glass.** In the shape window, use the open polyline tool to draw an object similar to the one shown in the following illustration.

Remember, with the Polyline tool, you click once to begin drawing, and click again to change direction. When the cross-section is finished, double click on the last point to stop drawing and enter the 2-D object into the shape workspace.



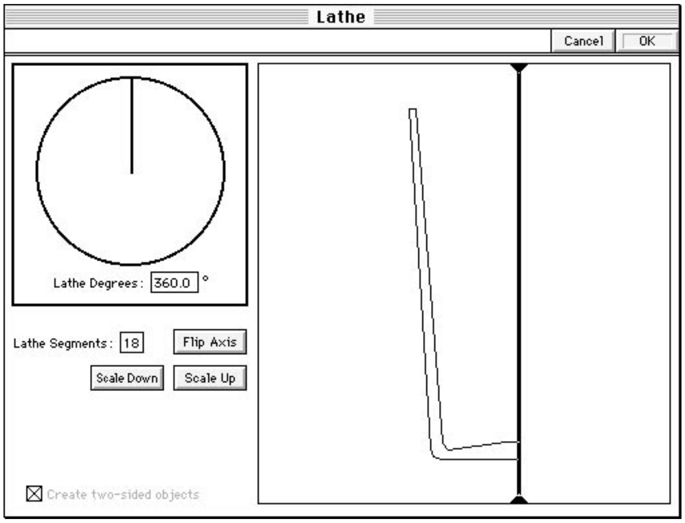
This is our juice glass cross-section. In other Quick Tutorial exercises, you have always started at the top. In this project, however, you will want to give some thickness to the glass. The best way to accomplish this is to start at the bottom center and draw upward to make the outer surface, then continue down again to define the inner surface, and ending at the center again. You will want to add a couple of extra points as you make the turn at the bottom of the glass. If you start at the zero point of the X axis ruler, you can also end there so that the two ends of the polyline line up for lathing. The height of the glass doesn’t matter as much as the proportions.

NOTE: *After drawing this template, you may want to use the 2d Sculpter to adjust the positioning of its vertex points. If you do, you could convert this polyline to a bézier object for a smooth curve on the bottom of the glass. For this project, because that area of the object will not be very visible and in the spirit of minimizing the amount of polygons being created, just keep it in polygonal form and adjust the vertex points to approximate the curve. When lathed, it will look just fine.*

12. **Lathe the template.** With the polyline selected, pull down the Modeling menu and choose the Lathe command from the Swept Surfaces sub-menu. Alternately, you can just click on the Lathe button from the Button bar.

Vision3d responds by opening the Lathe dialog box. The lathe template you just created is displayed on the right side of the dialog.

13. **Set the lathe parameters.** First, you position the lathe axis as shown in the following illustration.



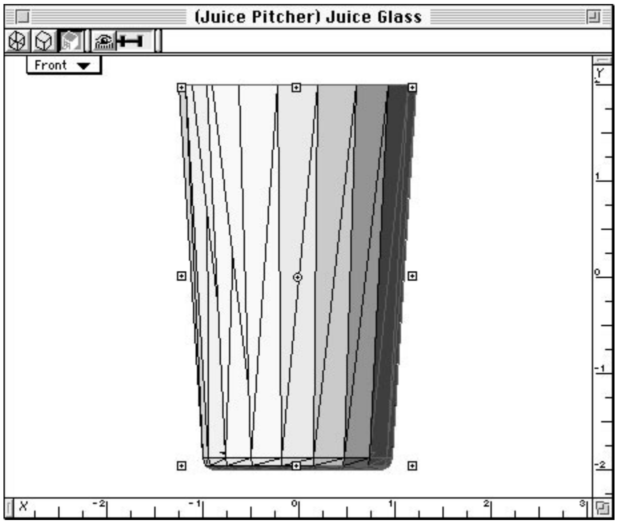
Lathe Degrees : 360.0 °

Lathe Segments : 18

☒ Create two-sided objects

Next, enter the lathe settings on the right side of the Lathe dialog. For this project, set them to a 360° lathe with 18 segments.

14. **Click on the OK button.** The Lathe modeler displays a progress bar as the lathe operation is performed. A fully-formed 3-D object is being constructed!



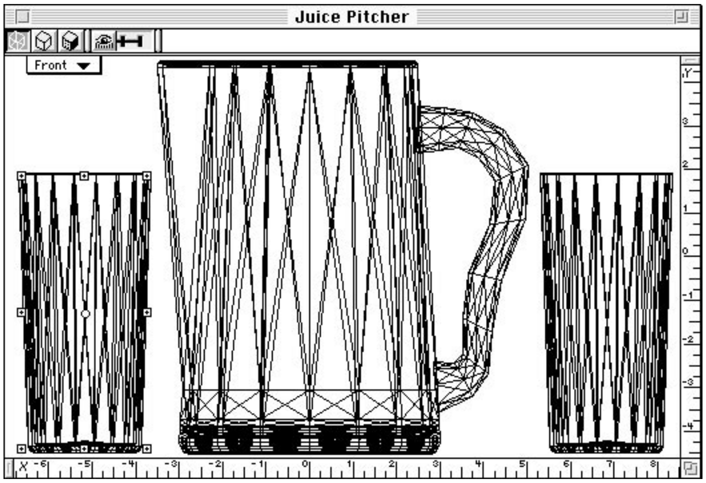
When it is finished, the lathed glass is drawn in the shape workspace's window.

15. **Close the shape window.** This is done by clicking on the window's close box. This makes the main model window the active window again. Even though the Juice Glass shape window is closed, that shape workspace is still defined in this model.

    This would be a good time to save. If this is the first time you have saved this model, you can use either the Save or Save As command to name the model and save it. Note that the Juice Pitcher and Juice Glass shapes are saved with the model, even though only the Juice Pitcher is showing in the model window.

16. **Insert the Juice Glass shape into the model as you did before with the pitcher.** Pull down the Shapes menu and select "Juice Glass" from the Insert Shape sub-menu. With the cursor loaded, click once anywhere in the model window. An instance of the Juice Glass shape is inserted into the model window.

17. **Copy the Juice Glass instance.** This is done by using the Option-drag feature. An Option-drag leaves the original object where it is at, and drags a copy to the new location.



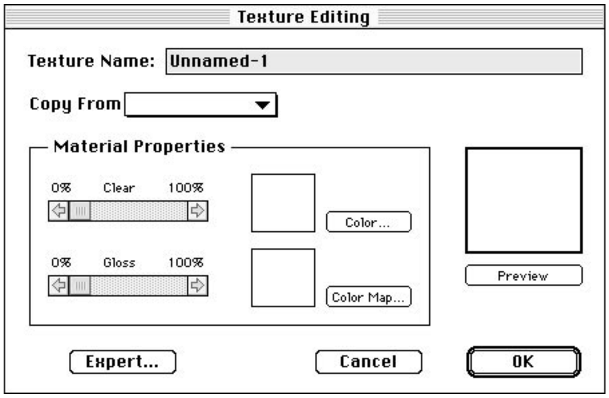
Make sure you're still in the Front view of the model window with the Juice Glass instance selected. Choose the Pointer tool from the Tool palette. Then, while holding down the Option key, click on the Juice Glass instance and drag in any direction. As you drag, a bounding box will move with the cursor. When you release the mouse button, a second Juice Glass instance is drawn in the model window.

Applying a texture

18. **Open the Juice Glass shape window again.** Pull down the Shapes menu and choose “Juice Glass” from the Edit Shape sub-menu. Vision3d opens a window into the Juice Glass shape workspace. Make sure the object in that window is selected.

    NOTE: *The juice glass is referred to as a shape object here, rather than a shape instance, because at this level it is the actual object. This is an important distinction.*

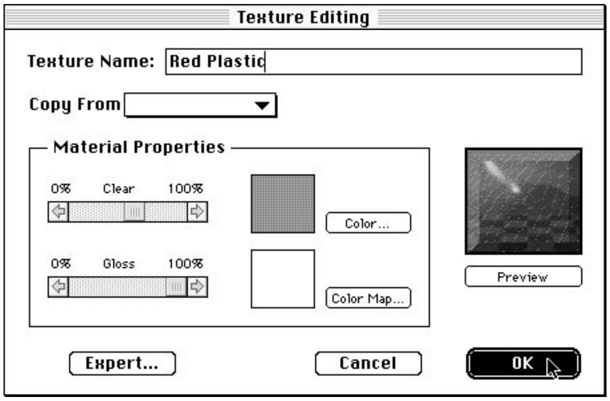
19. **Create a new texture.** Pull down the Textures menu and choose the New Texture command. The Texture Editing dialog box opens.



Texture Name: Unnamed-1

    This is where you define surface properties. This is contained in a texture, which can then applied to one or more objects. The creation and application of textures are presented in *Chapter 5: Using Textures*.

20. **Enter the texture parameters.** First, type “Red Plastic” in the texture name field. Second, click in the Color button. The Macintosh color picker appears. Choose a red color of your liking.



Texture Name: Red Plastic

Finally, match the gloss and clear slider settings shown in the illustration on the previous page

21. **Click on the OK button.** Vision3d creates the Red Plastic texture and applies it to the object. Notice the glass shape changes color in the shape window to show that the texture is applied.

    You can close the shape window, as before.

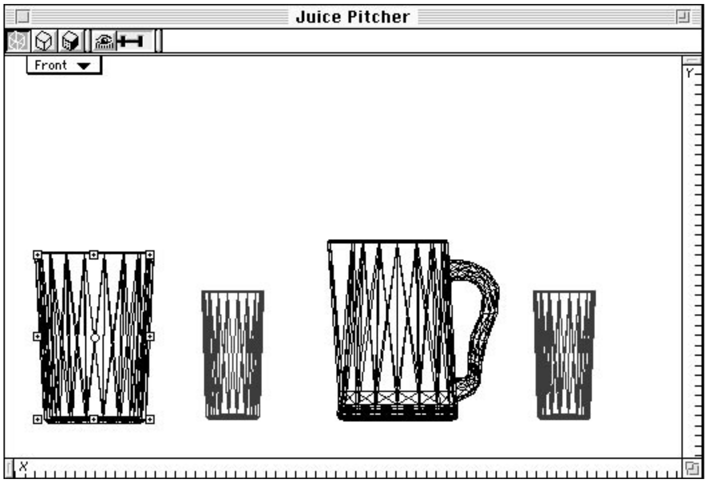
22. **Bring the model window to the front so that it is the active window.** Notice how all instances of the juice glass in the model window now have the Juice Glass texture applied, even though the texture was applied AFTER the shape instances were inserted.

    This should give you a hint of the power of shape hierarchy within Vision3d. When you make changes within a shape workspace, those changes carry through to all instances of that shape in all levels of the model.

23. **Insert a third juice glass.** Pull down the Shapes menu and choose "Juice Glass" from the Insert Shape sub-menu. Again, you will see the loaded cursor in the model window.

    Of course, you could just use the Option-drag process again to insert another instance of the juice glass. But it's being done differently to demonstrate more features.

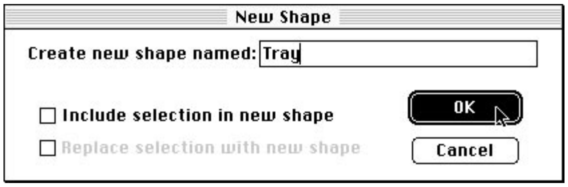
24. **Insert another Juice Glass instance.** With the loaded cursor visible in the model window, hold down the Shift key, then click and drag a marquee approximately double the size of the two Juice Glass instances now in the model. Vision3d draws a new instance of the Juice Glass shape, but scaled to the size of the marquee drawn.



Holding down the Shift key constrains proportions of the shape as it's scaled. From this process, you can see there are two ways to insert a shape instance into a model. The single-click method inserts the instance at the same size as the original shape. The click-and-drag method inserts the instance at a scale you define.

NOTE: *The scale change for the third juice glass only applies to this instance, at this level. The original object is not altered in any way, nor the previous two glasses.*

25. **Create another new shape.** As you've done previously in this exercise, choose the New Shape command.



New Shape

Create new shape named: Tray

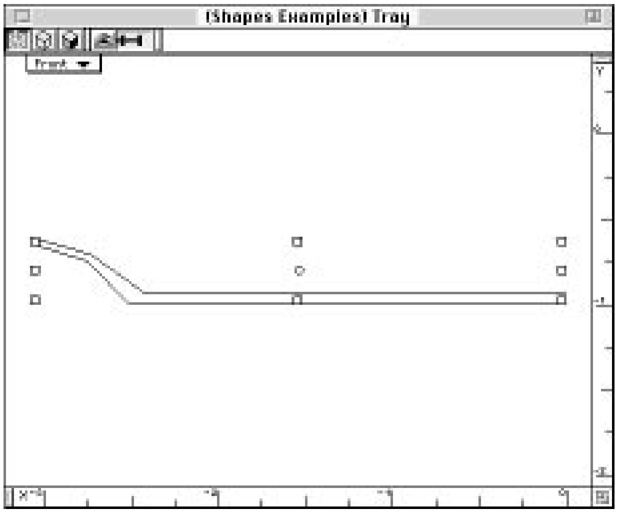
☐ Include selection in new shape

☐ Replace selection with new shape

When the dialog box appears, name the new shape, "Tray." Make sure both check boxes are empty.

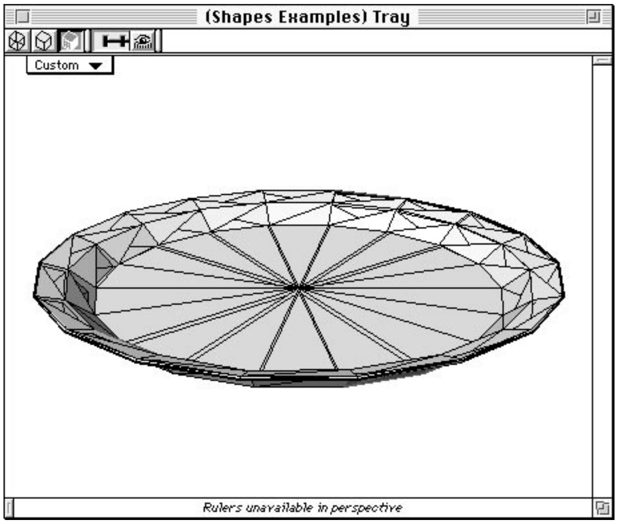
26. **Choose the open polyline tool from the tool palette.** Just as before, you will draw a cross section…this time for a tray.

27. **Draw a tray cross-section as shown.** As with the glasses, you can draw a 2-D template that gives some thickness to the geometry when used to construct the 3-D object. Remember, you are drawing just one-half of the tray profile.



Draw a template similar to the one shown in the illustration on the previous page. The exact geometry isn't critical. The width isn't important either, since you can always scale the instance larger when you insert it into the model.

28. **Lathe the tray template.** This is exactly the same procedure used to make the glasses. Notice that the settings you entered into the Lathe modeler for the juice glass are still there. Since you are after essentially the same result, but with a different template, you can leave the settings the same, i.e. — 360° lathe, 18 segments. Also make sure you position the lathe axis at the appropriate spot on the template.
29. **Click on the OK button.** You may want to check the tray's geometry by looking at it from both Front and Top views.

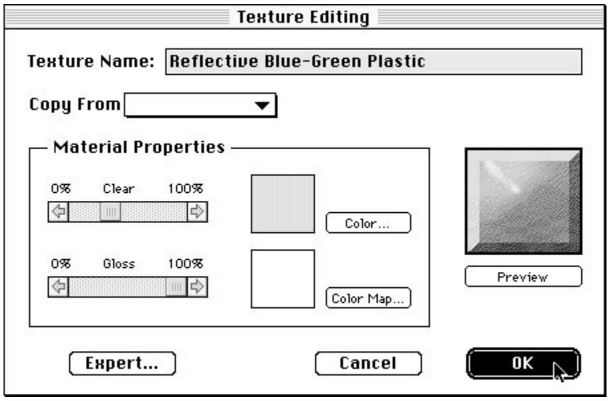


Perhaps you will want to turn on perspective and rotate the view upward to see it in a more natural appearance. If you find that the edge profile of the tray is too faceted, you can increase the number of segments used to lathe it. This is done by double-clicking on the tray to open the Lathe modeler again. Make the change in the segments field and click on the OK button again.

Alternately, you can select the tray and choose the Re-Lathe command from the modeling menu to open the Lathe modeler.

NOTE: *While you can look at the object from different views, you should avoid rotating the object. You don't want to change the object's orientation, because that will also affect how its instance is inserted into the model.*

30. **With the Tray shape selected, create and apply a new texture as you did to the juice glass earlier in this exercise.** Name the new texture “Reflective Blue-Green Plastic.”

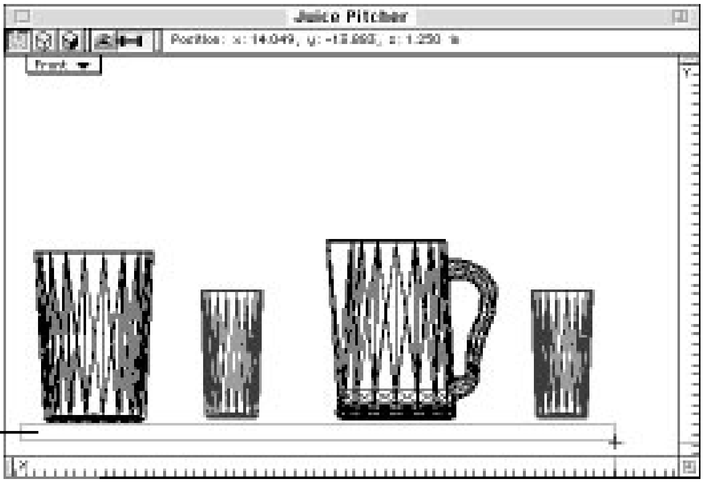


Texture Name: Reflective Blue-Green Plastic

Copy From

Choose a dark blue or blue-green color. Match the gloss and clear settings shown in the above illustration.

31. **Insert the Tray shape into the model using the click-and-drag method.** Hold down the Shift key to constrain the instance’s proportions as you click and drag.



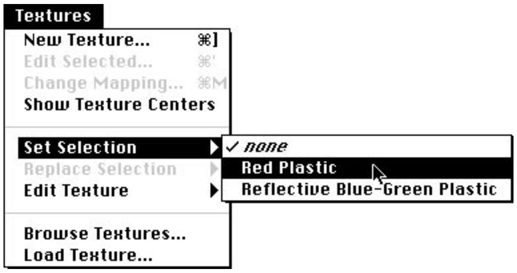
Insert the tray instance using the click-drag method

Try to scale the tray instance so it will match the scale of the other objects when you reposition the pitcher and glasses to sit on the tray. You will be able to re-scale the tray instance later, if needed.

32. **Edit the Juice Pitcher shape.** This is done by choosing the Juice Pitcher from the Edit Shape sub-menu. Alternately, you can locate the juice

pitcher's preview on the Shapes palette, select it, and click on the Edit button. Either method opens a window into juice pitcher shape.

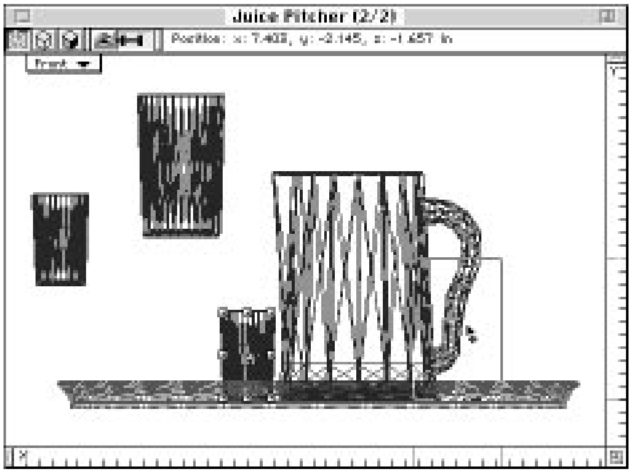
In the Juice Pitcher window, with the juice pitcher selected apply the juice glass texture to the pitcher object.



With the juice pitcher selected, pull down the Textures menu and use the Set Selection sub-menu to choose the Red Plastic texture. Alternately, you can locate the Red Plastic texture on the Textures palette, select it, and click on the apply button.

The texture is applied to the juice pitcher at the shape level, which of course is immediately disseminated to its instance at the model level. After applying the texture, you can close the juice pitcher's shape window again.

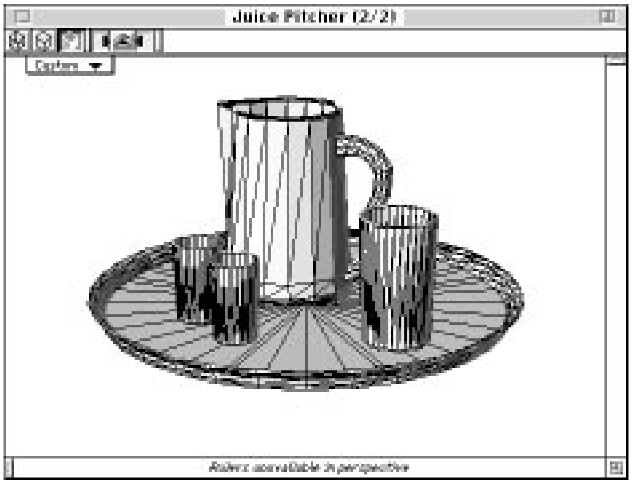
33. Align the pitcher and juice glass instances on the tray instance. With the model window the active window, make sure the lens control is set to Orthographic. Also make sure the Pointer tool is chosen from the Tool palette.



In the Front view, reposition the pitcher and glasses so they sit on the tray. Then switch to the Top view and again reposition the instances so they are on the tray and not overlapping. Finally, change to the Left or Right side view and repeat the repositioning process. By looking at all three views, you can accurately reposition objects in 3-D space.

If the tray instance is not big enough to hold the glasses and pitcher, choose the Object Scale tool from the Tool palette and select the tray. Then, holding down the Shift and Option keys (to constrain proportions and scale from the center), click on one of the corner selection handles and drag to scale the tray larger.

34. **Change the three-position lens control to Normal and the view to 18° degrees up from true Front.** This gives you a perspective look at the tray, pitcher and glasses.

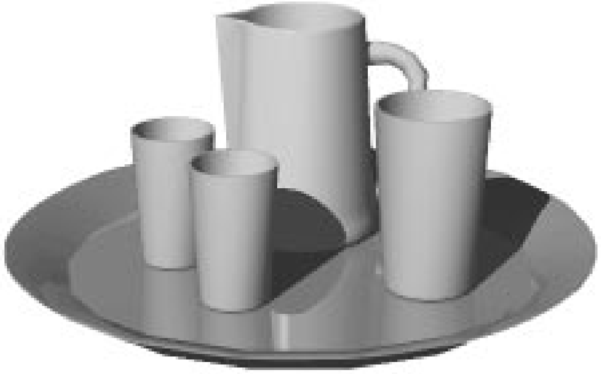


If the pitcher and glasses need additional repositioning, use the method described in the previous step. If necessary, use the "Fit View to All" command to fit the assembled model to the new view.

35. **If desired, render your model.** Choose the Camera tool from the Tool palette. Hold down the shift key and click once in the model window. Vision3d opens the Render dialog box.

Set the Renderer Selection field to "Phong Shading," set the Texture Detail to "Fine," and set Image Quality is set on "Smooth."

Clicking on the OK button begins a rendering of the model. Please refer to *Chapter 8: Rendering* for a presentation of the options when rendering images.



Summary

In this Quick Tutorial you have learned about Shapes, their hierarchy and how to create, edit and insert them into a model. Using what you've learned here about shapes will greatly simplify creating complex models with Vision3d.

Handling shapes in a model

Shapes created or loaded into a model become part of that model. Saving the model also saves its shapes as part of the document. This is true whether or not a shape has any instantiations in the model space. Shape workspaces are kept exactly as they were when the shape window was closed…including the window's size, view arrangements, display method, etc.

Creating new shapes

As discussed at the beginning of this chapter, any number of shapes can be created as part of a model. This is done by choosing New Shape from the Shapes menu, which opens a window into that workspace. Even if you put nothing in that shape and close the window, the shape workspace is still defined and remains part of the model, unless you delete it. A full explanation of this command is found later in this chapter.

Loading shapes

Shapes may be loaded into a model as needed. It is an easy way to transfer modeling work done previously in other projects to the current project. Once you have acquired a repertoire of shape files, either from commercial libraries or your own work, this feature allows you to assemble pre-built building blocks into a complex model very quickly

A distinction should be made between loading a shape and inserting a shape. When a shape is loaded, it is given a workspace in the model, but no instances of the shape are inserted into another workspace. Inserting a shape is the act of putting an instance of that shape into another workspace. As mentioned at the beginning of this chapter (and reiterated here for emphasis), the active view into which the instance is inserted determines the orientation of the instance. If the orientation is not as desired, you can either rotate the instance or delete it and insert the shape again in a different view orientation.

Many ways to load

There are six ways to load a shape into an open model:

1. **Double-clicking on its preview image in the Shapes palette.** This loads the shape only. No insertion or editing of the shape is attempted.
2. **Selecting a shape on the palette and clicking on the Insert button.** This loads and inserts the shape into the active view.
3. **Selecting a shape on the palette and clicking on the Edit button.** This loads the shape and opens its window for editing. No insertion is attempted.
4. **Clicking and dragging the preview image from the Shapes palette to the active view.** This loads and inserts the shape into the active view.
5. **Choosing the Load Shape command in the Shapes menu.** This loads the shape only. It is similar to method #1, above.
6. **Choosing the Browse Shapes command and clicking on the Load button in its dialog.** This is a another variation on methods #1 and #5, above.

The method you choose depends on what you want to do with that shape. The first four methods assume the shape to be loaded is represented on the Shapes palette. The last two methods are for shapes that are located somewhere other than in the Shapes folder (within the Vision3d application folder).

The loading of shapes contained on the Shapes palette is explained under the Shapes palette discussion later in this chapter.

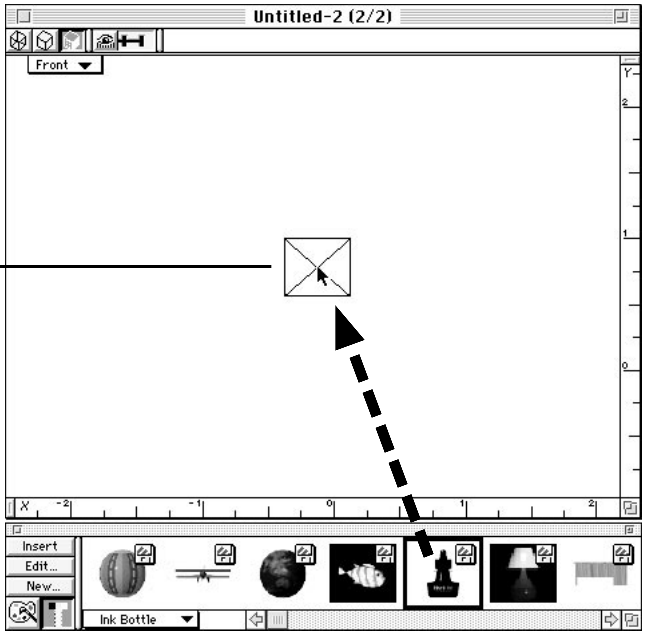
As mentioned, you can also load shapes in the active model from any source through the Load Shape command in the Shapes menu. This feature is explained further under a discussion of the Load Shape command later in this chapter. In addition, the Browse Shapes dialog box also contains a Load button. It functions exactly the same as the Load Shape command.

"Drag and drop" explained

As mentioned in method #4, you can click and drag the shape's preview image to a modeling window in order to load and insert the shape. This is referred to as "drag & drop." This feature is available with any operating system. It is a feature of the Vision3d application, and does not rely on System 7 for support.

A small marquee appears representing the shape as it is being dragged. The shape is loaded into the model and inserted in the active view at the position where you drop the marquee.

Shapes palette "Drag & Drop"



Click and drag the chosen shape into the active view

The shape is automatically loaded into the active model and inserted at the point where you drop the bounding box.

The marquee you see during a drag and drop operation does NOT represent the size of the shape being inserted. The shape's instance is inserted at its actual scale, regardless of the current scale of the view into which it is inserted.

Adjusting instance scale during insertion

Since Vision3d places no constraint on the scale at which a model is constructed, it is inevitable that you will experience the problem of loading shapes that are not the same scale as the model. Of course, it is always a good idea to know the size of the objects you are constructing. But using the same scale for all your work is not always practical. Some of the shapes you may want to use will be built by someone else with a different philosophy of size. Or, you own work may vary in scale depending on the project.

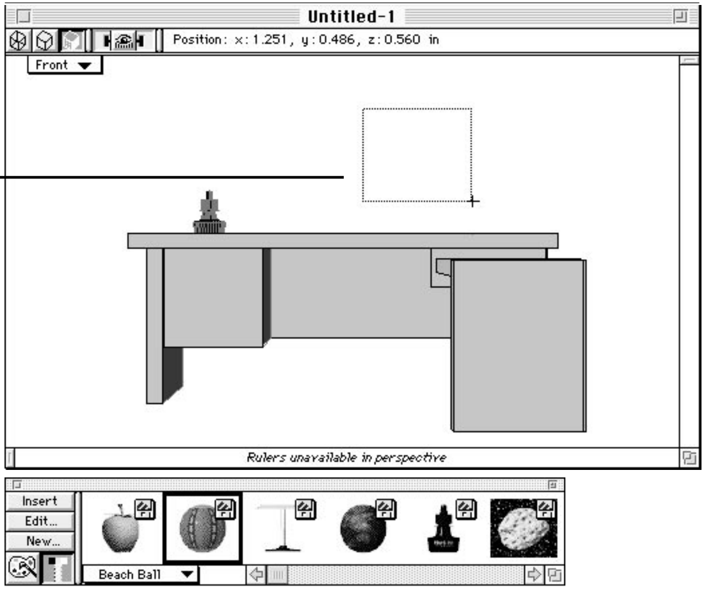
For example, you might build a chair or an appliance at actual size because it is usually easier to understand how big you need to make the root shapes and the texture maps, then everything assembles nice and easy. But the house in which you would place them would not be pragmatic at actual size. Although Vision3d could handle the geometry if you wanted to model a house at actual size, the texture maps for the walls and other details would be way too big to load, no matter how much memory you had. The Macintosh could not handle one image file that big, let alone multiple maps. So you would probably build the house at a scale of one inch to one foot. Then the maps can be at a more manageable size.

So, what do you do with the chairs and appliances that you bring into the model at a scale 12X bigger than the house? Hey! No problem… If you need to adjust the scale of a shape, of course you can edit the shape and change the size of the original object there. For simple objects, this is a viable option. But if that object is assembled from other shapes, this method could become overly difficult…going into each shape and adjusting the size, then putting everything back together. There is a better way.

Using the click-drag method to insert a shape

The feedback area identifies the x, y, z coordinates at which the shape is inserted

As you click-and-drag, the marquee indicates the size at which the shape is inserted in the model



Position: x: 1.251, y: 0.486, z: 0.560 in

At any point in the assembly process, a shape instance's scale can be changed, either bigger or smaller. You can scale the instance at the time it is inserted or after insertion. Of course, each instance can be a different scale, even of the same shape. And the good news is that everything about the shape is scaled with the instance, including geometry and texture maps.

To scale during insertion, use the click-drag method. This means you would need to use the Insert Shape command, which allows you to use the cursor for inserting the object. You would drag a marquee indicating the size of the instance, then release the mouse button to draw the shape instance in the active view. This method requires that the shape is already loaded into the model.

To scale an instance after insertion, use the Scale tool from the Tool palette or the Transform command. You can insert a shape instance using one of the methods that insert automatically — such as the Insert button or Drag & Drop. If you use the Insert Shape command, you would perform the insertion with a single click, without dragging any marquee.

Note that changing the size of an instance does not affect the original object in the shape workspace in any way. The scale transformation is applied only to the instance at the higher level.

Deleting shapes

A shape can only be eliminated or deleted from a model through the Browse Shapes command. The Shapes Browser dialog allows you to select one or more shapes and remove them with the Delete button. Deleting a shape cannot be undone. So, whenever this operation is performed, an alert appears asking if you are sure you want to remove the selected shapes. If you choose to continue, the selected shape is removed immediately from the open model. If the model was saved prior to deleting a shape, note that the saved document retains that shape until it is replaced using the Save or Save As commands.

Saving shapes

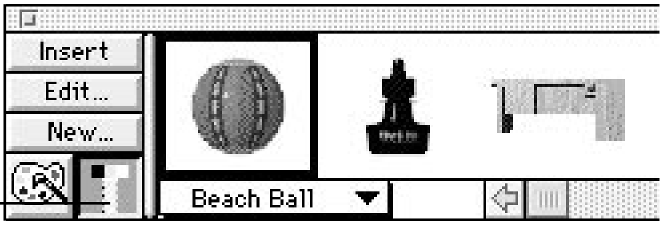
Shapes can be saved as separate files. This is done by choosing the Browse Shapes command, then selecting one or more shapes listed in the dialog and clicking on the Save button.

The name given when the new shape was created is used as the file name for the saved shape. In addition, a preview image is automatically generated for each shape file as part of the save operation. This is the image used on the Shapes palette or when using the Load Shapes command. If you wish this preview image to be anti-aliased for a better impression, hold down the Option key while clicking on the Save button.

Shapes Palette

The Shapes and Textures palettes share the same space on the screen. Clicking on the Shape button in the bottom left of this palette switches it to the Shapes palette; clicking on the Textures button switches it to the Textures palette.

Since so many of the issues involving shapes and textures are similar, you will find many aspects of Vision3d's interface share this parallel or dual-purpose functionality for shapes and textures. Note also how the Shapes and Textures menus contain similar commands.



Clicking on the Shapes button on the Textures/Shapes palette displays the available shapes

The Shapes palette contains a preview image and a listing in the pop-up menu for each shape available in the Shapes folder. At the beginning of a session the Shapes folder is scanned and those are the only shapes displayed on the Shapes palette. As you define new shapes or load shapes from other locations into the active model, they also appear on the Shapes palette.

When a new shape is created in a model, it immediately appears on the Shapes palette and can also have an image generated for the palette to help visually identify it. Shape files loaded from sources other than the Shapes folder bring their own preview that automatically appears on the Shapes palette.

In addition, those files imported into a model (which automatically creates a new shape workspace to contain the imported objects) are given a space on the palette, even if it does not have a preview image for the palette. A preview is automatically created for imported shapes.

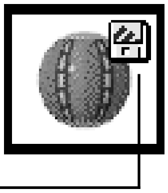
If more than one model is open, switching active models causes the Shapes palette to redraw. This is necessary so that it can accurately show the currently loaded shapes for the active model.

Operation

Representations of all shapes available to the current model are presented on the Shapes palette. The available shapes include those created in the model itself, those loaded into the model or those found in the

Shapes folder on the hard-disk drive upon startup of the application (but not loaded). A preview of these shapes may be viewed by scrolling their images horizontally in the palette window.

When scrolling left to right on the Shapes palette, the shapes created in or currently loaded into the active model always appear first in the palette's image preview area. Shapes found in the Shapes folder, but not loaded in the current model are identified by a floppy disk icon super-imposed on its preview image. Loaded shapes do not have the floppy disk icon.



A floppy disk icon super-imposed over a shape preview means the shape is not loaded in the model

To select a shape on the palette, click once on its preview image. Only one shape can be selected at a time on the palette. A black rectangle appears around the preview. Alternately, you can use the pop-up menu to select a shape by name. Choosing a shape from the menu automatically scrolls the palette to show that shape's image and moves the selection rectangle to that shape.

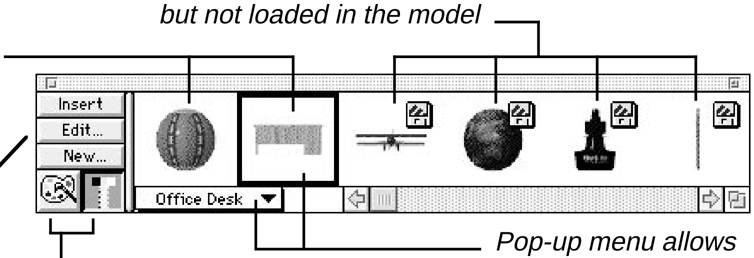
To load a shape from the palette, double-click on its preview image. The floppy disk icon is then removed from that image, indicating that it is now loaded in the model. Note that this process only loads the shape, but does not insert an instance of it into the model.

To insert a shape from the palette, select its preview on the palette and click on the Insert button. The shape will be loaded into the model and the cursor changes to the insert mode when positioned over the active view. The active view determines the insertion orientation. As explained earlier in this chapter, the active view can be for the model space or another shape's workspace. Clicking the cursor in the view inserts the shape.

If the shape is already loaded, clicking on the Insert button still performs the insert operation. Additionally, with already-loaded shapes, you can use the Insert Shape command in the Shapes menu.

Controls on the palette

Controls found on the Shapes palette allow you to manage the shapes represented on the palette. Note that if more than one model is open, as with other aspects of the interface these controls have effect only in the active model.



These shapes are loaded in the model

These shapes are in the Shapes folder, but not loaded in the model

Use these buttons to create, adjust, & insert shapes

Use these buttons to toggle between Textures and Shapes

Pop-up menu allows selecting shapes on palette by name

The Shapes palette controls have been alluded to previously in this chapter in explanations of certain operations. They are presented here item-by-item for reference:

- **Title bar.** Use this control to position the palette on the desktop. Vision3d remembers the position of the palette between sessions, and when showing the palette after it has been hidden.
- **Close box.** Click on this control to hide the Shapes palette. This is the same as choosing the Hide Shapes palette from the Palettes submenu in the Window menu, or clicking on the bottom right Hide/Show Palettes button on the Button bar.
- **Zoom box.** Click on this control to expand or contract the palette horizontally. Zooming out automatically expands the palette to the full width of the monitor on which it is displayed. Zoom in returns the palette's width to the size set by the re-size box. The palette does not expand vertically, even if there are more shapes than can be displayed on the palette at full width.
- **Re-size box.** Use this control to set the width of the palette on the screen. You can drag the re-size control horizontally. The widest setting possible is the full width of the desktop (even across multiple monitors, if available).

  Note that the width you drag with this control "snaps to" the closest whole preview. You cannot set the palette's width to show only a partial preview image of a shape.
- **Scroll bar.** Use this control to scroll horizontally in the palette to view all the shapes represented there. The scroll bar is active only when there are more shapes than can be viewed at one time on the palette.
- **Shapes preview area.** This area of the Shapes palette contains a preview image for each shape available in the active model.

Whether created in the model, loaded into the model, or still in the Shapes folder on the hard-disk drive, each shape is given a place on the palette.

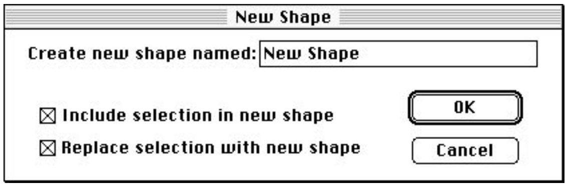
- **Insert button.** Click on this button to insert a selected shape into the active model. The shape is loaded into the model, then an instance of the shape is automatically inserted in the active view at the view set center. The instance's size is the same as the shape.

  Note that this behavior is different than the Insert Shape command in the Shapes menu, which gives you a cursor to position the instance in the view and adjust its scale during insertion, if desired.

- **Edit button.** Click on this button to edit the shape selected on the Shapes palette. A window is opened into that shape's workspace. If the shape chosen to be edited is not loaded into the model, it will be automatically loaded first. If the shape selected is already loaded, it is not re-loaded, but simply opens the shape's window.

  For those shapes that are loaded in the active model, you can also use the Edit Shape command in the Shapes menu to perform the same operation.

- **New button.** Click on this button to define a new shape. The new shape is created in the active model. A New Shape dialog box is presented in which you may enter the name of the shape and indicate whether you want to include the currently-selected objects (or instances of other shapes) in the new shape's workspace.



New Shape

Create new shape named: New Shape

☒ Include selection in new shape

☒ Replace selection with new shape

  If you choose to include the selected objects in the new shape, you also have the option of replacing them with instances of these objects in the active workspace. If no objects are selected when the New Shape dialog box opens, these checkbox items are dimmed.

  The New button performs the same task as choosing the New Shape command in the Shapes menu.

- **File size field.** This field indicates the size in Kbytes of the selected shape prior to loading. This is the size of the shape file as it is saved. The shape may actually take more space once loaded into the model, especially if any texture maps are used in the shape.

- **List of Shapes (pop-up menu).** Use this menu to choose a shape on the palette by name. The menu lists any shapes that are loaded in the active model first, with a divider between the loaded shapes and those not loaded, but available in the Shapes folder. Within each group, the listing is alphabetical.
- **Loaded/On-disk indicator (floppy icon).** The presence of this icon indicates that a shape is not loaded in the active model. It is removed from the preview images of shapes loaded in the model.

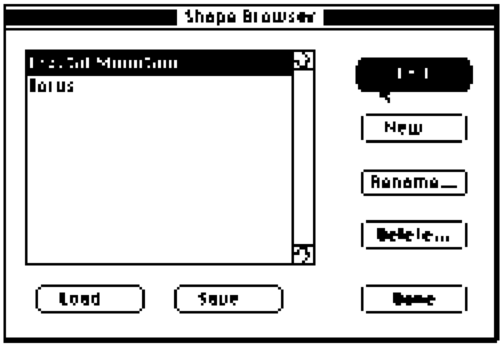
Commands used with Shapes and shape instances

The following commands are considered relevant to the handling of shapes as part of documents you create the software. A detailed explanation is included under each command in the listing.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

Browse Shapes command (Shapes menu)

The shapes browser is a specialized dialog box that provides several shape-handling functions in one location. The Browse Shapes command is available when at least one shape is defined in the current model.



When the shapes browser appears, a list of the shapes loaded in the current model is displayed for selection. One or more shapes may be selected from the list. Once a selection is made, the available operations that may be performed are indicated by the active buttons in the dialog.

Some of the functions provided in the browser dialog are duplicates of other Shapes menu commands or buttons on the Shapes palette. They are also provided here for convenience. However, some operations are available only through the browser dialog box, such as saving a shape as a library file or deleting a shape workspace from the model.

- **Edit:.** Clicking this button opens a window displaying a selected shape for editing purposes. Using this button also closes the Shape Browser dialog box.
- **New.** Clicking this button creates a new shape workspace. The New Shape dialog opens to provide the same operation as the New Shape command. After clicking on the OK button in the New Shape dialog, you are returned to the Shapes Browser dialog.
- **Rename.** Clicking this button presents a dialog box to change the name of a selected shape.
- **Delete.** Clicking this button deletes the selected shape(s). Since this action is not reversible through the Undo command, an alert is displayed warning of the consequences before any shape is deleted, along with a Cancel button to provide a graceful exit without taking this action.
- **Load.** Clicking this button loads a shape into the active model. A directory dialog opens pointing at the Shapes folder. You can select one of the shapes listed there or use the controls in the dialog to locate a file in another folder. This button provides the same function as the Load Shape command in the Shapes menu.

  NOTE: *The Load Shape directory dialog only recognizes files with the "VisP" file type. No other files appear in the directory of the dialog, even when present in the folder in which you are looking. "VisP" is Vision3d's specific file type given to shape files when saved.*
- **Save.** Clicking this button saves the selected shape(s) in a file format used for the shapes library. This file has the "VisP" file type designation and is recognized by the Load Shape dialog. During the save, Vision3d automatically renders and includes a preview image for use in the Load Shape dialog.

  To render an anti-aliased preview image, hold down the Option key while clicking on the Save button.
- **Done.** This button closes the browser dialog box and returns to the model.

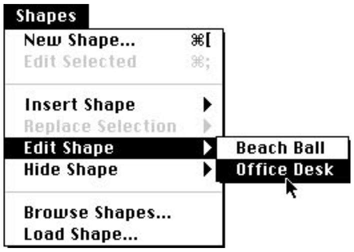
Edit Selected command (Shapes menu, ⌘; [semicolon])

The Edit Selected command opens a window for the shape of the instance selected in the active window. Edit Selected is available only when an instantiation of a shape is selected. If the object selected is not tied to a shape, the command is dimmed.

If more than one shape is selected, a modeling window for each shape is opened for editing.

Edit Shape command (Shapes menu)

Use this command to open a window into a shape workspace for the purpose of editing the object located there. This command is available when at least one shape workspace has been defined in the active model. If no shapes have been created or loaded, the command is dimmed.

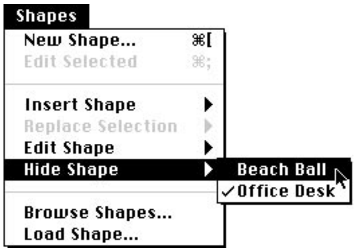


A sub-menu lists the available shapes. Only those shapes currently loaded into the active model are available for editing through this command. However, using the Shapes palette, you can select a shape that is not loaded and click on the Edit button. This automatically loads the shape and opens a window into its workspace.

Hide Shape command (Shapes menu)

Use this command to hide a shape. This command is available when at least one shape workspace has been defined in the active model. If no shapes have been created or loaded, the command is dimmed.

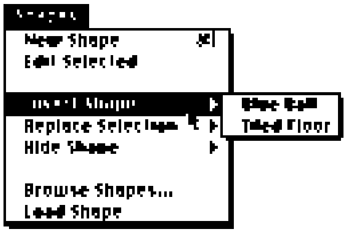
A sub-menu lists the shapes currently loaded into the active model. Choosing a shape from this list hides all instances of the shape throughout the entire model, at all levels.



Hidden shapes are indicated by a check mark next to the shape's name in the Hide Shapes sub-menu. Selecting a previously-hidden shape in the sub-menu toggles its show/hide status and un-hides it.

Insert Shape command (Shapes menu)

The Insert Shape command allows you to choose a shape by name from a sub-menu, then insert it into the current model. Insert Shape is available when at least one shape is defined in the model, either created directly or loaded from the library.



The desired shape is selected for insertion by highlighting its name in the sub-menu before releasing the mouse button. The cursor is then “loaded” with the shape for insertion into the active view of the model. Once the cursor is ready, the shape may be inserted by one of two ways:

- **By simply clicking once in the active view**
- **By dragging the cursor as it is inserted**

The single-click method inserts the shape at the same size it was created. The drag method re-sizes the shape relative to the scale of the active view.

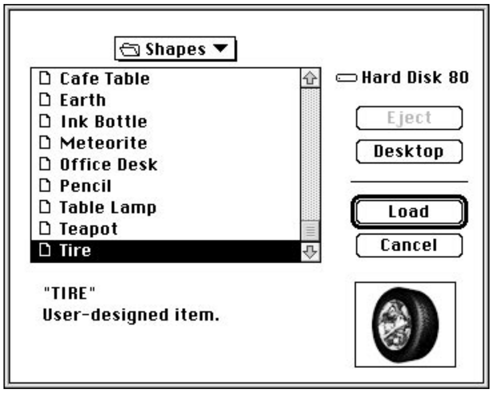
Each insertion of a shape is treated as an instance of the shape, not a separate object. Changes made to a shape in its work space are reflected in all instances of that shape in the model. As with any other object, the instantiation may be moved, rotated and scaled in the model using the object manipulation tools.

Ungrouping a shape instantiation at a different level will break its tie to the shape and cause it to be treated as a separate, regular un-named object.

Load Shape command (Shapes menu)

The Load Shape command reads a shape file from the disk into the current model and includes it in the shapes list. This command is available whenever a model is open.

Note that the directory dialog that opens when this command is called only shows shape files, using the “VisP” file type. You will not be able to load any other types of files through this command.



"TIRE"
User-designed item.

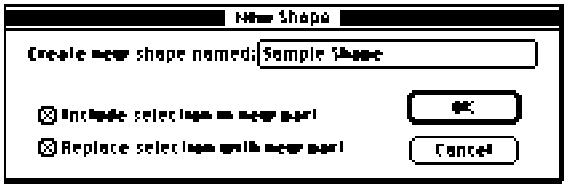
Vision3d initially looks for shapes in a disk-resident "Shapes" folder. The location of this folder must be within the Vision3d application folder. If you wish to load a shape not located in that folder, you can use the controls to look for other shape files in another folder.

New Shape command (Shapes menu, ⌘[)

The New Shape command initiates the process of creating a new shape in the active model. This command is available whenever a model is open.

Shapes are objects drawn or constructed in a separate workspace where they may be modeled and edited. Other shapes may be used in the construction of a new shape. A shape may even be inserted into its own workspace to create a recursion effect.

This command displays a dialog in which an object can be named and added to the shapes list. The dialog creates a work space for the new shape and allows access for modeling the shape's geometry and texture(s).



Create new shape named: Sample Shape

☒ Include selection in new part

☒ Replace selection with new part

Clicking on the OK button opens a new window into the shape's workspace, which is separate from the model space, but part of the same model document. Here one or more objects may be created and manipulated using the tools available on the tool palette. A shape's window looks the same as a model window. The only distinguishable difference is the name in the title bar. As with the model, more than one window may be open for the shape and more than one shape may be open at one time. The amount of memory available to the application is the only limitation to the number of windows open for shape workspaces.

Creating a completely new shape

The shape creation process may be as simple as drawing a single object. Or it may include using previously-created shapes, pasting an object from the scrapbook or importing 3-D data in another format from disk.

If the shape being created is completely new, no texture is applied to the shape. Textures can be applied using the texture-editing commands in the Textures menu or buttons on the Textures palette. Textures applied in the shape work space are included in all instances of the shape throughout the model at whatever level they may be inserted.

If the shape uses other shapes or contains pasted or imported objects as part of its creation, those items may bring their own textures or surface definitions with them. External files with surface definitions may be a simple color designation, such as those present in a DXF file.

New from Selection

This command allows a previously-drawn object in the current model to be designated as a shape. The essential difference between regular objects and shapes is that shapes are named objects. Because they are uniquely identified, Vision3d can provide special ways of handling them by name.

If no object is selected in the model when calling this command, the New from Selection option is not presented. The software assumes you wish to create a completely new shape.

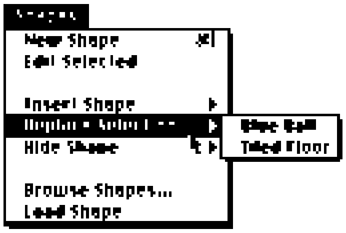
If an object (or group of objects) is currently selected when this command is called, a dialog is displayed to assign a shape name and add it to the shapes list. The object — now a shape — is then displayed in its work space.

You also have the option of making the original object (whose geometry was used for the new shape) become an instantiation of the shape. Converting the object to a shape automatically replaces the object with an iteration of the shape, establishing a direct connection between the original object and the shape for editing purposes and memory efficiency.

If you choose not to replace the original object with an instantiation, no link will be established with the new shape. The object remains a completely independent object.

Replace Selected command (Shapes menu)

The Replace Selection command replaces the selected object in the model with an instance of the designated shape. Replace Selection is available when a selection exists in the model AND at least one shape has been defined in the model. The selection may be a regular object or another shape's instantiation. If more than one object is selected, the shape chosen will replace each one separately.



The replacement shape is given exactly the same transformation parameters and animation scripting as the replaced item.

Chapter 5: Using Textures

Three-dimensional computer modeling deals not only with the geometry of objects, but also with the properties of their surfaces and how those surfaces interact with light and each other. The description and treatment of surfaces during rendering is at least as complex as modeling the objects themselves.

Because Vision3d offers extensive surface-description capabilities, the most efficient approach to enhancing realism in rendered images is through textures. Which leads to the truism: ***"A good surface map is worth a thousand polygons."***

That means, whenever possible you should rely on surface maps rather than creating explicit geometry to achieve detail. The resulting models are often more memory efficient and faster to render.

Textures are applied to objects to define their appearance when rendered. Several objects may use one texture, but only one texture can be applied to an object. The texture can be customized to each object's requirements with the "Change Mapping" command in the Textures menu. When a texture is modified through its editing dialog boxes, all instances of that texture are automatically adjusted wherever it is used throughout the model.

Mastering the use of textures in Vision3d, especially the advanced surface mapping features, is not a trivial matter. Properly executed, surface mapping enhances the realism of renderings such that they can become indistinguishable from an actual photograph. While this may seem a bold claim to make, nonetheless it is true. Many users vote their agreement again and again by choosing Vision3d to render contest-winning images throughout the graphic arts industry.

If you invest the necessary time to learn Vision3d's finer points of texture handling, advanced mapping techniques, and multi-property composition, the results can be very rewarding. It will not come for free, it will take time to learn and to do correctly. But nothing worth doing ever was easy. …If it was easy, everyone could do it, and you couldn't make any money at it!

This chapter contains the foundational and practical information upon which you can build your texture-creation skills. Once you begin applying the concepts, you will find that experimentation is really your best teacher, i.e — you learn by doing. Trying out your own ideas just to “see what happens” can also be a rewarding approach.

Vision3d supports both Solid (volumetric) and Surface (mapping) textures. The steps to create either type of texture are similar, but the way they work on objects is quite different. …And that difference is significant.

Solid textures offer greater technical precision and are computationally more efficient. Further, they are infinite in resolution and size. No matter how close you approach the surface, the pattern will not pixelize. No matter how big the object, a solid texture will cover its surface completely. On the other hand, solid textures can sometimes appear artificial, even cartoon-like, because they are mathematically derived. It takes a practiced eye to set up a solid texture’s parameters for a realistic feel.

Creating surface textures can be as simple as scanning a color picture, loading it into a texture, and applying it to an object. In more advanced functions, surface mapping offers the ability to control multiple surface properties at every point on the face of an object. This is done with specialized PICT images.

Properly used, surface textures can look quite real because the surface maps used are often digitized from actual objects. These surface maps are readily available to the average user and can be easily prepared with available software, such as paint programs with image processing tools.

Either type of texture has its strengths and weaknesses. These will be identified in this chapter.

Making and editing textures

Textures may be created in the active model at any time, as needed. Previously-built textures may also be loaded from disk. Once a texture is defined in a model, it remains part of that model and is saved with as part of the model file, even if it is not applied to anything. Textures may be deleted from a model only through the Browse Texture command. A delete button is provided in the Texture Browser dialog box to remove selected textures.

Once a texture is created or loaded into a model, it can be edited. It doesn’t matter where the texture originated, or whether it’s a surface or solid type texture…all textures are fully editable.

There are four ways available to open a texture for editing:

- **Using the Edit Selected command in the Textures menu.** This opens the texture dialog box for the texture applied to a selected object. This command is dimmed if no object is currently selected, or if multiple objects are selected with different textures.
- **Using the Edit Texture command, also in the Textures menu.** The texture to be edited is selected from the sub-menu. This command is available when at least one texture is defined. Its sub-menu lists only those textures currently part of the active model.
- **Using the Browse Textures dialog, also in the Textures menu.** In the textures browser, all loaded textures are listed by name. Buttons in this dialog box allow further manipulation, including editing.
- **Using the Edit button on the Textures palette.** In the palette, all textures are presented for selection by representative preview image, whether they are loaded in the active model or not. Selecting a texture from the palette, then clicking on the Edit button, automatically loads the textures and opens its editing dialog.

When the editing of a texture is complete, i.e — the OK button is clicked and the texture editing dialog closes, the program returns to the location in the model that invoked the editing. This is especially significant in the case of the textures browser dialog box, which then allows you to conveniently select another texture for editing.

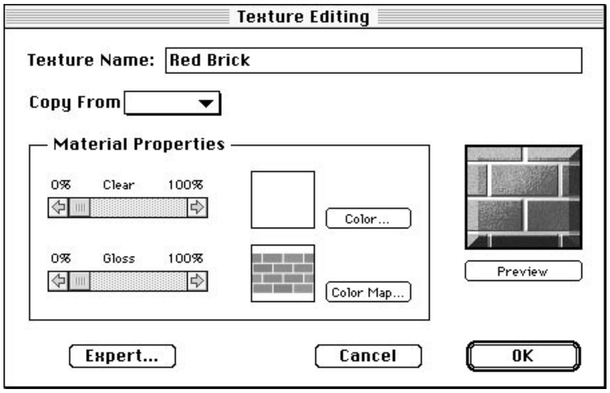
Surface textures — Explanation of fields

There are two dialogs available in the editing of surface textures, a primary and an expert. All surface textures use the same dialogs. The settings in these dialogs determine the properties of the texture named in the primary dialog.

Primary level dialog

When you select a texture for editing, the primary “Texture Editing” dialog appears first. It contains simple graphic controls and fields to set the various surface properties at a coarse level. For many textures this primary dialog provides sufficient control.

The primary dialog is graphic in design, allowing you to interactively adjust the color, gloss and clear values and immediately see the effect of the changes in a preview area of the dialog’s preview area. You may modify the color, gloss and clear settings of any surface textures, even those containing surface maps. Adding color has the effect of tinting a color surface map.



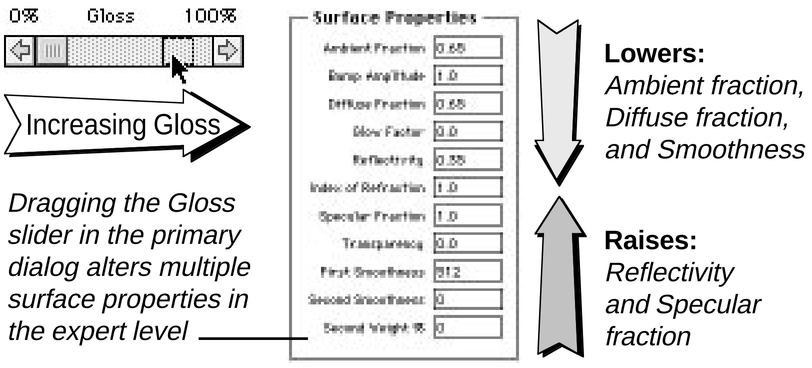
Texture Editing

Texture Name: Red Brick

The fields presented in the primary surface texture editing dialog are as follows:

- **Texture Name.** Enter a unique name in this field. The name chosen is entirely your choice…there are no naming constraints. Typically, the name you would use should be descriptive of the texture defined, not of the object to which you intend to apply it. This is especially recommended if more than one type of object will use the same texture.
- **Copy From.** Use this pull-down menu to copy the properties from a previously-defined texture that is also in the active model. Simply chose the desired texture from the menu. Releasing the mouse button copies all of that texture's settings to the open texture. The copied texture is unaffected, and any changes made apply only to the texture being edited.
- **Material Properties.** Two slider controls, titled "Clear" and "Gloss" allow you to control the properties of the texture in a simple, direct way. These controls are often all you need to set a texture's surface properties.

  The Gloss and Clear sliders actually control multiple surface properties at once. They are designed to adjust the appropriate surface properties in just the right combination to produce the desired result. While these same properties are editable in the expert level for individual adjustment, they require additional understanding to enter the appropriate values. Thus, these combination sliders are often preferable to direct control.



Surface Properties

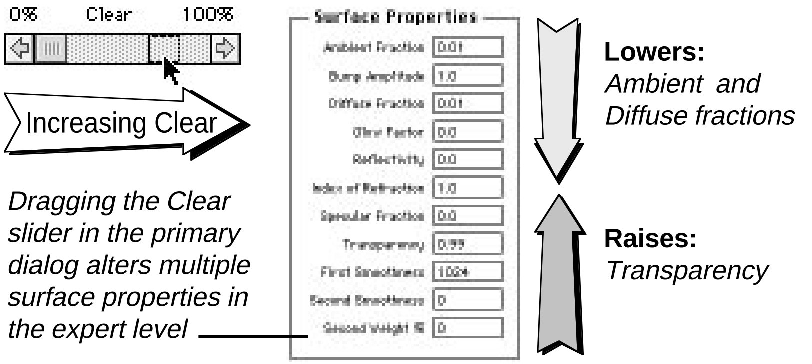
Ambient Fraction	0.65
Bump Amplitude	1.0
Diffuse Fraction	0.65
Glow Factor	0.0
Reflectivity	0.35
Index of Refraction	1.0
Specular Fraction	1.0
Transparency	0.0
First Smoothness	512
Second Smoothness	0
Second Weight %	0

Dragging the Gloss slider in the primary dialog alters multiple surface properties in the expert level

Lowers:
*Ambient fraction,
Diffuse fraction,
and Smoothness*

Raises:
*Reflectivity
and Specular
fraction*

The Gloss slider affects the amount of ambient response, the amount of diffuse light reflected, the amount of specular light reflected, the amount of reflectivity, and the level of smoothness. Dragging the Gloss slider to the right increases the "shininess" of the surface; dragging it to the left reduces the shininess.



Surface Properties

Ambient Fraction	0.01
Bump Amplitude	1.0
Diffuse Fraction	0.01
Glow Factor	0.0
Reflectivity	0.0
Index of Refraction	1.0
Specular Fraction	0.0
Transparency	0.99
First Smoothness	1024
Second Smoothness	0
Second Weight %	0

Dragging the Clear slider in the primary dialog alters multiple surface properties in the expert level

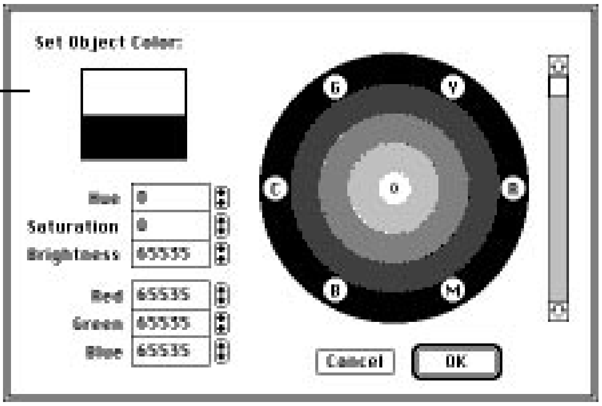
Lowers:
*Ambient and
Diffuse fractions*

Raises:
Transparency

The Clear slider affects the amount of ambient response, the amount of diffuse light reflected, and the level of transparency. You can see that several properties are affected by both slider settings. Dragging the Clear slider to the right increases the transmission of light through the surface; dragging it to the left reduces the level of light transmission.

- **Color button.** Click on this button to choose a color as part of the texture definition. An Apple Color Picker dialog box opens. The configuration of this color picker dialog, and the number of colors that you may choose from depends on the current color depth of your monitor.

Apple Color Picker displayed on mono-chrome monitors

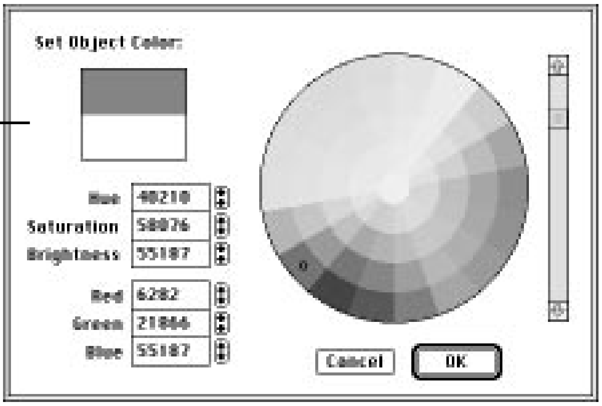


Set Object Color:

Hue 0
Saturation 0
Brightness 65535

Red 65535
Green 65535
Blue 65535

Apple Color Picker displayed on 8-bit (256 colors) monitors



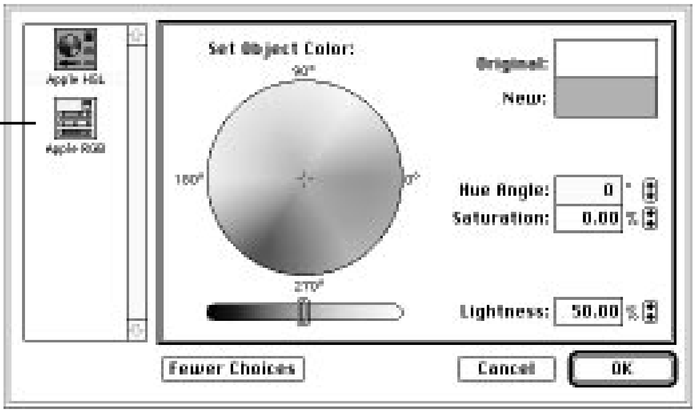
Set Object Color:

Hue 40210
Saturation 58076
Brightness 55187

Red 6282
Green 21866
Blue 55187

Apple Color Picker displayed on 32-bit (16.7 colors) monitors

(This color picker is part of the Color Sync™ package. It is a System 7 extension that replaces the standard format.)



Set Object Color:

Original:
New:

Hue Angle: 0 °
Saturation: 0.00 %

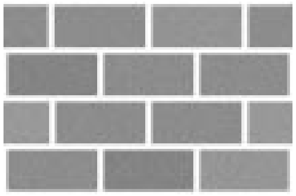
Lightness: 50.00 %

This field gives the texture a global color. When applied to an object, the entire surface of that object is tinted with this color. If used in conjunction with a color map, this color will also tint the colors present in the color map.

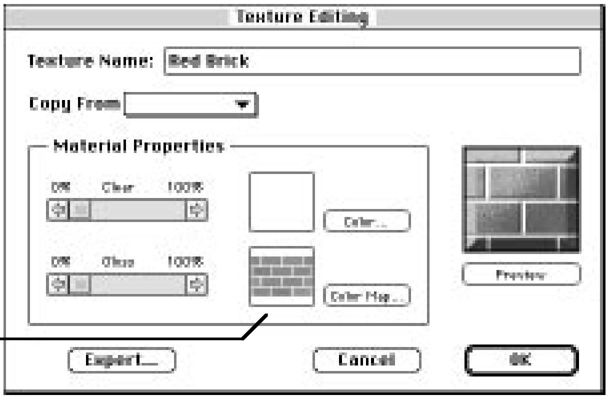
- **Color Map button.** Click on this button to bring a PICT image into the texture as a color map. When the texture is applied to an object, this map colors the surface pixel by pixel, based on the colors used in the map.

This is a small sample of a brick surface

(Note how the map's edges are set up to tile seamlessly, in all directions.)



Actual size: 2" high by 3" wide



Texture Editing

Texture Name: Red Brick

Copy From

Material Properties

0% Clear 100%

Color...

0% Gloss 100%

Color Map...

Preview

Export...

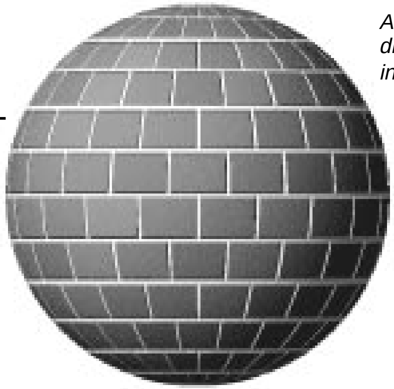
Cancel

OK

It is loaded into the Color Map field of a surface texture's primary dialog

…then applied to a sphere using spherical mapping and normal tiling

(The map's coverage was set to 15% hor. and 30% vert., then tiled or repeated as needed to cover the entire surface.)



Actual size: 4" in diameter, 12.6" in circumference

- **Preview.** The preview area uses a generic object to show the texture's appearance. This preview object is separate from any objects in the active model.

- **Expert button.** Click on this button to open the second expert-level dialog. This is for adjusting the surface texture's properties in a more direct manner.

Once you're satisfied with the settings, click on the OK button or push the Return key to accept the texture you have defined and return to the model. If you make changes to an existing texture, then decide you want to revert to the previous settings, simply click on the "Cancel" button when exiting the dialog to discard any changes.

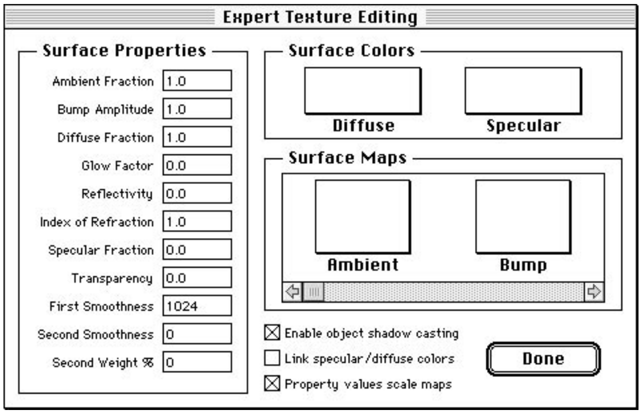
Expert level dialog

In the primary Texture Editing dialog, there is a button marked "Expert." Clicking on this button opens a secondary dialog providing specialized fields for defining more of the texture's parameters and using the advanced mapping features. A field-by-field explanation of the "Expert Texture Editing" dialog is located later in this chapter, under the Texture menu section.

This dialog is divided into three main areas:

- **Surface Properties**
- **Surface Colors**
- **Surface Maps**

The specific combination of values and PICT images stored within the texture creates a unique surface appearance, which can then be mapped onto object shapes.



Ambient Fraction	1.0
Bump Amplitude	1.0
Diffuse Fraction	1.0
Glow Factor	0.0
Reflectivity	0.0
Index of Refraction	1.0
Specular Fraction	0.0
Transparency	0.0
First Smoothness	1024
Second Smoothness	0
Second Weight %	0

☒ Enable object shadow casting
□ Link specular/diffuse colors
☒ Property values scale maps

Surface Properties fields are the principal settings that specify the overall appearance of a surface. Surface colors allow you to pick the global color properties for diffuse and specular reflections that may come from the surface. Surface maps work in conformity with the surface properties and surface colors fields to provide pixel-by-pixel control of these properties over the surface of an object.

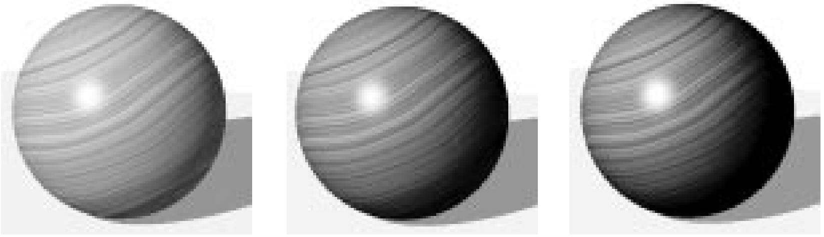
The expert-level dialog is where various types of surface maps are added to a texture. The various map fields use maps to provide pixel-by pixel control on the surface of an object for their respective property. For example: With a grayscale image in the Reflectivity map field, an object's surface becomes reflective only where the image is not black, and the reflectivity can vary with the levels of gray in the image. This is the mechanism used in the other mapping fields, as well.

Surface properties description

Vision3d's properties fields in the Expert dialog control parameters that directly affect how a surface responds or reflects light shining on the object to which it is applied. While experimentation is the most useful way to understand how these fields interact, the following field descriptions provide a beginning point and simple guidelines for their use:

- **Ambient Fraction.** This field determines how much of the available ambient light a specific surface reflects. Thus, even though the ambient light is constant throughout a given model, individual objects can react to that level differently. The range for this field is zero to one. A setting of zero means the surface will reflect no ambient light and will tend to be very dark (or black) in the shadows. A setting of one reflects all ambient light striking the surface.

Effects of the Ambient Fraction setting in texture definitions:

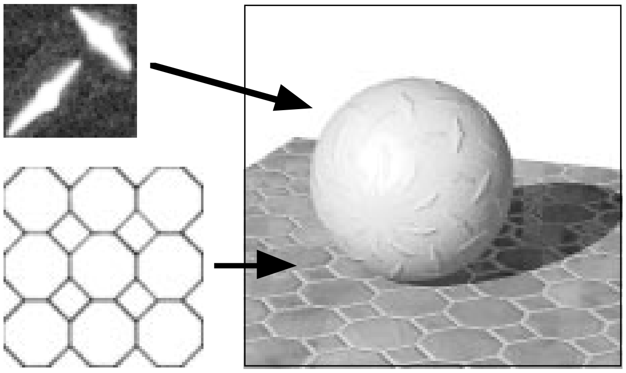


HIGH Ambient Fraction *MEDIUM Ambient Fraction* *LOW Ambient Fraction*

Notice that the lower the Ambient Fraction setting, the higher the contrast in the shaded areas of an object. The un-lit side is almost black in the image with a low ambient fraction.

The level of ambient light in a model is set through the Lighting palette. Ambient light is a non-directional background illumination. It is part of the diffuse component of the lighting in a model.

- **Bump Amplitude.** This field works in conjunction with the Bump Map field. It sets the depth of surface variations specified by the bump map field. If no bump map is included in the texture, this setting has no effect. Bump mapping allows the simulation of uneven surfaces too small or too detailed to be explicitly modeled.



White areas in boilerplate's bump map creates raised areas on ball

Dark areas in ceramic tile's bump map creates depressions in floor for seams of the tiles

A bump map is created as a PICT image. It is a flat representation of a depression or raised pattern you want applied to an object. The grayscale values in a bump map are used to vary the direction a surface's normals point, thus changing the brightness of the reflected light from a surface. Bump mapping does not alter the actual geometry of an object, but rather creates the illusion of depressions or highs in the surface.

While there is no limit to this field's range, the optimum values usually fall between one and ten. The higher the value, the greater the surface variations…assuming a bump map is present.

- **Diffuse Fraction.** Use this field to set the amount of diffuse light a surface reflects. A setting of zero reflects no diffuse light; a setting of one reflects all the diffuse light shining on it.

Diffuse surfaces reflect light in all directions in a somewhat uniform manner, producing a matte or dull appearance. A surface's apparent brightness is set with this field. The range of this field is zero to one. A zero value means no diffuse light is reflected and the surface appears dark, even when directly illuminated. Increasing the field value increases the amount of diffuse light reflected and makes the surface appear brighter.

NOTE: *Typically, diffuse and specular levels are mutually exclusive, i.e. — real world objects do not have high diffuse and high specular at the same time. These are the two main light reflection fields.*

(The Reflectivity field is a special case of Specular reflection.) Other fields in the texture dialog may modify a surface, but can never completely obscure the basic nature of the light reflecting from it, as set by the Diffuse Fraction and Specular Fraction fields.

- **Glow Factor.** Use this field to simulate the glow of a surface that is illuminated from behind, i.e. — light bulbs, translucent covers on lights, lamp shades, etc. Because the rendering algorithms cannot detect back lighting, this field allows you to create the effect. A setting of zero means no glow factor; there is no maximum value, but a value of one is usually enough to make the object fully glowing, without any shading of the surface from outside light sources.

The light bulb in this table lamp is just a regular lathed object with a point light inside. It was given a texture with a high Glow Factor to simulate the light glowing within.

This field enhances the level of diffuse light that a surface reflects, even beyond what the full setting in the diffuse fraction field would produce on its own. At higher levels, the object appears to glow or emit light. However, do not confuse a glow factor-enhanced surface with an actual light source. A glowing object will not illuminate surfaces adjacent to it, except when rendering with Raydiosity which can calculate inter-object diffuse illumination. You may simulate the effect, however, by combining a glowing surface with appropriately placed light sources. This is often done to give the proper appearance of light fixtures and neon shapes.

- **Reflectivity.** Use this field to control the level of reflectivity on a surface. Reflectivity is a special phenomenon of specular light. At its shiniest — when the intensity of the reflected light is equal to that of the incident light and where the light being emitted has little diffuse component — the surface functions like a mirror. If there is no diffuse component at all, the surface becomes a perfect mirror.

The color of the reflected light is filtered by the Specular Color field, if a color choice is present. The range of this field is zero to one.

NOTE: *When creating a texture with high reflectivity, you must also reduce the ambient and diffuse fractions a corresponding amount. Otherwise, the ambient and diffuse light reflecting from a surface will obscure its reflectivity, no matter how high the reflectivity field is set.*

- **Index of Refraction.** Use this field to control the amount of refraction visible through a transparent surface. The range of acceptable values starts at 1.0, which is air or no refraction. There is no upper limit, but in the real world there is no known transparent substance with a refraction index higher than 2.0. Glass typically has a refraction index of 1.5 to 1.6.

Refraction takes place at the surfaces of an object, where the light rays enter and exit the material. A transparent surface's refractive effect on the light coming from behind it depends on the angle of incidence and the shape of the surface at the points of incidence. The general effect of refraction is to shift the incident light to a parallel path as it exits the surface. At sea level, air has a refraction index of 1.00. Surfaces heavier than air have an index higher than one.

In the real world, the index of refraction for a substance is also a function of the wavelength of the incident light, such that different color components of the light will be refracted at different angles. In Vision3d, only the raytracing algorithm calculates refraction. It does not consider wavelength in its refractive process. Instead, an average index of refraction is used and all colors are refracted the same amount.

In other words, *do not expect to be able to create a light spectrum on a wall by shining a white light through a prism-shaped object with a transparent and refractive texture applied. There are two reasons for this: First, as mentioned, the color of light sources is modeled as function of RGB color space, not wavelength; Second, for the same reason you cannot illuminate objects with back lighting, raytracing traces rays of light backwards from the view plane through the model. Thus, a light source "behind" a transparent object will not be detected and have no effect on the object until after that object has been rendered.*

- **Specular Fraction.** As with the Diffuse Fraction field counterpart, this control directly affects the level of specular light coming from a surface. A high specular component in the light normally gives strong highlights or pinpoints of light that makes the surface appear glossy.

  Specular light reflections are typically in the color of the light source, unless a specular color is specified. Specular highlights are a subtle, but important, part in giving rendered images their photo-realistic "feel." The range of this field is zero to one.

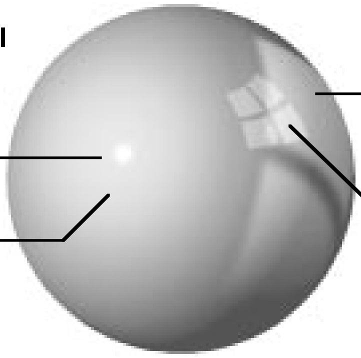
- **Transparency.** Use this field to control the amount of transparency for a surface. The value range in this field is zero to one. Zero defines an opaque surface, a value of one is fully transparent. Light passing through a surface is called transmitted light. Like reflectivity, the more transparent a surface, the lower the ambient and diffuse fields should be. This is to remove light reflecting off an object's surface, which creates a haze that obscures the transparency. If values in the ambient, diffuse and specular fields are zero, a fully transparent surface may also be fully invisible.

  When a transparent texture is applied, light coming from an object is modified to include contributions from light transmitted through it. In most cases, the light coming through an object is light-reflecting from surfaces of other objects. Diffuse and specular light passing through the surface are transmitted without adjustment., but is filtered by the Color field, if a color choice is present. However, light coming from the surface is controlled by the Ambient and Diffuse fields.

- **Smoothness and Weight Percentage.** The smoothness of a surface affects the nature of the light reflecting from it. The range is from zero to infinity. Higher values (500 and higher) mean smoother surfaces, which render with tighter specular highlights and sharper reflectivity. Lower values spread the specular hot spots and slightly blur the reflected objects.

The First Smoothness field is used by all textures. However, the second smoothness field is provided specifically for the creation of metallic surfaces. As you examine metallic objects around you, note how they have a high degree of specularity and that the hot spots tend to have halos around them. The specularity is controlled with high specular fraction (and correspondingly low ambient and diffuse fractions). The halo effect is simulated by using the first and second smoothness fields in combination, mixed by the Weight % field.

How to make metal objects look like metal



Specular highlight from primary light source

“Halo” around highlight

Halo is from a very low value (6) in the second smoothness field

Windows gel projected from secondary light

Reflection from Windows-reflected environment.

Note that reflections are in color of the metal, not of the light. Use “Link Specular/ Diffuse Colors” checkbox in Expert textures editing dialog to set this.

Think of the First Smoothness field as the fineness of the machined surface and the Second Smoothness field as the coarseness of the crystalline structure of the material itself. The amount that the underlying crystal structure influences the surface is determined by the Weight % field. For metals, the first smoothness is typically set high (around 1,000), and the second smoothness is set very low (around 5), with a weight percentage usually between 25% and 50%. The settings in these smoothness fields have a subtle, yet observable influence on the light reflected from a surface. The Weight % field is a percentage value with a range from zero to 100%.

Example settings for popular textures

A table of popular textures is included in the appendix of this manual. It provides example values for the expert Surface Properties fields. Note the interaction between the various fields to achieve each type of surface, especially how the Ambient Fraction and Diffuse Fraction vary in relationship to Specular Fraction, Reflectivity, and Transparency.

Even if the particular texture you want is not represented there, you should be able to extrapolate the texture you want from these example, but only if you understand the concepts involved.

To review

From this discussion, there are some general rules that indicate patterns in the settings of texture surface properties:

- **Surfaces become visible** when light is reflected off them toward the camera's direction. The amount and type of light reflected tells you what material that object is made of.
- **Light reflecting from the surface of an object contains two components:** 1) diffuse and 2) specular. Surfaces may reflect just one type of light or both types in combination. A surface with a high diffuse component reflects light in all directions, causing it to appear matte or dull. Matte surfaces have the same brightness from all viewing angles. Surfaces with a strong specular component tend to reflect light only in the direction of the angle of reflection, giving them a shiny or gloss appearance. For most surfaces, specular reflections appear not in the color of the surface, but rather the color of the light source.
- **Dull or matte textures** must have high ambient and diffuse fractions, with low specularity and reflectivity.
- **Shiny or reflective textures** have a high specularity fraction (and/or high reflectivity), with low ambient and diffuse settings.
- **Glowing objects** should have full ambient, full diffuse, no specular, and no reflectivity. Set the glow factor to .5 or higher, depending on how much you want it to glow. Glowing objects are not light sources, but can hold light sources inside or behind them.
- **For black objects,** a surface with no ambient, no diffuse, no specular, no reflectivity (and no glow factor) is essentially a "black hole." Since no light of any type reflects from its surface, it appears black. This is often a better technique than using the color black. If the object is too black — no details on the surface — increase the ambient and diffuse slightly for a dark gray; more for a lighter gray.
- **For shiny black,** just increase the specular (and/or reflective) fields, but keep the ambient and diffuse at zero.
- **For a perfectly invisible surface,** use no ambient, no diffuse, no specular, and no reflectivity, but set full transparency. If the texture

has a refractive index greater than one, it will be perceivable only by the distortion of objects located behind it.

- **For a perfectly reflective surface,** use no ambient, no diffuse, no specular, and no transparency, but set full reflectivity. The object is discernible only by the objects it reflects, which is light is purely specular in nature (even though the specular fraction is off), and not from any diffuse component coming from its surface.

Also remember the three checkboxes at the bottom of the expert-level texture editing dialog box:

- **For metals,** use the "Link diffuse/specular colors" checkbox. Metallic surfaces reflect in the color of the metal, not the color of the light, like other types of surfaces. Note that there must be something to reflect in the object to make it look metallic. Otherwise the surface will appear dark and flat. You can use an environment for realistic reflections of lights and darks, or build a fully-developed model to reflect in the metallic surface.
- **To allow light to pass through a surface,** turn off the "Enable object shadow casting" checkbox. You do not have to make the surface transparent. This feature would be used with glowing textures where a light source is positioned inside or behind the objects.
- **When using surface maps,** make sure the "Property values scale maps" checkbox is enabled. Otherwise, the maps are given full range.

The most important thing to remember with textures is that the settings for a given surface appearance are not just picked out of a hat. They must be carefully measured and composed. There is no guarantee of realism; it is up to the designer to analyze a surface being simulated and to supply a rationale for the resulting texture definition of the desired surface.

The second most important thing to realize is that the settings in the Surface Properties fields described above interact with each other in complex ways, and that there is no single set of "right" values. Additionally, the fields in a texture definition are not logically exclusive. It is possible to create a surface that is at once transparent, highly reflective, yet coarse and with a matte surface. Or think about the possibility of transparent wood — something that does not exist in reality, yet is easily described and rendered in Vision3d.

In a more useful example, a texture may have a surface map giving the appearance of marble, a high smoothness factor, and a reflectivity that hints of a polished finish, but with just a touch of transparency thrown in to give the appearance of a thin sliver of stone with a translucent quality. The possibilities are limitless, but it does take experimentation and extra patience at first to learn the best techniques.

Surface maps — explanation of fields

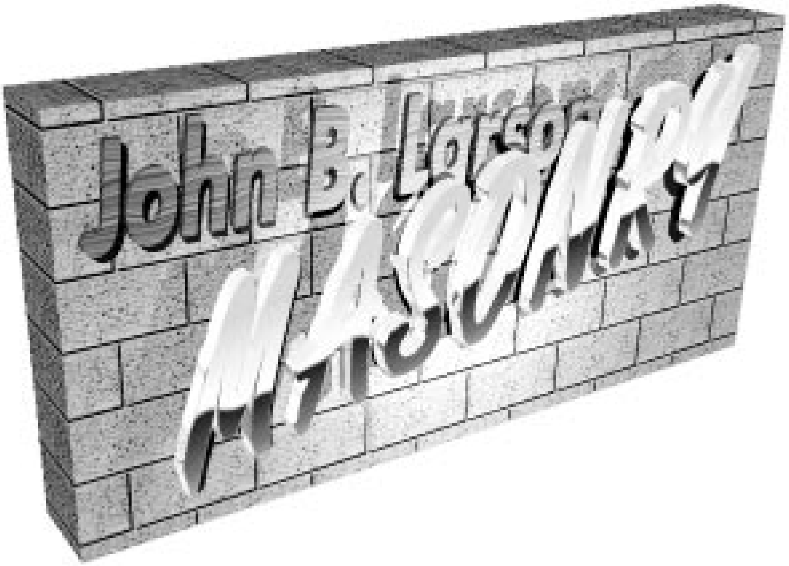
Rendering objects with textures using only the Surface Properties fields produce images with nicely smoothed, uniform surfaces. …Which makes them look computer generated. This is in contrast to the majority of objects we see and feel in the world about us. Our goal, then, should be to render images that look like they DIDN'T come from a computer. This is where surface maps can help.

The usual direction taken to enhance surface detail is to build complex surfaces with explicit geometric modeling. *This is the wrong technique!* Surface maps, in the form of PICT images, are more efficient, easier to create and offer an attractive alternative to building complex surfaces.

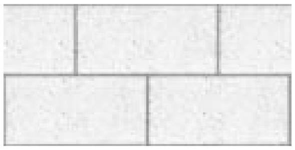
For example: Rather than creating a cinder block wall piece-by-piece in Vision3d, you need only use a digitized image of a section of a real wall and map it to a single polygon object of appropriate size. In addition to being easier and more efficient, the model renders faster because there are fewer objects (and bézier patches or polygons making up those objects) to process.

Cinderblock Wall

This wall is modeled from a cube primitive for efficiency (using the Scale tool to flatten it), then the cinderblock texture is applied from the front. The mapping style is set to cubic, with tiling to repeat the map so it covers the entire surface of the object.



Example of cinderblock surface map in texture applied to above wall



Most of the surface properties fields described in the previous section have a corresponding surface map field. They work together, if a map is present. The surface properties fields set the level or ceiling for each property. The map modulates within the range set. If no surface map is in place for a certain property, no modulation occurs and all areas of the surface appear at the level set by the Surface Properties field; it becomes a global setting.

The properties that don't have a corresponding map field include Index of Refraction, First Smoothness, and Second Smoothness. Thus, refraction and smoothness properties are, by definition, always global in their effect.

Color or Grayscale maps

Two of the mapping fields accept color images: Diffuse Color (A.K.A. Color map in the primary texture editing dialog) and Specular Color. The rest of the surface map fields REQUIRE grayscale or B&W only.

NOTE: *Do not confuse the Diffuse Color and Specular Color maps, which use color maps, with the Diffuse and Specular maps, which use grayscale maps, or the Diffuse and Specular color fields, which are not map fields but global color settings. While their names are similar (because they all have effect on the reflected diffuse and specular light), each of these fields perform a different function.*

Color maps control the color of light reflecting from a surface, based on the color of the pixel in the map at each point on a surface. The shading algorithm then uses those colors, along with the color, intensity and direction of the light falling on an object to calculate the actual rendered color of each point on its surface.

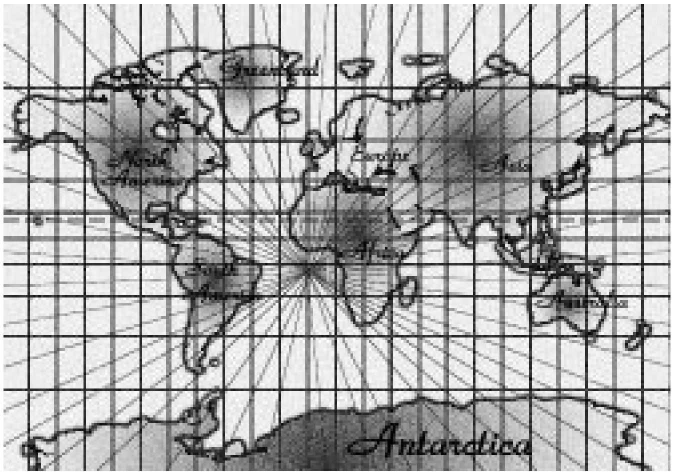
Grayscale maps control the amount of the reflected light for each type of surface property, again based on the gray level of the pixel in the map at each point on the surface.

Using Diffuse and Specular maps.

As noted, diffuse color maps are the main method of putting color images on objects. They are visible when diffuse light reflects off the surface. On the other hand, specular color maps allow you to have pixel-by-pixel color control of the specular light reflecting from surfaces.

STRATA™
VISION3d

Diffuse Color and Specular Color surface maps are usually created in a paint, drawing or image-processing application with any color palette — from one- to 24-bit. For realistic appearance, a color palette with at least 256 colors (8-bit) is recommended for color maps; a maximum of 256-grays is supported for grayscale.



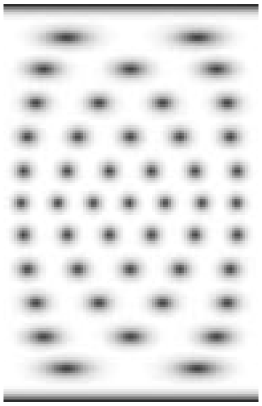
The depth at which color or grayscale surface maps are stored within Vision3d is set at the time they are loaded into a texture. this can be different than the pixel depth at which they were saved from other applications. Setting a surface map's pixel depth is discussed later in this chapter.

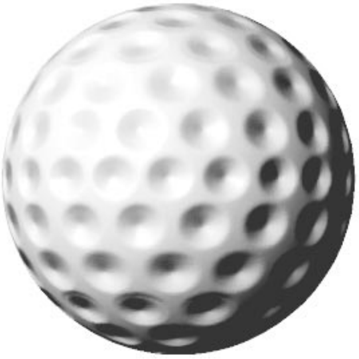
Using Bump, Diffuse, Glow, Specular, Reflectivity, and Transparency maps

Pixel-by-pixel color control is not the only kind of information that can be mapped. Surface texture can be simulated using a technique called bump mapping.

In addition, a pixel-by-pixel description of a surface's reflectivity, transparency and glow characteristics can also be mapped to an object. These maps do not use colors to generate their effects, but work with the brightness values of gray-scale images.

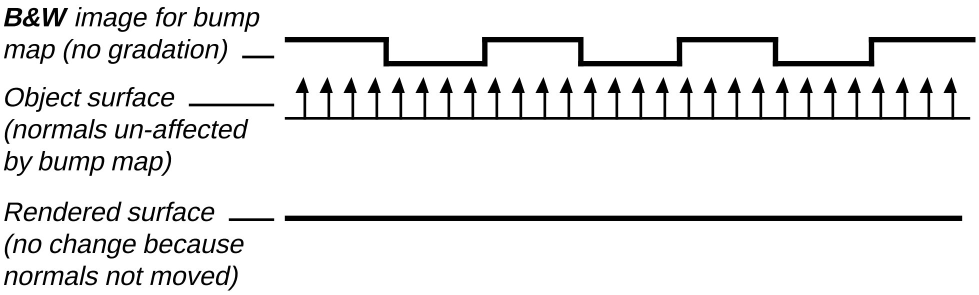
- **Bump Maps.** A bump map is a flat representation of an uneven surface which manipulates the interaction of light with a surface. In this manner, it fools the renderer into thinking the surface has highs and depressions in it — the dark areas visually pushing the surface down; the white areas leaving it at normal height. Bump mapping avoids the complexity and overhead of having to model explicit geometry to create secondary surface variations or "bumps."

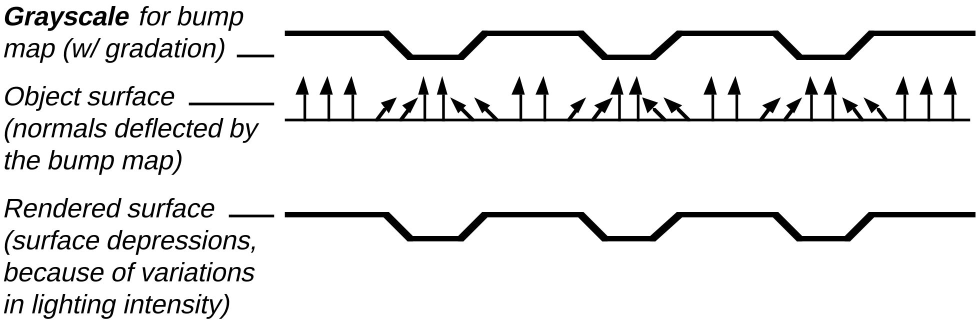




It is very acceptable for the majority of usages, even though the bump mapping illusion can break down when viewing profile edges — it does not change the silhouette of an object. Bump mapping does not alter the actual geometry of an object's surface, but rather alters the direction its "surface normals" are pointing.

Because bump mapping works with surface normals, the map requires transitional grays between the white (high) and black (low) areas of the map to determine how deep the depression appears. This can be introduced in the map by simply blurring the image once or twice in the image processing application before loading it into the Bump map field of a texture.


B&W image for bump map (no gradation)
Object surface (normals un-affected by bump map)
Rendered surface (no change because normals not moved)


Grayscale for bump map (w/ gradation)
Object surface (normals deflected by the bump map)
Rendered surface (surface depressions, because of variations in lighting intensity)

In technical terms, these transitional grays provide gradient vectors to "perturb" or bend the surface normals. A sudden change from white to black, with no gradient between them, will not register as a surface depression.

- **Diffuse Maps.** A diffuse map controls how much diffuse light is reflected from an object's surface, based on the pattern in the map. Black areas prohibit any light from being reflected; Dark areas suppress diffuse light according to the level of gray; white areas allow full diffuse reflection (at the ceiling specified in the Diffuse Fraction surface property field)

  Again, do not confuse the function of a diffuse map with that of a diffuse color map. The diffuse map controls the amount of diffuse light reflected, while the diffuse color map determines the color of the light being reflected.

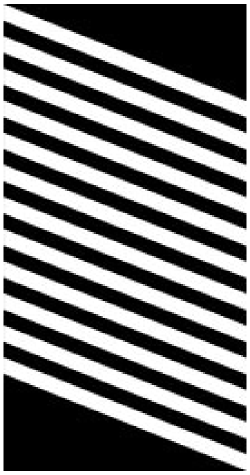
- **Glow Maps.** The glow map determines where (and how much) glow occurs on an object's surface, based on the pattern in the map. As with other grayscale maps, glow maps use whites to allow the highest amount of glow, while grays lessen the glow, and blacks completely suppress it. The maximum glow is determined by the setting of the Glow Factor surface properties field.

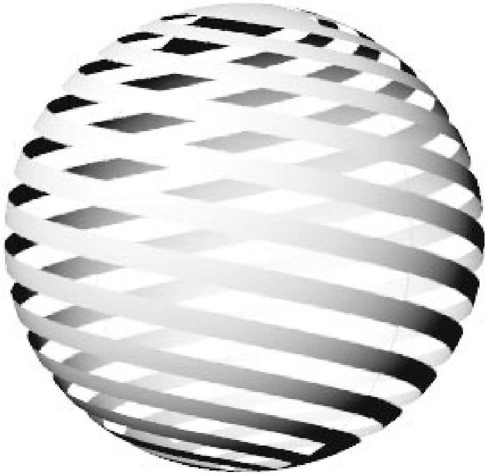
Vision3d's "glow" feature is provided to simulate the appearance of items illuminated from within or behind. A glow map lets you control where the glow occurs on a surface. Without a glow map, the entire surface glows at the brightness set in the Glow factor field.

As noted earlier in this chapter, objects with a glow factor in their texture are not light sources and will not illuminate adjacent objects.

- **Specular Maps.** Using a specular map in combination with its corresponding specular fraction field allows you to model surfaces that vary in their gloss or shininess. When manipulated properly, the interaction of the specular and reflectivity maps can provide important visual clues about the composition of objects and accurately simulate surfaces ranging from metals like gold, aluminum or brass, to plastics, glass, paper and paint.
- **Reflectivity Maps.** Reflectivity mapping uses a grayscale image to vary the level of reflectivity across an object's surface, based on the pattern in the image. Again, the Reflectivity field in the Surface Properties determines the maximum reflectivity and scales the range of the reflectivity map. Whites in the image correspond to the most reflective areas of the surface. Grays lessen the reflectivity, and blacks produce no reflectivity at all.

  A reflective surface is defined using the global setting of the Surface Properties reflectivity field. However, few surfaces (except perhaps mirrors) are perfectly uniform in their reflectivity. By using a surface map to define the pixel-by-pixel reflectivity values, you can more realistically model polished or shiny objects.
- **Transparency Maps.** Use transparency maps to define areas on a surface that are transparent. This mapping type, more than any other, can give the illusion of geometry different than the actual object. You can make holes through objects, eliminate parts of an object, add features that do not exist in the object. Examples of transparency mapping include doorways, windows or other openings in walls, windows in an airplane fuselage, holes through the surface of any object, and mapping complex outlines to simple objects.





Gray-scale maps also allow you to simulate an object's translucent patterns or irregularities. With a gray scale map, the white areas of the map correspond to the highest transparency level (that allowed by the ceiling value set in the transparency field), grays produce translucence and blacks result in opacity.

Using Multiple Surface Maps

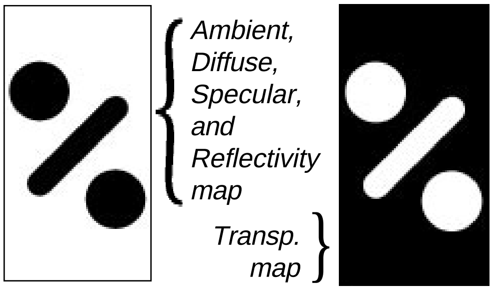
It is very common to use the same grayscale image, or an inverse of it, in several map fields. Typically, the various maps are created from the same master image. That way, they are guaranteed to register correctly.

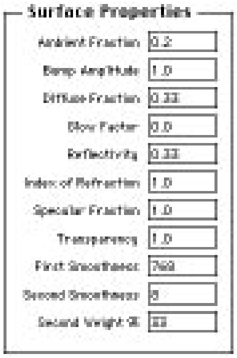
For example, suppose you have a lathed object that you want to make partly transparent, i.e. — appear as if there is a hole through the surface. Now, you could try to model the object so its geometry actually did have a hole. But, sometimes the tools and modeling extensions in Vision3d just aren't able to do what you want. What then? Where the tools leave off is where the power of mapping begins.

When using mapping to make holes, there is one very important concept to be remembered. Those areas of the surface you want to make a hole must also suppress the ambient, diffuse, and specular light that would normally reflect from the surface. Otherwise the hole will not be believable. This requires using multiple surface maps to control the various surface properties in specific ways.

A transparency map (at right), is combined with ambient, diffuse, specular, and reflectivity maps

Then, the Surface Properties fields are set as shown (far right)



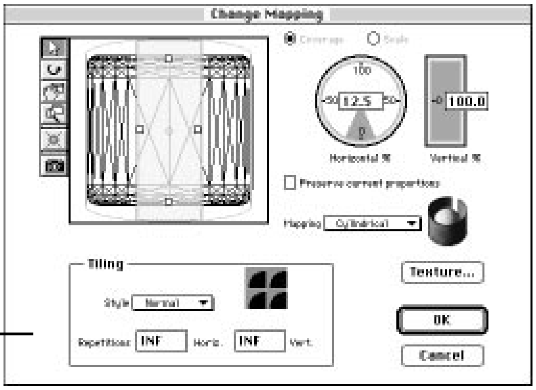


Surface Properties

Ambient Fraction 0.2
Bump Amplitude 1.0
Diffuse Fraction 0.33
Glow Factor 0.0
Reflectivity 0.33
Index of Refraction 1.0
Specular Fraction 1.0
Transparency 1.0
First Smoothness 768
Second Smoothness 8
Second Weight % 33

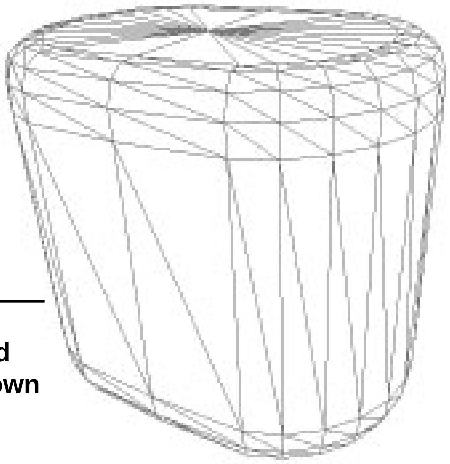
The texture is applied to a stretched, rounded cylinder.

In the Change Mapping dialog, the texture is set to wrap in a cylindrical pattern, with normal tiling

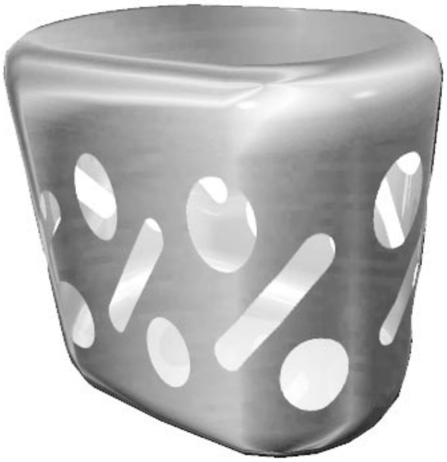


Change Mapping

● Coverage ○ Scale
Horizontal % 12.5
Vertical % 100.0
□ Preserve current proportions
Mapping Cylindrical
Tiling
Style Normal
Repetitions INF Horiz. INF Vert.



Wireframe object



The rendered object is shown at far right!

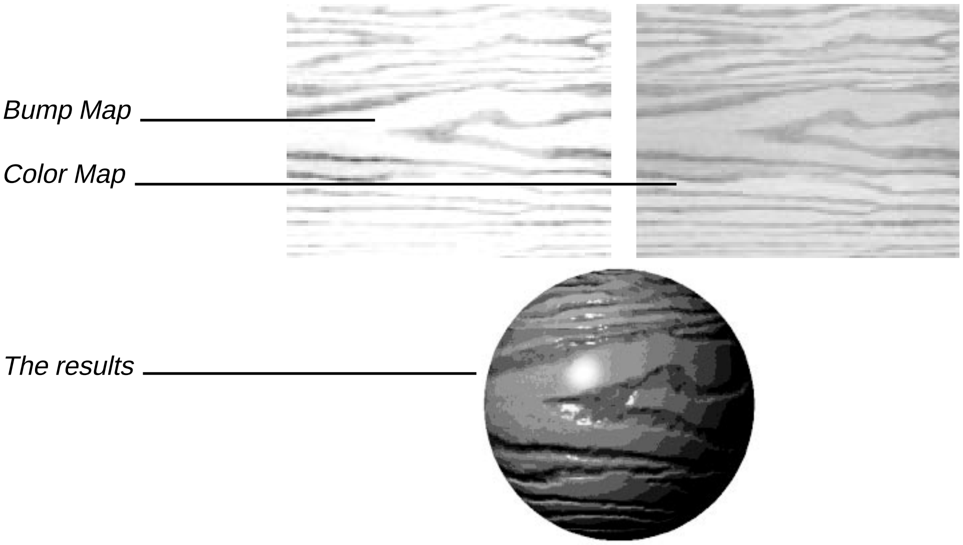
Thus, from the above illustration, you can see how the ambient, diffuse, specular, and reflectivity maps are the same image. In this case, because the surface has been given some amount of reflectivity, a reflectivity map is also required to suppress visible reflections on the surface where there are supposed to holes.

Note how the four light-handling fields all use the same map. On the other hand, the transparency map must be an inverse of the other maps. Remember, each of these mapping fields work exactly the same way, i.e. — white allows the specific property for that field, black suppresses it, and gray modulates some amount in between depending on the level(s) of gray. In the case of the transparency map, white is the area of the surface that is to be a hole. But note that this is the same area on the surface that the ambient, diffuse, specular, and reflectivity maps must be black to suppress those properties, since no surface is supposed to be apparent where there is a hole.

This is a typical use for multiple surface maps. When transparency mapping is used in conjunction with maps in the other fields, the object responds to the light sources as if there are actual holes in the surface…and the rendered image is totally believable.

Registering multiple maps

You should insure all the maps you intend to use together in a texture will align or register with each other, and work correctly with each other, before inserting them into a Vision3d texture. No controls are provided to independently position surface maps once they are within a texture, so the details or patterns within the maps must be registered with each other during their creation in the paint or image processing application used to make them.



Thus, when creating your own multiple surface maps for use in a texture, they should usually be the same size so that the areas of detail in each match up. If the texture is to be repeated to fully cover the surface, the tiled repetitions will match properly.

For example, this would be necessary if you wish the roughness of a bump map to match the grain in a wood color map.

This is important to understand. When applying a texture with multiple surface maps, they are manipulated as one, i.e. mapping style, scaling, rotation, etc. All maps are aligned or centered on the texture origin point of the object and are oriented according to the active view in which the texture was applied. All maps in the texture are scaled in unison using the Change Mapping command in the Textures menu. Changing a texture's orientation on an object (by using the Control key with the object rotate tool), rotates all maps together.

Using maps of different sizes

If required, you may use maps of different sizes in the same texture definition. This could occur when you wished to repeat a small pattern as a bump map, but had a larger color map meant to fit the object without tiling. When the tiling style is enabled, maps of different sizes repeat independently across an object's surface.

Making composite maps

As mentioned earlier in this chapter, only one texture can be applied to an object. At first thought, this may seem to be a significant obstacle to effectively using the mapping capability in Vision3d. In actual practice, with appropriate planning, this is a minor challenge.

It does require the use of “composite” maps, however. Composite mapping is a method of putting the properties of different surfaces in the same set of maps. Using this technique, you can have a colorfully painted area, a transparent area, a highly reflective area, and even a metallic area next to each other on the same object.

It is just a matter of identifying how you will be applying the texture, i.e. — the direction of application, the mapping style, and tiling method (if any). Then, decide what surfaces you want and where they need to be placed on the various maps as they wrap onto the object’s surface geometry. The specific maps required are determined by the surfaces you want, and how you plan to “communicate” the appropriate values for each surface property.

There is usually more than one way to build composite maps. Start with something simple — even a simple rectangle — and experiment with the concept. Composite maps works incredibly well, you just have to gain some experience to tap into their power.

Using wireframe images as templates

How do you know where to put the components or elements in your maps? Easy! Just get a template of the object on which you want to apply the surface maps. Save or screen-capture one or more wireframe PICT images of the object directly from the modeling windows (from the direction you plan to apply the texture). Then, open these wireframe images as templates in the application where you create the maps. That way, you can “paint” right over the wireframe (erasing to white the areas you don’t cover up) and you know the maps will be the right proportions and the elements within the maps will be positioned correctly relative to the object.

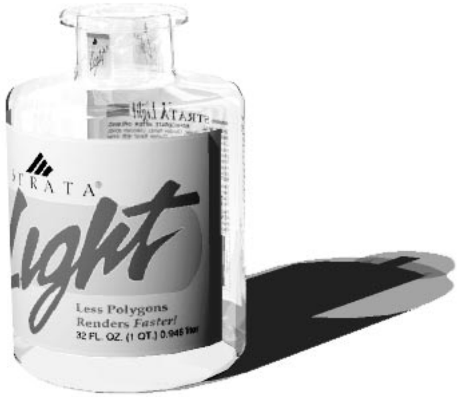
If you need to make the maps larger to improve the resolution, because you will be rendering at higher resolution, you can increase the size of the template an appropriate amount BEFORE you start drawing over it. The wireframe template will not be as sharp, but you’ll cover it up or erase it anyway.

NOTE: *The amount of resolution you need to build into maps for higher resolution renderings is discussed later in this chapter. Please refer to it AND UNDERSTAND IT before beginning a composite map project. You may find you do not need as much as you think.*

An example: Making a glass bottle with labels

Suppose you model a bottle that you want to appear made of glass except where two labels are positioned on the front and back. By compositing some appropriate maps, you will be able to make the bottle transparent except where the labels are positioned, which must also have a high amount of diffuse light reflecting from them for brightness.

Using the one texture/one object rule, if would not seem possible to have part of the bottle transparent and the part where the label is be opaque. Yet the following illustration is proof that it is not only possible, but it is a very useful and powerful technique!



The question is not whether it can be done,
but rather where do you begin?

Looking at the lathed bottle in the model window from the Front view (in wireframe, orthographic display), choose the Save As command. Switch to PICT as the save format, enter an appropriate name, and click on the OK button. Vision3d will save a draw-PICT file of the active view. Saving a PICT directly from the model window is usually the preferred way to generate the templates you need.

Alternately, you can use a screen capture to obtain the wireframe template. If you do not have a commercial screen capture extension that allows you to drag a marquee around the area of the screen that you wish to capture, you can use the built-in screen capture feature in System 7. However, this screen capture mechanism saves a PICT file of the entire desktop, so you will have to crop in on just the bottle template afterward. To initiate a System 7 screen capture, hold down the Command and Shift keys, then push the three (3) key. The Operating System responds with a camera shutter sound. After a brief pause, a file titled “Picture 1” appears in the root window of your machine’s start up hard-disk drive. Each subsequent screen capture is named Picture 2, Picture 3, etc. For more information, please refer to the Operating System’s documentation concerning this feature.

Planning the mapping

You will use the template to make a double-width map that wraps completely around the bottle. That way, you can have different labels on the front and back of the bottle, as well as any other surface detail you want anywhere on the bottle's surface…because you are using composite maps!

Most bottles are constructed with the lathe modeler. This implies that the mapping type to be used with a surface texture applied to the bottle would be cylindrical. The first thing you must decide is the dimensions of your bottle. It is essential for setting the proportions of the surface maps you will make to define the bottle's various surface properties.

Setting map size

So, assume you make your bottle 4″ in diameter and 6″ high. (This discussion assumes you know how to draw the bottle template and lathe it into a 3-D object.)

Using simple junior high math, you know that the circumference of the bottle is found with the formula C=2πr (where "r" is the radius) or its derivative C=πd (where "d" is the diameter). Thus, for a 4″ diameter bottle, the map would be 6″ high by 12.567″ wide. This will wrap completely around the bottle without distortion.

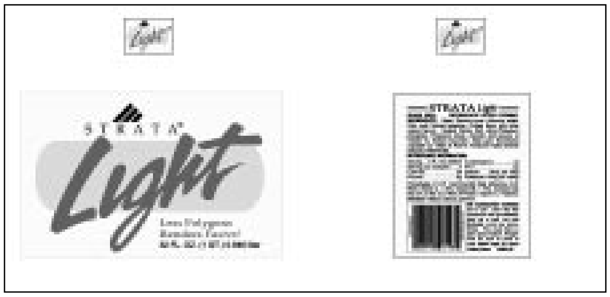
Actually, the only thing you're really interested in is the proportion or aspect ratio of the map. If the width is too long or too short for the height, the map will distort when wrapped around the bottle. The actual size isn't as important as the proportions, but if you work in actual size, it may be a little easier to visualize.

Making the maps

You can use a paint program, but it may be difficult to work with all the elements. It is usually better to use a drawing program to construct the label, then convert it to a PICT image for mapping. Adobe Illustrator works well as the drawing application, with Photoshop™ doing the conversion to PICT, then generating the other special grayscale maps that register with the original color map.

The color map

The first map you want to make is the color map (A.K.A. Diffuse Color map in the expert-level texture editing dialog). Here you will define the position and contents of the labels. Again, if you want to build this map in a drawing program, just make sure you can convert it to a PICT afterward. Of course, you want to crop the PICT document to the 6” high by 12.567” wide dimensions calculated earlier.

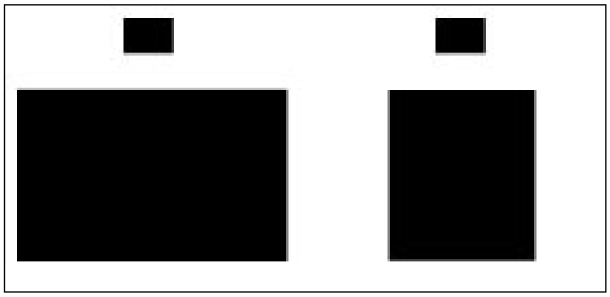


As you can see in the above illustration, there will be two labels — one front and one back. There are also two little logos that will show on the neck of the bottle. Notice that the rest of the color map is white. This is the area that corresponds to the area you want to be transparent.

If you want the glass in your bottle to have a color, the white part would be where you define that color. For best reproduction the color map is usually full 32-bit color. However, if you can switch it to 8-bit color, you will be able to save a substantial amount of memory. Save this map in PICT format, with a name like “Bottle - color map.”

Transparency map

Once you have the color map built and saved, you can use it as a pattern to make the transparency map. In this example, all you need to do is select the areas in the image that you have defined as the labels and fill them with black.



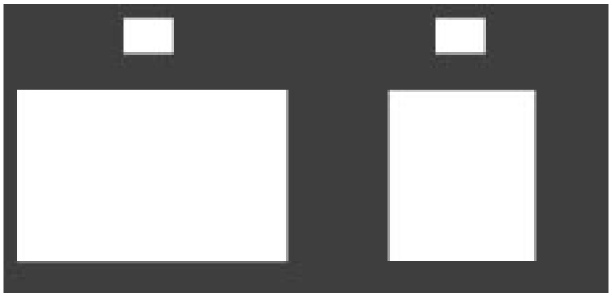
The label areas are nice rectangles, so the selection and fill process is fairly straightforward. The concept behind the transparency map is that the white parts allow that area of the bottle's surface to be transparent, while the black prohibits transparency. Since this is a grayscale map, in other projects you can also use levels of gray to set different levels of transparency on a surface.

Before saving this map, convert it to grayscale. Save it with a name like "Bottle - transparency." In actual use, when you create the Bottle Label texture, this same map will also be used in the Specular and Reflectivity map fields as well.

Ambient and Diffuse maps

Next, you need to make a map that designates which areas of the bottle's surface will reflect diffuse light. A little thought about this should give you an understanding that the labels should reflect the highest amount of diffuse light. On the other hand, you do not want a lot of diffuse light reflecting from the transparent area of the surface, because that would obscure the transparency. It would have the effect of making the transparent area milky or even opaque.

Remember, each of the mapping fields that use grayscale maps use the same principle: white allows that field's surface property, black prohibits it, and grays suppress the property depending on the level of gray.



As you look at the transparency map, you can see that by simply inverting it, the white parts of the map would nicely correspond to the labels and the black part would suppress the diffuse light where it will be transparent.

If the transparent area is pure black, no diffuse will be reflected from the surface of the glass, making it completely transparent. In actual practice, it is more realistic to make the glass area of this map about 80% black (or 20% white, whichever way you want to look at it). So, fill the black area with an 80% black — making sure the white parts are not encroached upon. You want that area of the surface to reflect ALL of the diffuse light.

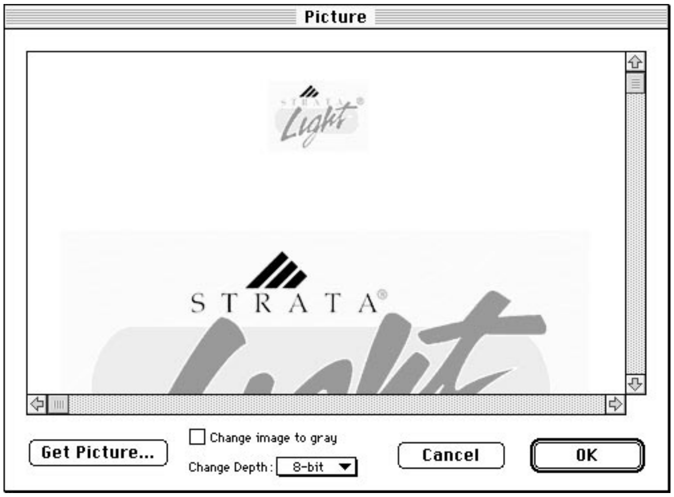
When this is placed in the ambient and diffuse map fields, only 20% of the diffuse light striking the glass area will reflect, providing just a hint of the surface.

Save this map with a name like "Bottle-amb/diff."

Putting it together

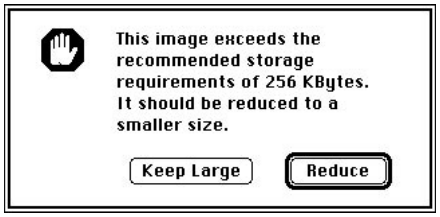
Now it's time to return to the Vision3d model of your bottle. With the bottle selected, choose the New Texture command in the Textures menu. Alternately, you can click on the New button on the Textures palette.

In the primary texture editing dialog, click on the Color Map button. The Picture dialog box appears. Click on the Get Picture button and load the Bottle - color map file. You should see your label masterpiece loaded into this dialog.



☐ Change image to gray
Change Depth: 8-bit

Notice the controls at the bottom of this dialog. The color depth defaults to 32-bit, even if the PICT file was saved in 8-bit as suggested. If you are using an 8-bit image for the color map, switch the setting in the Change Depth pull-down menu to 8-bit, then click on the OK button. The color map draws in miniature on the texture editing dialog.



This image exceeds the recommended storage requirements of 256 KBytes. It should be reduced to a smaller size.

You will see an alert saying this image is larger than the suggested 256K. For this project, you want this map and all the other maps to remain the size you made them. (Other projects may not have the same requirements and it would be appropriate to reduce them to save memory.) Choose the “Keep Large” option. *THE “REDUCE” BUTTON IS NOT THE CORRECT CHOICE HERE!* When using multiple maps that you have created to register with each other, reducing an image changes that map’s dimensions by whatever amount necessary to bring the image under the 256K memory-storage threshold; causing the planned registration between multiple maps to fall short of expectations!

Expert level

Next, click on the Expert button to enter the expert-level of the texture. This second dialog allows you to load the advanced maps and set the surface properties to match the maps you have used.

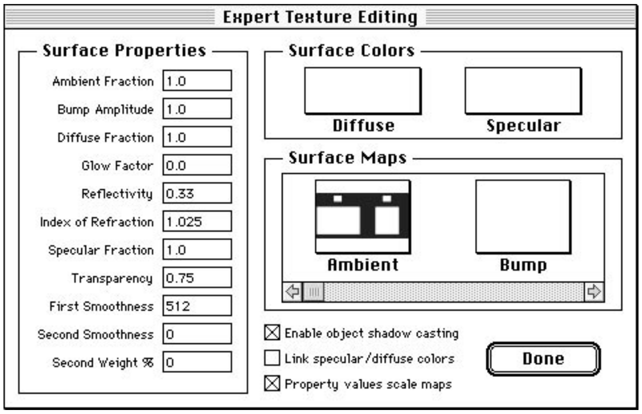
Click on the Ambient map rectangle. You will see the same Picture dialog used previously. Load the Bottle-amb/diff file, then click on the OK button. Note that the default color depth for these map fields is 8-bit grayscale.

Repeat the process for the Diffuse map, using the same file. Note that even though they are the same file, each map field must have its own image loaded into it.

Now, scroll the Surface Maps area of this dialog to the right to show and insert maps into the Reflectivity, Specular, and Transparency fields. This time, use the “Bottle-transparency” image for these three fields. Why? Because not only do you want to designate where the bottle will be transparent (the white part of the image), but you need to control the areas where the surface will be reflective and shiny. In this case, they happen to be the same areas. Thus, you can use the same map for these three surface properties.

Setting surface properties

With the maps in place, you now must set the ceiling values for the surface properties under which the maps are to work. Obviously, if you have installed a transparency map then you must turn on transparency. If the transparency surface property is set at zero, then the white part of the transparency map will still be zero (no transparency).



Expert Texture Editing

Surface Properties

Ambient Fraction	1.0
Bump Amplitude	1.0
Diffuse Fraction	1.0
Glow Factor	0.0
Reflectivity	0.33
Index of Refraction	1.025
Specular Fraction	1.0
Transparency	0.75
First Smoothness	512
Second Smoothness	0
Second Weight %	0

☒ Enable object shadow casting
☐ Link specular/diffuse colors
☒ Property values scale maps

Set the various surface properties to the values shown in the above illustration of the expert-level texture editing dialog. The rationale for the numbers should be fairly straightforward. Take time to try and understand them. Click on the Done button to close the expert dialog. Enter "Bottle Label" as the name of the texture and click on the OK button to accept the texture.

The Index of Refraction field deserves special mention again. As mentioned earlier in this chapter, the typical setting for glass is 1.5 or 1.6. You could use this value, of course, but the rendered image would give the appearance of a bottle filled with a very thick liquid, because the refraction would be so strong. The glass would show an exaggerated distortion of the back label through the glass of the bottle. If this is the effect you want, fine. Others may think the bottle looks more realistic with just a hint of refraction, like you would see from a thin-walled, empty glass bottle. This is an area in which you can experiment during the preview renderings.

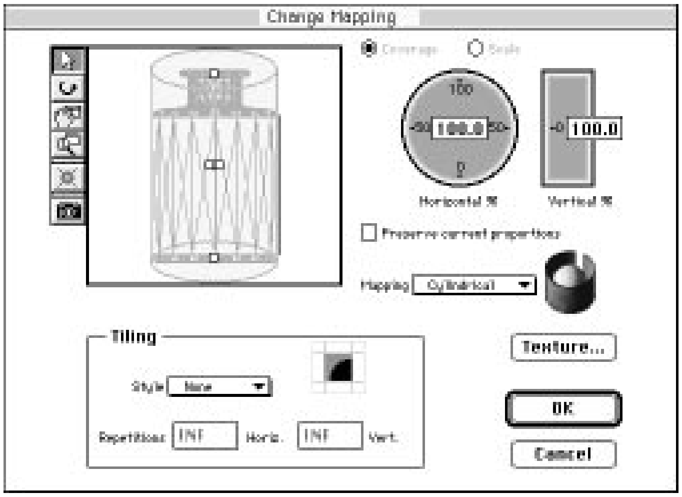
Applying the texture

As with any surface texture, the direction of application determines the orientation of the maps within it on the object's surface. Remember that the bottle's maps were created with the front label on the left side and the back label on the right. That means the center of the label is really meant to be the side of the bottle. A little mental visualization should help you understand the relationship of the maps to the bottle.

Thus, you want to apply this texture from the right side of the bottle, so the front label appears on the "front" of the bottle. Switch to the Right view, then choose the Set Selection command in the Textures menu. The Bottle Label texture should appear in the sub-menu. Move the cursor to highlight it in the sub-menu, then release the mouse button to apply it. Alternately, you could select the representation of the just-created texture on the Textures palette and click on the Apply button.

Setting the mapping parameters

Next, you must set the mapping parameters through the Change Mapping command. Choosing this command opens a dialog box that allows you to set how the bottle label texture is mapped on the bottle.



Change Mapping

● Coverage ○ Scale

Horizontal % 100.0

Vertical % 100.0

□ Preserve current proportions

Mapping: Cylindrical

Tiling

Style: None

Repetitions: INF Horiz. INF Vert.

Obviously, the first thing to set is the mapping type. Click on the Mapping pull-down menu and choose Cylindrical. Next, set the tiling style to none. This is not absolutely necessary, since the texture is designed to completely cover the bottle's surface without tiling, but turning it off allows you to see easier where it may not be covering (in case off an error in the dimensions).

Finally, you want to set the scale of the surface maps. Remember back to the beginning of this project. Using the Transform dialog, you made sure the bottle was 6" high and 4" in diameter, then calculated that the width of the label maps would need to be 12.567” to have the appropriate proportions. Thus, it is a simple matter to set the vertical coverage to 100.0 (meaning 100% coverage) and the horizontal coverage to 100.0 (meaning 100% around). Click on the OK button to accept these mapping parameters.

Rendering the bottle

Everything is now in place to render the bottle. You may want to make a couple of preview renderings to verify the mapping is the correct orientation and scale. You may also want to adjust the lighting and put some other objects in the scene to add to the realism. Also, be sure to check the Index of Refraction for your personal tastes and what you want to communicate as far as the thickness of the glass.

Of course, Raytracing is strongly recommended for realism in the rendered image. Only raytracing will accurately calculate the refraction and transparency. In the rendered image shown earlier, an environment was also loaded into the model to reflect off the glass and further enhance the glass surface. Again, note that the reflections only occur on the glass, not on the label. That’s because of the Reflectivity and Specular maps you loaded into the texture.

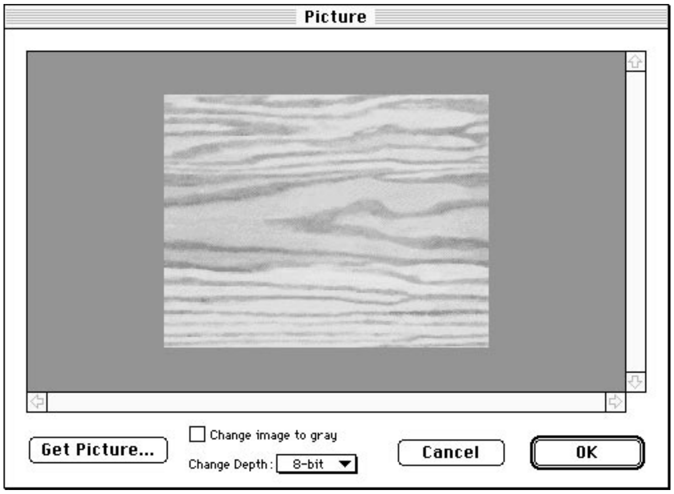
In the example used here, the bottle was lathed from a simple single-wall template. You may want to construct your bottle with a double-wall template so you can have stronger sense of glass thickness. That would also facilitate modeling something inside the bottle and still retain the glass refraction.

Loading and Removing Maps

Surface maps may be added or removed from a texture at any time. Using the Texture menu’s “Edit Texture” command or the Edit button on the Textures palette to open a texture for editing, then moving through the various dialogs as needed, you can insert previously-created surface maps quite easily. Each type of surface map is stored in a separate field within the expert-level texture editing dialog.

To load a surface map, click on the appropriate surface map filed in the expert dialog box for the map you are loading. A Picture dialog box appears. This is the dialog where surface maps are loaded into and/or deleted from an texture

Click on the “Get Picture” button at the bottom of the dialog and select the desired PICT file in the directory dialog that is displayed. You may also copy and paste images to and from the Scrapbook into this dialog. In other words, you have full access to the PICT images used for maps.



□ Change image to gray

Change Depth: 8-bit

To delete a map, open this dialog for the map you wish to remove and push the Delete key or choose the Delete command in the Edit menu. You may also remove it using the Cut command from the Edit menu.

Solid textures

As indicated, solid textures are volumetric in nature. That means, their patterns are defined as a three-dimensional volume out of which the objects appear to be carved.

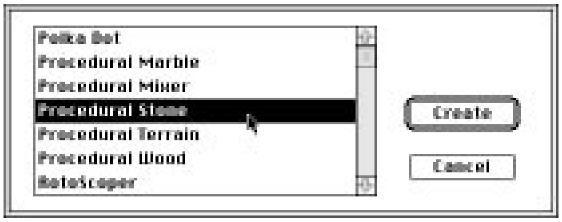
While solid textures allow you to set the full suite of surface properties, just as described in the surface texture discussion earlier in this chapter, they do not have the extended pixel-by-pixel control afforded by surface maps. Surface property settings in a solid texture are global in their effect. The pattern contained in a solid texture is roughly analogous to the diffuse color map in a surface texture, except that it is volumetric in the way it propagates the pattern and it is infinite in size.

Thus, although the scale of the pattern is adjustable, no mapping or tiling settings are required. When using the Change Mapping command to adjust the scale, position, and orientation of solid textures on objects, notice how the irrelevant portions of the dialog are dimmed. A Scale field is also included in each solid texture's editing dialog boxes.

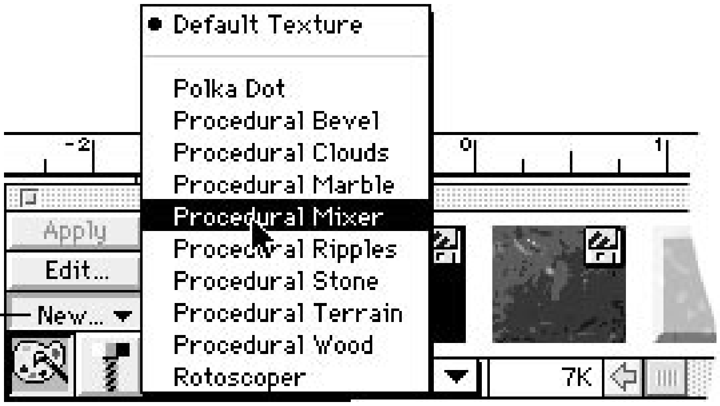
Alternately, you can use the Object Scale tool in combination with the Control key to scale the texture instead of the object in the model view. This works for both surface and solid textures, and is discussed in *Chapter 3: Modeling Objects*.

Several solid textures are shipped with the software and can be loaded into models, modified, and copied between models as needed. Additionally, you may make your own solid textures, based on the available texture extensions installed in the Extensions folder (within the Vision3d application folder).

Normally, selecting the New Texture command gives you the standard dialog for a surface-type texture. If you wish to create a new solid texture, hold down the Option key while selecting the New Texture command.



You are presented with a list of the available texture extensions (upon which solid textures are based). Choosing one of these extensions and clicking on the "OK" button opens the appropriate dialog box for creating a new solid texture.



Hold down mouse button when selecting New to choose from texture extensions.

Alternately, you can click on the New button found on the Textures palette. If you hold down the mouse button, a pop-up menu appears listing the same texture extensions. Choosing a texture extension from the menu opens the texture editing dialog box connected to that texture extension.

Texture extensions explained

Each solid texture requires a "texture extension" upon which to generate its color texture pattern. Because of their complexity, the texture extension files must be compiled and assembled externally with special segments of code that describe how the volumetric pattern will propagate volumetrically through an object. However, any number of solid textures can be created (and saved) from one texture extension; variations on a theme, as it were.

Sometimes, these texture extension files are also called “procedure maps.” Each procedure map (or texture extension) carries its own dialog boxes for choosing colors, setting scale, and generally describing how a solid texture using that map should appear when used in models.

It is important that you understand if a solid texture’s texture extension file is not found in the Extensions folder, within the Vision3d application folder, when you try to re-open the model, it cannot be accessed. In fact the model cannot even be opened…that is how important the texture extension is!

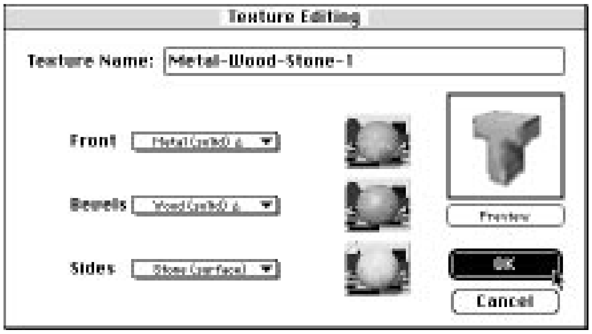
A solid texture can be saved as a separate file, which is then available for loading into other models as desired. But the texture extension that supports that solid texture must also be present for it to work.

Texture extensions, and the solid textures that use them are similar to the operation of PostScript® fonts — i.e., each font has two parts to work properly: the bitmap/screen font and the Type 1 or Type 3 outline support document. You may use a screen font in a document, when it is installed in your system. But to get the font-smoothing benefits and to re-open the document or open it on another machine, the postscript portion of the font must be present. Solid textures work in a similar manner.

Special solid texture #1: Bevel Texture

To use this extension, choose Procedural Bevel in the Procedural Texture Extension dialog, then click on the Create button.

Choose currently loaded textures for the front, bevels and sides.



Texture Editing

Texture Name: Metal-Wood-Stone-1

Front: Metal(solid)

Bevels: Wood(solid)

Sides: Stone(surface)

The Bevel texture allows you to specify three specific textures — either surface or solid type — that will be applied to the front, bevel, and sides of an object. A bevel texture can be applied to any type of object, although extrusions usually benefit the most from this type of texture.

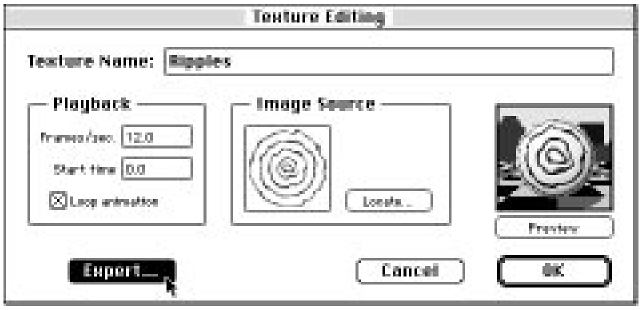
You can choose from textures currently loaded in the model. Only textures currently loaded in the model are available in the Procedural Bevel extension. Textures CANNOT be chosen from the texture palette while creating a new bevel texture; they must be loaded into the model FIRST.

To preview the newly created bevel texture, click on the Preview button. When satisfied with the bevel texture choices, click on the OK button. You can now apply the texture as usual. Remember, the bevel textures you create will be direction sensitive just like any other texture. That means you must take care to apply them from the appropriate direction in order to get the effect you want, i.e. — front texture applied on the faces that are parallel to the active view, side texture applied on the faces that are at right angle to the active view, and the chosen bevel texture to anything in-between.

Special solid texture #2: Rotoscoper texture

The Rotoscoper extension allows you to use a QuickTime movies or PICS files as maps that can be applied to objects. As with other solid textures, the Rotoscoper texture is accessed by holding down the Option key while selecting the New Texture command.

This texture allows you to "play" an animation file as a surface map on objects as part of a rendered animation. The texture dialog provides fields for setting play speed, as well as when to actually start playing the chosen animation within the larger animation. Once a Rotoscoped animation map is applied, you can scale it to fit the object as you would any surface-type texture through the Change Mapping command in the Textures menu.



Texture Name: Ripples

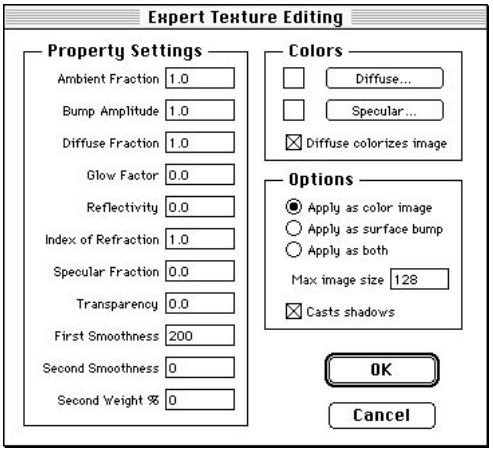
Frames/sec. 12.0

Start time 0.0

☒ Loop animation

As with every other texture, there is an expert-level secondary dialog with fields for setting the general surfaces properties. From discussion earlier in this chapter, you should already be familiar with these fields.

Additionally, there are some special rotoscoping options for setting the maximum frame size of the animation being used and whether the frame will be applied as a color image, a bump map, or both.



Expert Texture Editing

Property Settings

Ambient Fraction 1.0

Bump Amplitude 1.0

Diffuse Fraction 1.0

Glow Factor 0.0

Reflectivity 0.0

Index of Refraction 1.0

Specular Fraction 0.0

Transparency 0.0

First Smoothness 200

Second Smoothness 0

Second Weight % 0

Colors

□ Diffuse...

□ Specular...

☒ Diffuse colorizes image

Options

● Apply as color image

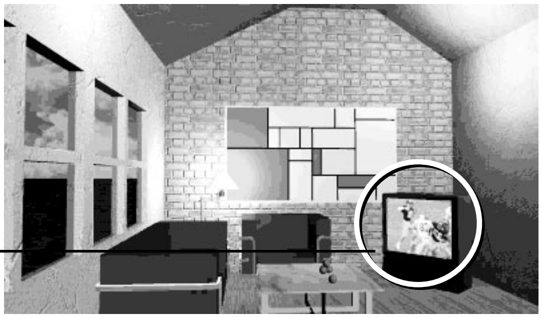
○ Apply as surface bump

○ Apply as both

Max image size 128

☒ Casts shadows

Uses for the RotoScoper may include playing digitized video clips on the screen of a television during a walk through of the room or applying a person's face onto a model of a human head, animating the facial gestures to smile, wink, talk, etc.



The RotoScope texture feature was used to "play" a football game on the television during an animation where a camera object "walks" through the room

If you find that the frame size is too small to fit the object, as indicated by a pixelized appearance to the animation when rendered, you can increase the maximum frame size field. This field defaults to 128 pixels square to conserve memory, but has no upper limit. This means larger frames of the animation are sampled down to a maximum of 128 pixels of whichever dimension is larger, the other dimension is also 128 pixels or smaller. If you increase the maximum frame size, you must reload the animation file into the Rotoscoper to force a re-sampling of the frames at the larger size.

Handling textures in a model

Textures created or loaded into a model become part of that model. Saving the model also saves its textures as part of the document. This is true whether or not a texture has been applied to any objects in the model space.

Creating new textures

As discussed at the beginning of this chapter, any number of textures can be created as part of a model. This is done by choosing New Texture from the Textures menu, which opens the surface texture editing dialog boxes. Holding down the Option key while choosing this command allows you to choose a texture extension upon which to create a solid texture instead. A full explanation of this command is found later in this chapter.

Loading textures

Textures may be loaded into a model as needed. It is an easy way to transfer work done previously in other projects to the current project. Once you have acquired a repertoire of texture files, either from commercial libraries or your own work, this feature allows you to assemble prebuilt building blocks into a complex model very quickly.

A distinction should be made between loading a texture and applying a texture. When a texture is loaded, it is given storage space in the model, but it is not applied. Applying a texture is the act of putting it on an object or shape instance.

As mentioned at the beginning of this chapter (and reiterated here for emphasis), the active view from which the texture is applied determines the orientation of the texture. This is especially important with regards to surface maps. If the orientation is not as desired, you can either rotate the texture on the object or re-apply it again in a different view orientation.

Surface textures with maps, as well as solid textures, can use the Change Mapping command to adjust how the texture fits on the object(s).

How to load

There are six ways to load a texture into an open model:

1. **Double-click on its preview image in the Textures palette.** This loads the texture only. No application or editing of the texture is attempted.
2. **Selecting a texture on the palette and clicking on the Apply button.** This loads and applies the texture to the selected object(s) in the active view.

3. **Selecting a texture on the palette and clicking on the Edit button.** This loads the texture and opens its dialog box for editing. No application is attempted.
4. **Clicking and dragging the preview image of a selected texture from the Textures palette to an object in the active view.** This loads and applies the texture into the active view. This is referred to as a "Drag and Drop" operation.
5. **Choosing the Load Texture command in the Textures menu.** This loads the texture only. It is similar to method #1, above.
6. **Choosing the Browse Textures command in the Textures menu and clicking on the Load button in its dialog.** This is a variation on methods #1 and #5 above.

The method you choose depends on what you want to do with that texture. The first four methods assume the texture to be loaded is represented on the Textures palette. The last two methods are for textures that are located somewhere other than in the Textures folder (within the Strata Vision3d application folder).

The loading of textures contained on the Textures palette is explained under the Textures palette discussion later in this chapter.

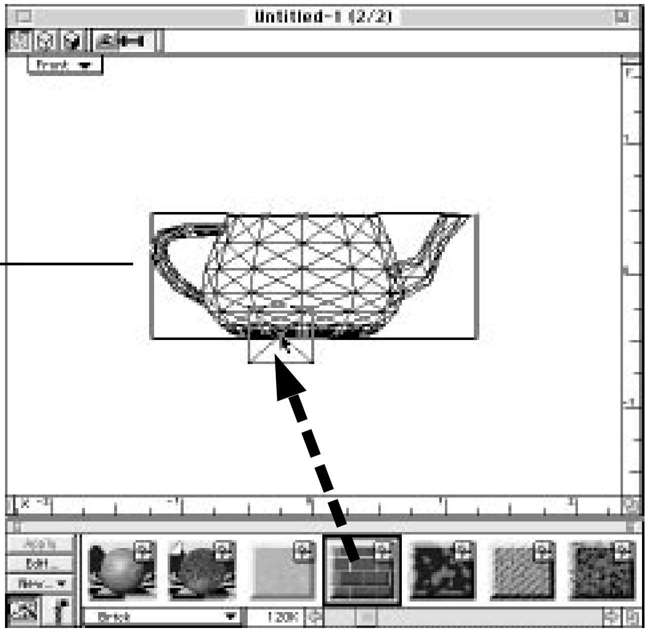
As mentioned, you can also load a texture into the active model from any source through the Load Texture command in the Textures menu. This feature is explained further under a discussion of the Load Texture command later in this chapter. In addition, the Browse Textures dialog box also contains a Load button. It functions exactly the same as the Load Texture command.

"Drag and drop" explained

As mentioned in method #4, you can click and drag a texture's preview image to an object in the modeling window to load and apply the texture. This is referred to as "drag & drop." This feature is available with any operating system. It is a feature of the Vision3d application, and does not rely on System 7 for support.

A small marquee appears representing the texture as it is being dragged, as shown in the illustration on the following page. As the texture marquee is positioned over an object, a marquee appears around it to indicate the connection is made. The object does not have to be previously selected. Upon releasing the mouse button, the texture is loaded into the model and applied from the direction of the active view on the object where you drop the marquee.

Textures palette “Drag & Drop”



Click and drag the chosen texture onto an object in the active view

The texture is automatically loaded into the active model and applied on the object where you drop its bounding box.

Note that the marquee you see during a drag and drop operation does NOT represent the size of the maps in the texture being inserted. You will still need to use the Change Mapping command to make adjustments in the texture’s mapping parameters, as with any other method of texture application.

Applying textures

Any texture can be applied to any object. The suitability of using a particular texture on a specific object is left for you to decide; you have full creative license. At first, you will probably use the textures provided with the software package. Later, you begin to explore making your own textures; experimenting with the surface property settings for different effects, then adding a color map or a bump map, as well as other advanced mapping techniques, as you gain experience.

The work of constructing the object and creating the texture culminate when the two are brought together. By this point, you have a pretty good idea what you want the rendered image to look like. It is truly an exciting time when your project all comes together and the rendered image looks just like what you saw in your mind.

Application methods

A texture can be applied to one or more selected objects at a time. There are three methods available to apply a texture:

- **Using the Set Selection command in the Textures menu**
- **Using the Apply button on the Textures palette**
- **Using the "Drag & Drop" feature, also on the Textures palette**

The Set Selection command can apply textures that are currently in the active model, either created as part of the document or loaded into it. The other methods work with any texture represented on the Textures palette, whether in the model or not. Applying a texture that is not in the model, but represented on the palette, will automatically load the texture.

These application methods are discussed further later in this chapter. The balance of this discussion is concerned with the other considerations in texture application, including application direction, mapping type, scale, and tiling. As you can see, there are several variables that allow you to "fine tune" the way a texture is applied to the object.

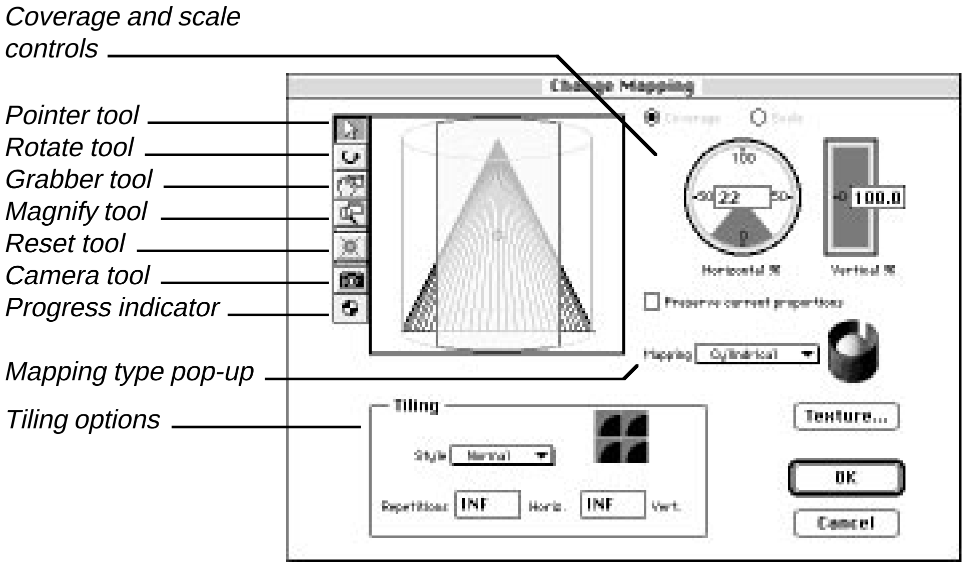
Previewing the work as you go

Preview renderings are a integral part of the texture application process. During the fine-tuning phase, you will probably want to take several "snapshots" to check the appearance of the surface maps on an object. Attention to mapping details now can eliminate rendering surprises later when you commit to the final image, only to find it didn't turn out right when you return the next morning.

Of course, you will want these preview snapshots to render as quickly as possible. You will want to turn off all non-essential rendering effects, disable anti-aliasing, and perhaps even lower the sampling rate for surface maps. This can be done manually or you can use the Draft and Good settings in the Stills sub-menu of the Tool Palette's pull-down menu. These are pre-defined rendering settings that control all the variables automatically. Rendering effects and options are discussed in detail in *Chapter 8: Rendering*.

Change Mapping dialog

Applying the texture is only the beginning. The way a surface texture is wrapped onto an object is determined first by the direction from which it was applied and second by the settings in the Change Mapping command. The change mapping dialog is shown on the following page, with its controls identified.



While a texture can be applied to more than one object, how that texture is applied to each of those objects can be customized in this dialog box. The Change Mapping command becomes active when an object is selected. However, the command may be dimmed for many reasons:

- **No object is selected**
- **More than one object is selected**
- **The object selected does not have a texture applied**
- **The selection is an instance of a shape and the texture was applied on a different level, thus it is not controlled at the current level**
- **The object selected has a texture applied (at this level), but does not require mapping control**

NOTE: *A distinction must be made between a texture using global surface property settings ONLY and a texture that contains surface maps. If there are no surface maps in the surface texture, it does not matter from what direction the texture is applied, or the scale, since its properties are the same across the entire surface. Thus, for textures without surface maps, the Change Mapping command is not active when the object with that texture is selected. If the texture has at least one surface map, then the Change Mapping command becomes active so you can control the position and scale of the map(s).*

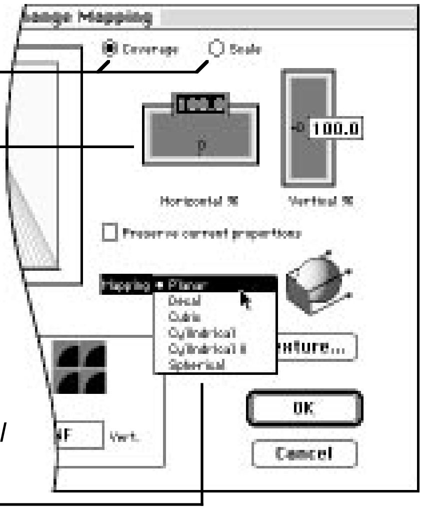
As mentioned earlier, solid textures also have mapping considerations, but do not require all of the controls in the Change Mapping dialog. This is because mapping type and tiling style are not an issue with volumetric/procedural maps. However, scale and position are important, so the Change Mapping command is available for limited functions on solid textures.

Coverage vs. Scale

The Change Mapping dialog is a full-featured control panel for setting all the variables required to "fit" a surface texture to an object. The most important part of this dialog is the Mapping Preview area. This preview allows you to interactively position, rotate, and scale the applied texture to fit the selected object.

The selected object appears in wireframe form in the background with the same orientation that was used in the active view to apply the texture. A red marquee appears over the object as a representation of the surface map(s) contained in the applied texture.

The scale of the texture marquee in the mapping preview indicates its actual size relative to the object. This is important visual information as you decide how to wrap a texture onto an object, and at what scale.



Coverage/scale radio buttons

Interactive coverage/ scale controls

(note Horizontal control changes from circle to rectangle when in coverage/ scale mode).

Coverage and scale modes are available only when Planar, Decal or Cubic mapping types are chosen.

Note the two radio buttons, title "Coverage" and "Scale," at the top right of the dialog box. Depending on which you choose, you may set the size of the texture as a percentage of object coverage or as a percentage of actual size. In many cases, the difference between the two modes is minor. But if you have taken the time to work out the exact size you want, the scale mode may be preferable.

Using the tools in the change Mapping dialog

Tools are provided to assist you in the interactive fitting of the texture in the mapping preview area. They operate as follows:

- **Pointer tool.** Use this tool to move the texture relative to the object. This is the same functionality as moving the texture origin point in a model window. The center of the texture is determined by the texture origin point. Conversely, moving the texture here in the Change Mapping dialog also moves the texture origin point in the model window.

  Double-clicking on the tool re-centers the texture on the object.
- **Rotate tool.** Use this tool to rotate the texture on the object. You may rotate it on any axis to change the texture's orientation from the one defined by the active view when the texture was applied to the object. This is the same functionality as using the Rotate tool with the Control key down in a model window. The object does not rotate, but rather the texture rotates on the object.

  Double-clicking on this tool resets the rotation back to the orientation defined when the texture was applied in the active view of the model window, whatever that may have been.
- **Grabber tool.** Use this tool to see other portions of the object in the preview. It scrolls the mapping area vertically, horizontally, or diagonally. Normally, the entire object is visible within the mapping preview area. But when the Magnify tool is used to zoom in on an object, you may need to move around to see how the texture is fitting.

  Double-clicking on this tool re-centers the object in the preview area.
- **Magnify tool.** Use this tool to zoom in or out in the mapping preview area. This is useful for checking exact placement of a texture on an object. Each click magnifies the preview area 2X. With the Option key down, each click reduces the view scale 2X.

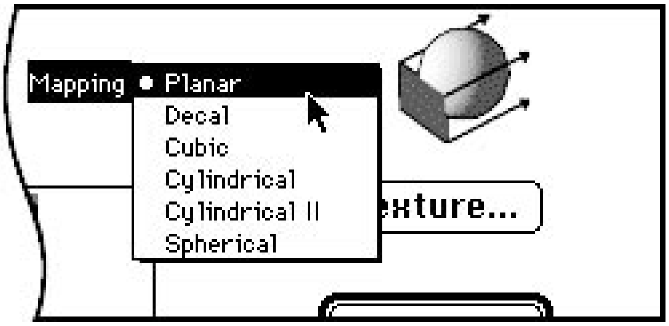
  Double-clicking on this tool fits the entire object and texture in the preview area again.
- **Reset tool.** Use this tool to re-set all texture transformations back to the default settings. The texture is re-centered, un-rotated, unscaled, etc. This is useful when you wish to start over in setting the mapping parameters.
- **Camera tool.** Additionally, you can render a preview image of the texture in this preview area by clicking on the Camera tool. To initiate a preview rendering, click once on the camera tool.

  You do not have to wait for the preview rendering to complete before making changes. If you make a change, just click on the Camera tool again to start the preview rendering over again using the new settings. To cancel a preview rendering, just select any other tool. The preview will return to the wireframe display.

Repetitions, scale and tile settings

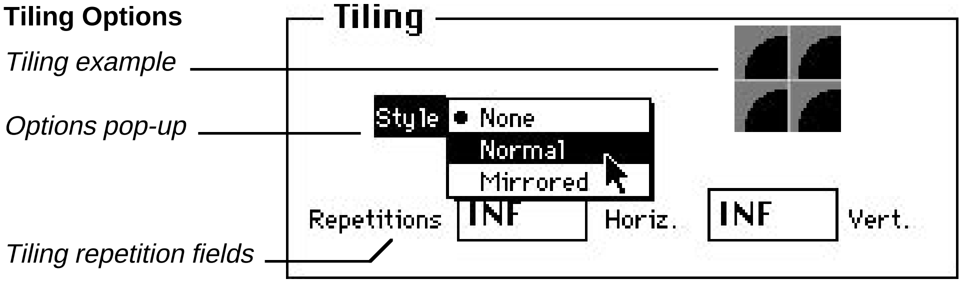
As mentioned, mapping type is set independently for each object. That way, one texture can be customized for use with many different objects. The Change Mapping dialog is accessed through the Textures menu. To use this command, you must first select an object that has a texture currently applied.

Choose the best mapping type for the object.



In the dialog, the "Mapping " pop-up menu presents different ways surface maps may be applied to an object. Typically, you would choose the pattern that would most closely match the geometric shape of the target object(s).

Tiling Options



Scale and tiling are surface texture issues because surface maps are finite in size. You also want to anticipate whether the tiling pattern should be normal or mirrored across the surface. If you choose normal tiling, it is important that the maps match appropriately at the seam, otherwise the tiling will be obvious. Mirrored tiling can minimize a tiling problem when the pattern doesn't match.

Included in the Change Mapping dialog are fields used to size an applied texture's surface maps, to set tiling characteristics, and to designate the number of surface map repetitions if tiling is enabled.

In combination with the scale and coverage settings, two different tiling patterns are provided. You may also select no tiling. If no tiling is chosen, the surface is mapped once only at the indicated scale. The map is centered on the object's texture origin point, facing the view from which it was applied.

Normal tiling repeats a surface map exactly as it is oriented for the number of times specified. Mirrored tiling alternates a map end-for-end across the surface, thus minimizing the visibility of a tiling seam.

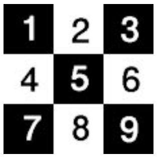
Mapping variations

Here are some examples of how these Change Mapping settings affect the way a surface map appears on an object. The surface map used was a simple checkerboard pattern.

The texture containing this surface map was then applied once to the object in the examples from the active Front view. The variations shown came from changes in the Change Mapping dialog fields only.

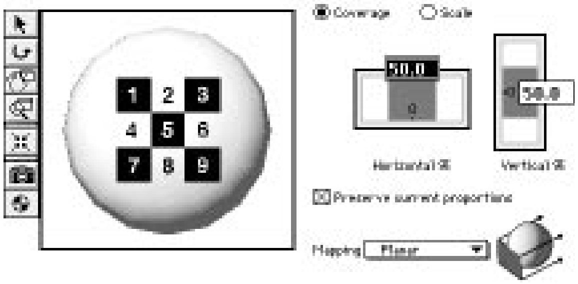
These mapping examples are just some of the possibilities. You may notice that no example of mirrored tiling is presented. With this pattern, the mirrored tile setting looks identical to the normally tiled pattern, except that the numbers are reversed.

This pattern was used as an image map for the mapping examples presented below



Example #1

Planar mapping, no tiling, 50% coverage


● Coverage ○ Scale

50.0

50.0

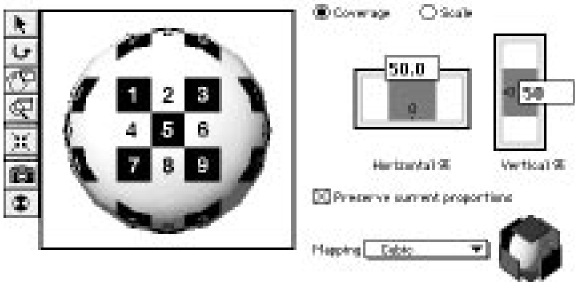
Horizontal % Vertical %

☒ Preserve current proportions

Mapping Planar

Example #2

Cubic mapping, normal tiling, 50% coverage, 1 repetition horizontally and vertically


● Coverage ○ Scale

50.0

50

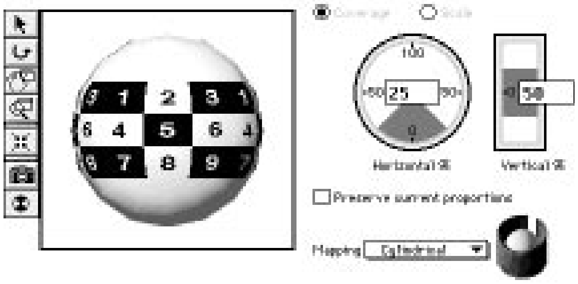
Horizontal % Vertical %

☒ Preserve current proportions

Mapping Cubic

Example #3

Cylindrical mapping, normal tiling, 50% vertical and 25% horizontal coverage, Infinite repetitions horizontally and 1 vertically



● Coverage ○ Scale

Horizontal % 25

Vertical % 50

□ Preserve current proportions

Mapping Cylindrical

You may want to experiment further on this theme to gain a better understanding of how the Change Mapping dialog settings control the way textures can be mapped on an object. You should create a distinctive pattern such as the one used here, possibly including numbers in each of the squares. Adding numbers can tell you a lot about the positioning and pattern repetition of surface maps.

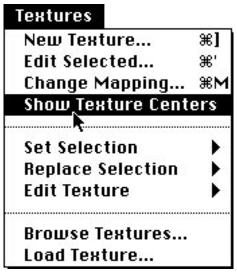
Texture transformation

In addition to setting mapping type, scale and tiling parameters, the Change Mapping dialog allows you to control the position of the texture and its orientation.

The mapping origin point

Each object has two origin points — one for the object and another for the texture. The object origin's location is represented by a small white dot; the texture origin by a small diamond. The object origin point appears whenever an object is selected. This is explained in greater detail in *Chapter 3: Modeling Objects*.

The texture origin point only shows if the "Show Texture Centers" command in the Textures menu is enabled AND a texture has been applied to the object. Note that the Show Texture Centers command defaults to off at the beginning of each session.



Textures use the texture origin point to position surface maps on an object. The center of the map or multiple maps contained within a texture is represented by an object's texture origin point.

A texture origin point may be moved in any direction, and even positioned outside of the object itself, if desired. Moving an object's texture origin point moves the applied surface map(s) with it.

The texture origin point automatically follows the geometric center of the object during a Scale tool operation, unless the Command key is held down. Re-scaling objects automatically scales all currently-applied surface map(s) with it. However, re-applying the texture reverts the surface map(s) to original size in relationship to the re-sized object.

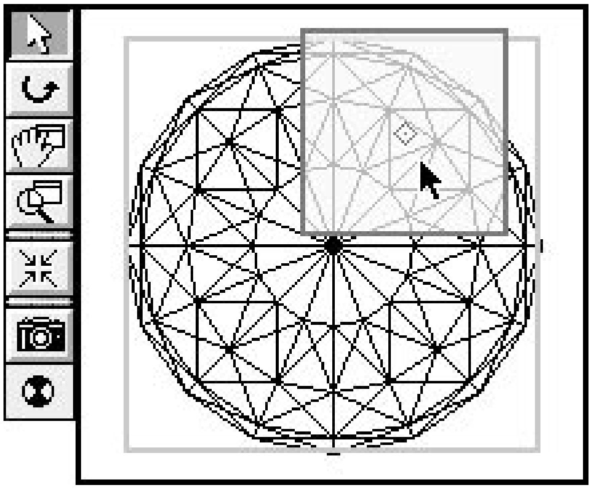
Holding the Control key down when using the scale tool causes the applied surface map(s) to be scaled instead of the object. When holding the Control key down, all other scale tool modifier keys also apply to scaling surface maps, just as they normally would for scaling an object.

Moving the texture

As implied, the texture may be moved or re-positioned on an object so that the center of the surface map(s) is not at the geometric center of the object. This can actually be done in two ways:

1. **Click-drag the texture marquee in the Change Mapping dialog.**
2. **Move the texture origin point in the model.**

Moving the texture marquee relative to the object in the mapping preview is usually the preferred way because it is interactive and allows you to see the effects immediately. The texture may be rendered in the mapping preview as you are positioning it.



This texture movement may be identified when you return to the model and show the texture origin point. You will see it is shifted from the geometric center of the object.

However, moving the texture origin point directly is sometimes necessary because the mapping preview allows you to work in only two dimensions. Thus, if you want to shift the texture's position on the Z axis (relative to the mapping preview image), you must return to the model and use a perpendicular view to drag the origin point at right angles to the active view from which the texture was applied.

To move the texture origin point relative to the object:

1. **Select one of the pointer tools.** Any of the three object manipulation tools can select and move origin points.
2. **Position the cursor over the object origin point.** If the selected object has a texture applied, you will see two origin points, one behind the other. If no texture has been applied to the object, only the object origin point is visible.
3. **Hold down the Command key.** Without the Command key, dragging the origin point(s) also moves the object.
4. **Drag it to the new position.** The two origin points will move as one.

NOTE: *To separate the two points, hold down the Control key while dragging either point. If both origin points are visible, the object origin point is in front and may be moved by simply clicking and dragging the dot. The diamond-shaped texture origin point always appears behind the object origin point when they are together, but may be selected by clicking and dragging one of the visible diamond points. With the Control key down, the chosen origin point moves while leaving the other point, as well as the object itself, stationary.*

Modifier keys that apply to moving object and texture origin points:

A. **Shift key** — Constrains the cursor to drag the origin points horizontally or vertically only. Unconstrained, the cursor can move diagonally.

B. **Command key** — Moves the origin points (both object and texture, if visible) without moving the object. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

C. **Control key** — Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other as well. With the Control key down, either the object origin point or texture origin point (if visible) can be moved with moving the other point or the object itself.

Rotating maps on objects

From time to time, it may be desirable to re-orient surface maps on an object. For example, you may want a wood grain texture to run a different direction, or prefer a specific portion of a map to be more visible.

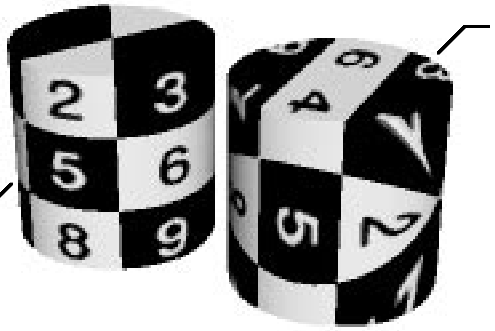
Surface maps may be easily rotated on the object in two ways:

1. **Choose the rotate tool in the Change Mapping dialog and rotate the texture in the mapping preview.** Again, this is usually the preferred method because it allows you to see the results interactively.
2. **Using the object rotate tool in the model window, select the object and hold down the Control key while performing a rotation operation.** This rotates the surface map around the object, but leaves the object in the same position. The selection handles work the same for rotating surface maps as for rotating the object.

   Note that rotation of the texture is around the texture origin point, rather than the object origin point for normal object rotation.

   When holding the Control key down, all other rotate tool modifier keys also apply to rotating surface maps, just as they normally would for rotating an object. You should be aware that this also changes the direction a surface is mapped to the object, as shown in the following illustration.

Rotating an image map also rotates the mapping type orientation as well


Original Cylindrical map
Same cylindrical map with texture rotated 90° clockwise in Front view

Surface maps may be rotated on any axis. The texture does not need to be re-applied for the rotation to take effect. In fact, re-applying the texture causes the surface map(s) and mapping type to revert to the orientation of the active view again.

If you desire to rotate a surface map, but not change the mapping type orientation, you must extract the surface map from the texture definition, via the Cut/Copy/Paste editing commands, then use a paint or drawing program (one that supports the desired color depth of the image) to adjust the map's orientation. You may then re-insert it into the texture.

How to make maps to correct scale and resolution

Ever-present memory limitations dictate that surface maps be kept as small as possible. Yet, they must also be large enough not to appear pixelated when sized to fit the target object. But how big is enough? What other factors must be considered when deciding how large or what resolution is required to maintain the surface maps' quality when rendered?

In general, pixelization of surface maps can be avoided by keeping their scale at one (1.0000) or less. Yet surface maps can sometimes be too small to completely cover an object. Surface maps scaled larger than 1.0000 can often render with good results, but is not usually recommended. In those situations, you should repeat or tile the map across an object's surface rather than scale it larger if possible.

These are all considerations that must be planned ahead of time when constructing a surface map. Any number of full-color paint-type applications are available to aid in this process. The important thing to remember is not to over-do the resolution factor. Use what you need, but be aware that you have a finite amount of memory with which to model, create/load/apply textures, and of course render.

An example

There are several methods to calculate the resolution requirements of surface maps. Once you understand the concept, you may have a preference that is easier for you to visualize. The one presented here is for concept only; real projects are rarely this clean — mathematically speaking.

Suppose you are modeling a room in which a picture is hanging on the far wall. Except for the picture, all of the other textures are either simple colors or solid textures. The surface map to be used for the picture MUST be of sufficient resolution to render clean — without pixelization or "jaggies."

This is the focus of the example — how do you calculate the minimum resolution required of the image that is to be applied as a surface map to the picture.

Further suppose that the final image is to be rendered at 200 d.p.i. This is determined because you plan to it use as part of a flyer that will be printed with a 100 line screen. Thus, 200 d.p.i. is the least resolution that you need for the rendered image to print with any expectation of reproduction quality.

About image resolution

Why would a 100 line halftone screen dictate using a 200 d.p.i. resolution for the rendered image? For background, there are four types of resolution that are important to a digital image:

- **Color depth.** This is the measure of the number of bits of information per pixel. This resolution determines how much color information is stored in the pixels within the image. The common options for this type of resolution range from 1 bit per pixel (B&W) to 32 bits per pixel (16.7-million colors)
- **Dots per inch.** This is the measure of the amount of information in an image. The image resolution and its dimensions determine the overall size of the image. The greater the size, the more memory it requires and the more disk space it requires to store.
- **Halftone screen.** This is the measure of the density of the halftone ruling or frequency. Because the halftoning is arranged in lines, this resolution is measured in the number of lines per inch.
- **Device resolution.** This is the measurement of the number of dots per inch that the output device can produce (such as a laser printer at 300 d.p.i. and an imagesetter at up to 3300 d.p.i.). The higher the device's resolution, the more levels of gray it supports. The maximum levels of gray is 256.

It does not matter whether the image is scanned, captured from a video source, or rendered in Vision3d, every image you plan to output must be concerned with these resolution parameters. Of these, the color depth and dots per inch (image resolution) is the one you must be concerned with here.

Your image's resolution is a critical factor in determining the quality of the printed output. If the resolution is too low, the PostScript language in the output device will use a single pixel's color to create more than one halftone dot. This results in pixelization, or a very coarse-looking output. If the resolution is too high, you will send more information than the output device needs, taking longer to print than it should.

Determining the best resolution for the output

The best resolution for the printed output is dependent upon the quality of the output you want, as well as on the resolution of your printer. If you are printing to a high resolution device, you will need a higher image resolution than if you are printing to a low-resolution one. Low resolution devices support fewer levels of gray for a given halftone screen density. That means that fewer pixels are required to supply sufficient data to create the halftone dots, so you don't need as high a resolution.

There's one more point. Just because your trusty, rusty laser printer has 300 d.p.i. device resolution, that does not mean you can print to it with a 300-line halftone screen. In fact, the recommended halftone density for a 300 d.p.i. laser printer is only 53 l.p.i. And that only gives you 33 levels of gray for tonal gradations.

Digital halftone screen "spots" are made up of printer dots. The number of dots in each spot corresponds to the number of gray levels the spot can represent. The larger the spots, the more dots there are to represent the levels of gray, at a cost of a coarser screen. Finer (or denser) halftone screens create smaller spots composed of fewer dots. This may sharpen details, but reduces the numbers of grays that can be represented. This results in posterization, creating a stair-step or banding effect in moving from one shade to another in the image.

Here is the formula for calculating the relationship between gray levels, screen frequency and resolution:

$$\left(\begin{matrix}\text{Output}\\\text{Resolution}\end{matrix} \div \begin{matrix}\text{Screen}\\\text{Frequency}\end{matrix}\right)^2 + 1 = \begin{matrix}\text{Available}\\\text{gray levels}\end{matrix}$$

Imagine you are printing to a 300 d.p.i. laser printer, using a 60-line screen. That means there are 60 halftone spots per inch. Thus, since there are 300 dots per inch and each halftone spot is 1/60th of an inch, each spot contains 25 laser dots (in a 5 by 5 square). Conclusion: on a 300 d.p.i. laser printing, printing with a 60-line screen, there are 26 possible levels of grays. Having ONLY 26 levels of gray with to which to work results in noticeable posterization.

Choosing the optimal screen frequency

Output Device Resolution *(dots/inch)*	Screen Frequency *(lines/inch)*	Available levels of gray
300	53	33
300	75	17
300	100	10
600	53	129
600	75	65
600	100	37
1000	53	356
1000	75	179
1000	100	101
1200	65	341
1200	85	200
1200	100	145
2400	85	790
2400	100	577
2400	150	257

Okay, so now that you know the need for halftone screens and how to figure the levels of gray available, you can decide on the halftone density. But how does that relate to image resolution? That is the simplest part!

AS A GENERAL RULE, to produce sufficient quality for a given halftone density, the image's dots per inch resolution should be twice to two-and-one-half times the halftone screen density used.

Back to the picture example

Your first thought might be that the surface map for the picture must then also be 200 d.p.i., but that is not necessarily the case. If the picture in the model is to be rendered at actual size then, yes, the image for the map would have to be 200 d.p.i. But, if the apparent size of the picture is one-half its actual size in the rendered image, then you would only need a map of 100 d.p.i. If the picture is one-quarter actual size, the map would only need to be 50 d.p.i. Do you see the pattern?

Now look at it the other way. If the image you are going to use is 72 d.p.i., and the amount it is being reduced when rendered is known, what is its effective resolution? Just as reducing or enlarging an image in a page layout program increases or decreases that image's effective resolution, the final resolution for the map is based on its apparent size in the final rendered image.

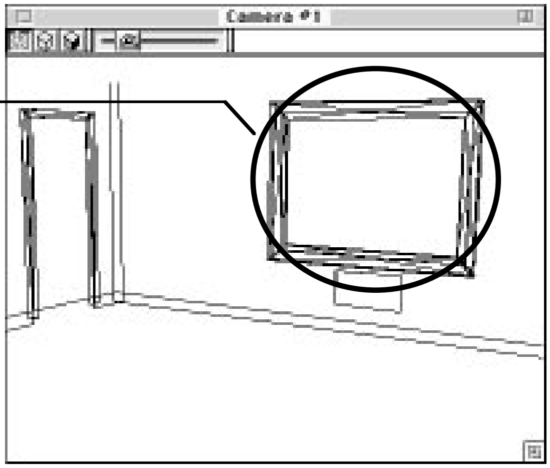
Note that none of this discussion is concerned with the scale at which the image is mapped to the object. That is a separate issue. It is the final rendered size compared to the original size.

Original image

Dimensions:
544 pixels wide by
333 high (actual size)

Rendering image

Dimensions of picture in image: 177 pixels wide by 136 high (apparent size, with some cropping of the original's width)



Effective resolution

The rendered image is .32 of original size. Thus, the apparent resolution of the original image in the rendered image is 225 dots per inch

In this example, the original image was 72 d.p.i, but its apparent size in the rendered image is just under one-third of the actual size. The formula to calculate the effective resolution is as follows:

ER = 72 dpi / .32 = 255 dpi

For MIAs (Mathematically Impaired Adults): To refresh your memory, when dealing with fractions, you convert to decimal format and divide. (You do know that .32 is the same as 32/100, don't you?) This calculates out to 225 d.p.i. as the effective resolution of the picture…more than enough for the 200 d.p.i. needed for the rendering. Obviously, using a higher resolution map for this project would have been excessive and taken more memory than necessary.

Of course, things get more demanding if the rendered image is going to be larger than actual size, such as for use as a poster. If a surface map will be rendered at 1.5X its actual size, then the resolution of that map must be 1.5X greater than the resolution of the rendered image.

Whether larger or smaller, the question is the same: *WHAT IS THE SURFACE MAP'S ACTUAL SIZE COMPARED TO ITS APPARENT SIZE IN THE RENDERED IMAGE?*

Map resolution in animations

Determining the resolution requirements of a surface map is relatively straight forward for still images. The size of the map in the finished image will be fixed, so you can work backward to figure the map's minimum requirements. However, in the case of animations, those requirements can get rather more difficult to calculate.

The general rule here is to determine the largest each object in an animation will appear in the rendered frames, then use that as the basis for establishing the minimum resolution of the maps that may be used on the object. Yes, it does take more time to lock down these additional details, but the results are worth it if you can avoid a map becoming pixelized during the critical moment as it swings closely past the view plane at larger-than-normal size.

On the other hand, the eye is more forgiving of detail when an object is moving than when it is still, so you can often get away with lesser resolution than the formulas would suggest. Again, you should plan the surface mapping to factor in sufficient resolution, but avoid overkill that gobbles up memory you may need for something else.

Re-sampling the pixels in surface maps

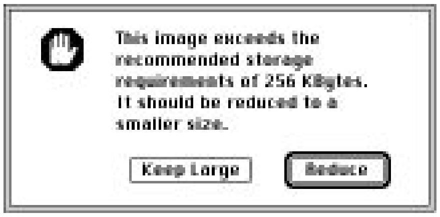
Under System 7.x, PICT images with resolutions greater than 72 d.p.i. may be used as surface maps without stripping away the "excess" data. When an image with higher than 72 d.p.i. resolution is loaded, the resulting map is re-sampled to 72 d.p.i. and made correspondingly larger, rather than ignoring the extra data. Thus, a 3 inch by 3 inch image at 144 d.p.i. (with a size of 547K) becomes a 6 inch by 6 inch map at 72 d.p.i. upon loading. It contains the same number of pixels and uses the same amount of memory in both cases.

The size of such a higher-resolution map must be adjusted to fit the object it was intended for. Thus, if you have a rectangle that is 3″ by 3″ square, and the 3″ by 3″/144 d.p.i. map becomes 6″ by 6″/72 d.p.i. upon loading into Vision3d, you know that the scale setting in the Change Mapping dialog must be 50% to make it fit the 3″ by 3″ object. And the map's effective resolution is again 144 d.p.i. (If in the rendered image, the 3″ by 3″ object actually appears only 2″ by 2″, then the effective resolution of the map becomes 216 d.p.i.)

Alternately, you could just switch the Change Mapping dialog to "coverage" mode and simply set the values to 100%. That automatically reduces the map's size to fit the object. The choice of modes — coverage or scale — depends on how you want to visualize the mapping process

How big can a map be?

Loading a higher-resolution image for a surface map often results in a very large map that must then be scaled smaller to fit the designated object, thereby obtaining the desired increase in effective resolution. The rendered image often looks much sharper because of the map's tighter pixels.



This image exceeds the recommended storage requirements of 256 KBytes. It should be reduced to a smaller size.

While some projects require higher-resolution maps to improve image quality in the finished project, high resolution images also require a great deal of memory. Memory you may not have or need to use in some other way. Experience indicates that the vast majority of maps can be less than 256K and perform their function quite well.

To that end, when you load a map into Vision3d that is larger than 256K (through any combination of size and color depth), you will see an alert asking if you wish to reduce the map or keep it full size. Note that this limit is advisory and you may keep the image full size, if needed. The alert is presented so you are aware of the need to be economical with the memory requirements. If you choose to reduce the map, Vision3d maintains the aspect ratio during reduction and will shrink it just enough to come under the 256K limit.

Alternately, you may find that reducing the color depth (from 32-bit to 8-bit, for example) can sufficiently lower the memory requirement without shrinking the dimensions.

The size requirements of a map depend on many variables — i.e., scale of the object, scale of the map, tiling style, details in the map, actual size of the final rendered image, etc. The optimal situation is to use a map slightly bigger than required, then apply a small scale reduction through the Change Mapping dialog box in the Textures menu. This helps insure no pixelization.

Generally, you should avoid sizing a map larger than 100.0 in the Scale mode of the Change Mapping dialog once it has been applied to an object, since that can induce pixelization of the map, even when its apparent size in the final image is smaller. Maps are scaled using the Change Mapping command in the Textures menu. A scale of 100.0 means 100% of actual size. If you find you are always having to scale up maps, you should consider increasing their size…but just do it sparingly.

NOTE: There is a maximum size limitation imposed on Vision3d by the operating system, via the QuickDraw imaging model used in every Macintosh. This applies to images used for surface maps, as well as the rendered product itself. At this writing, the limit is 4,000 pixels on the horizontal dimension.

How much memory will it take?

If you plan to insert any textures from a commercial library, or create your own textures containing surface maps, use the following formulas to determine the memory requirements of each PICT file you include in a texture:

- **Color maps.** Multiply the height (in pixels) by the width (in pixels), then multiply that number by four. The result is the number of bytes required to store a 32-bit image as a color map in memory. Color maps stored in the model in 8-bit format are one-quarter the 32-bit version, or the same as grayscale maps.
- **Grayscale maps.** Multiply only the height by the width (in pixels). The result is the number of bytes required to store the image as a gray-scale map. Grayscale maps are stored as 8-bit images.

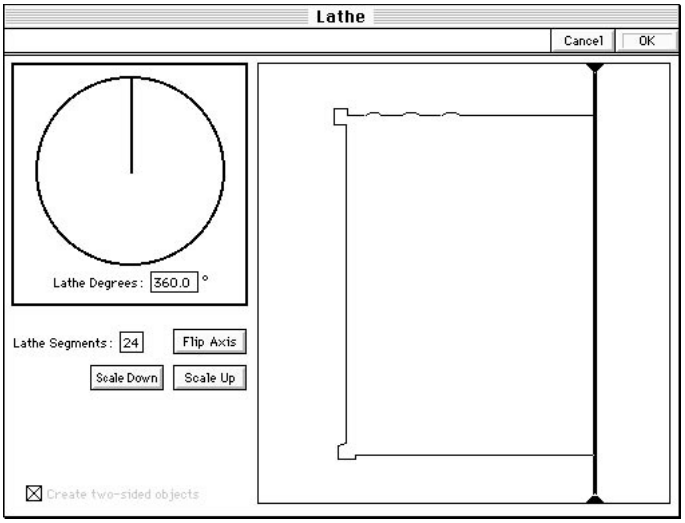
Quick Tutorials

The object of these quick tutorials is to give you a basic understanding of how textures work in Vision3d. You will learn how to load a texture, how to create a texture, how to apply textures, and how surface maps work. You'll also learn how to compose a texture with multiple surface properties using composite maps.

Exercise 1: Label on a can

In Exercise 1, you will learn how to use a PICT image to make a simple surface texture (a label) and how to apply that texture to a cylindrical object in a Vision3d model. You'll also learn how to adjust both the size of the object and the mapping parameters to make sure the texture fits correctly on the rendered object.

1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Lathe a can-object.** Choose the polyline tool as you have used in previous tutorial exercises of other chapters . Using the same techniques as in these previous exercises, draw an outline similar to the one shown in the following illustration.



Lathe Degrees: 360.0 °

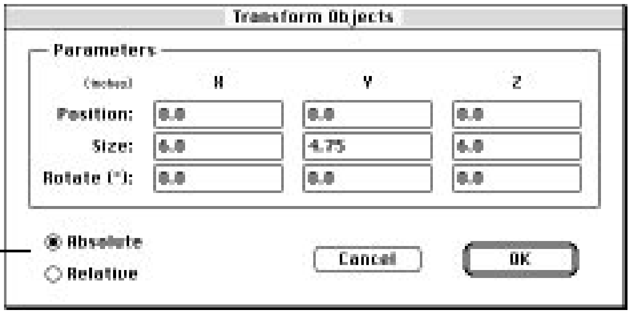
Lathe Segments: 24

☒ Create two-sided objects

Don't worry about getting the size right. You can adjust that later. Once the outline is drawn, choose the Lathe modeler. Match the axis position and settings shown and click on the OK button. Vision3d draws the can in the model window.

3. **Adjust the size of the object.** With the 3-D can object selected, pull down the modeling menu and choose the Transform command. Vision3d opens the Transform Objects dialog box.

   Make sure you are working in Absolute mode, by clicking on the Absolute radio button in the lower left of the dialog. Match the numerical values in each field with those in the following illustration and click on the OK button. Vision3d redraws the can in the model window using the precise values you set in the Transform Objects dialog window.



Transform Objects

Parameters

(inches) X Y Z

Position: 0.0 0.0 0.0

Size: 6.0 4.75 6.0

Rotate (°): 0.0 0.0 0.0

● Absolute

○ Relative

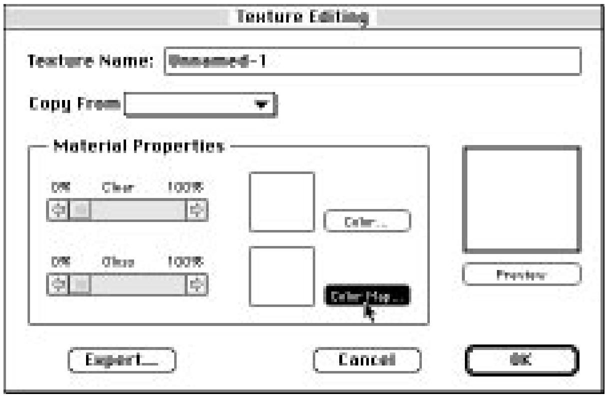
Set the Position numbers to zero (0), Size to 6, 4.75, 6, and rotation to zero (0).

Make sure that Absolute is selected

The Transform command gives you another method to scale, position, and rotate objects on all three axes. You already know you can perform these operations interactively using the click-and-drag method with tools on the Tool palette, as you have done in earlier tutorial exercises.

Or you can use the Transform command and input numerical values for more precise control of object position, size, and rotation. In this exercise, you need to know the precise dimensions of your can object so you can ensure the label's proportions will be correct and the texture will wrap around it in realistic way when the object is rendered.

4. **Create a new texture to place on the can.** Pull down the Textures menu and choose the New Texture command. Vision3d opens the Texture Editing dialog for a surface-type texture.



Texture Editing

Texture Name: Unnamed-1

Copy From

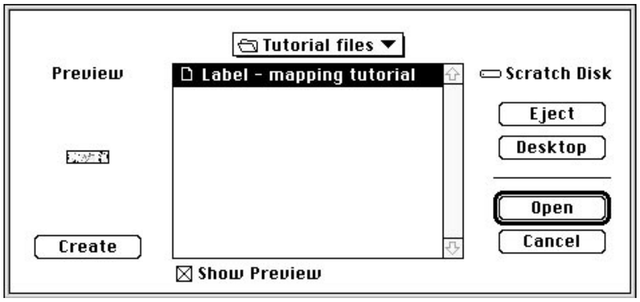
Material Properties

0% Clear 100%

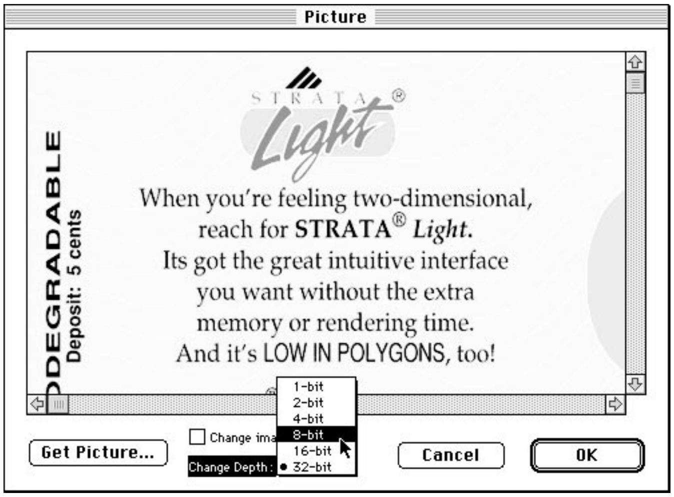
0% Gloss 100%

5. **Click on the Color Map button inside the Material Properties box.** Vision3d opens the Picture dialog box.

6. **Click on the Get Picture button within the Picture dialog.** This opens the Load File directory dialog, which allows you to navigate to the location of the PICT file you want to use as a texture.



7. **Find the "Label - mapping tutorial" in the Tutorials folder.** Choose this file and click on the OK button. Vision3d opens the PICT in the Picture dialog window.

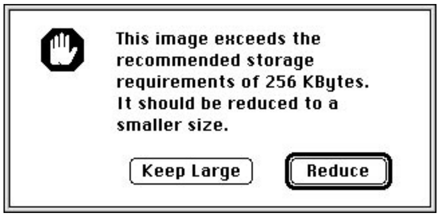


When you're feeling two-dimensional, reach for STRATA® *Light*. Its got the great intuitive interface you want without the extra memory or rendering time. And it's LOW IN POLYGONS, too!

8. **Switch the Change Depth field to 8-bit and click on the OK button.** This accepts the surface map at the 8-bit color depth. The surface map that you have loaded was saved in 8-bit color originally. But all color maps are loaded into the model in 32-bit color by default. Click on the pop-up menu to change the color depth of the surface from 32-bit to 8-bit color, which requires one-quarter the amount of memory.

9. **Click on the Keep Large button.** Since the file size of the image is greater than 256K, Vision3d notifies you the color map is larger than

the recommended size limit. You are given the option to reduce the dimensions of the color map or keep it large – that is, at its original dimensions.



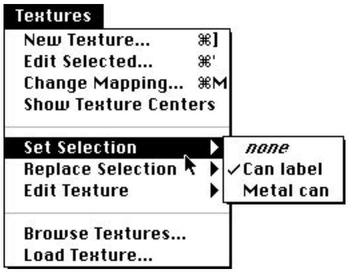
This image exceeds the recommended storage requirements of 256 KBytes. It should be reduced to a smaller size.

This prompt appears only when the surface map file exceeds the recommended color map file size limit of 256K. Because this map was prepared so it would fit on the can at original size, you want to keep it that size. The surface map is placed into the color map field in the Texture Editing dialog window at the size it was created.

10. **Name the texture.** Type "Can Label" in the Texture Name field of the Texture Editing dialog. Then click on the OK button. Vision3d returns you to the modeling window and renders a preview of the texture on a tile in the Textures palette.

11. **Make sure the Can Label texture is applied to the can-object.** If an object in the model window is selected when you create a texture, the texture is automatically applied to that object.

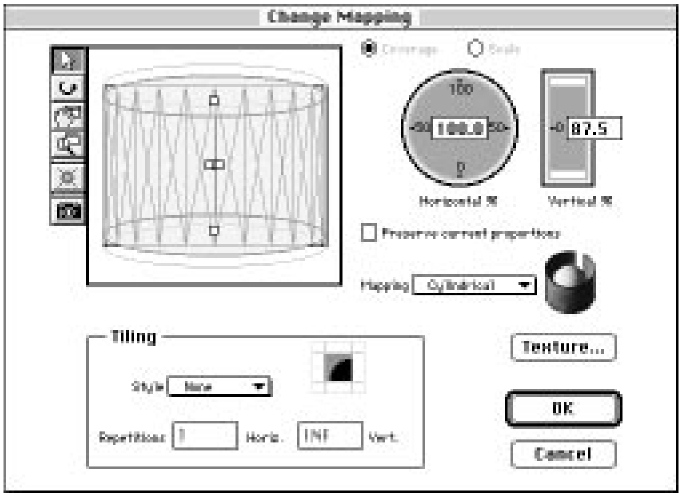
    You can check to see if the Can Label texture is applied to the can-object by pulling down the Textures menu and choosing the Set Selection command. If you see "Can Label" with a check mark in front of it in the Set Selection sub-menu, then the texture has been applied to the can.



If the check mark is not present, then you need to apply the texture to the can-object as follows: A) Make sure the active view is the Front view, B) In the model window select the can object, C) Select the Can Label texture preview on the Textures palette, D) Click on the Apply button on the left side of the Textures palette.

Use the Set Selection command as described above just to verify the texture is applied to the can.

12. **Adjust the texture to fit the can object, then click on the OK button.** Pull down the Textures menu and choose the Change Mapping command. Vision3d opens the Change Mapping dialog window. In the Mapping Preview box, you see a representation of the texture overlaid on a representation of the can object.



Change Mapping

● Coverage ○ Scale

Horizontal % 100.0

Vertical % 87.5

□ Preserve current proportions

Mapping Cylindrical

Tiling

Style None

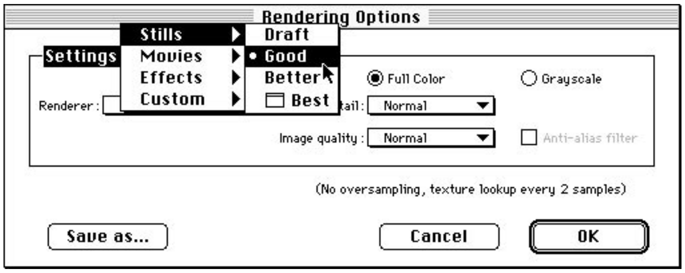
Repetitions 1 Horiz. INF Vert.

Set Tiling Style to None, and Mapping to Cylindrical, as shown. Make sure the Preserve Current Proportions checkbox is NOT checked, then match the Horizontal and Vertical percentages shown here. If necessary, center the texture overlay so it aligns with the object representation in the Mapping Preview box, as shown. This can be done easiest by double-clicking on the pointer tool.

Clicking on the OK button puts you back in the model window.

13 **Set rendering quality.** To see whether our label texture has been placed correctly on the can object, you must render the model. The first step is to set the quality of the rendering. Pull down the Rendering menu and choose the Rendering Options command.

Vision3d opens the Rendering Options dialog box. Pull down the Settings menu and choose “Stills.” Next, choose “Good” from the sub-menu. Clicking on the OK button returns you to the model window ready to begin a rendering.



Rendering Options

Settings

Renderer :

● Full Color ○ Grayscale

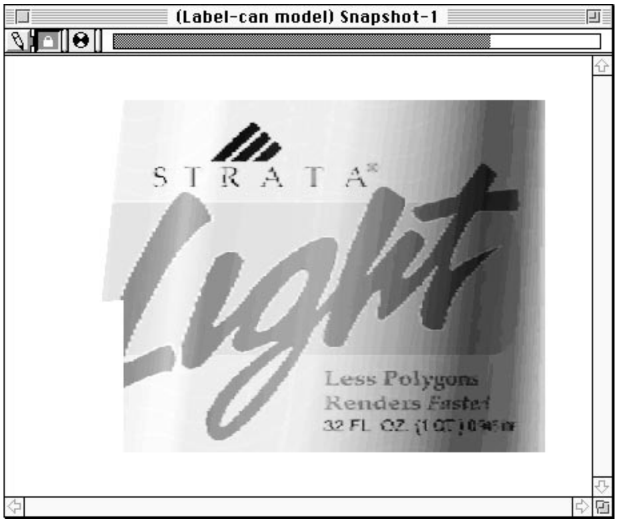
tail : Normal

Image quality : Normal

□ Anti-alias filter

(No oversampling, texture lookup every 2 samples)

14. **Render the model.** Choose the large camera icon from the bottom of the Tool Palette. Notice when you move the cursor into the model window it changes to a camera, indicating the rendering tool is active. With the camera-cursor in the model window, click once. Vision3d now begins to render everything in the model window.



(Label-can model) Snapshot-1

Your finished rendering should look similar to the one shown here. If desired, you can try rendering different variations of this model. For example, you might add perspective by changing the three-position lens control to Normal, or change the view angle by choosing another one from the View pull-down menu.

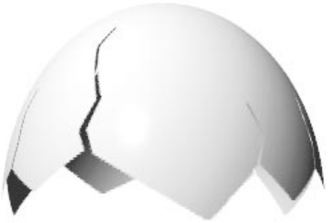
In this example, rendering different views will reveal previously unseen areas of the label. You may also want to experiment with other rendering settings in the Rendering Options dialog.

When rendered, you should see an object similar to the one shown on the next page.

The issues of rendering and the many options available in that phase of a project is as involved as the other phases. Thus, rendering is discussed in greater detail in *Chapter 8: Rendering*.

Exercise 2: Cracked eggshell

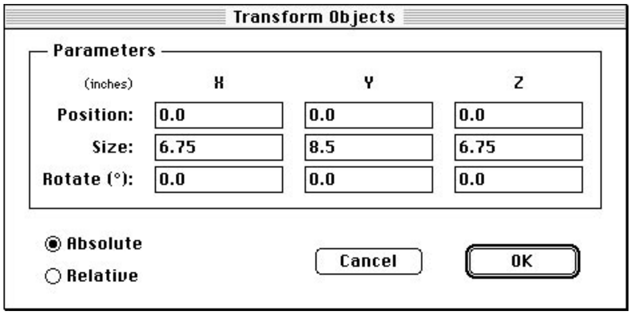
In this exercise, you'll learn how to create a more complex texture. You will place several simple PICT files (like those created by a paint program, for example) into Vision3d's expert texture mapping fields. These maps, in conjunction with the expert surface property fields, will form a texture resembling an egg shell.



You'll apply this texture to a simple sphere primitive and render it, creating a finished image of a broken egg shell. You'll also learn how to use a background environment as part of a rendering.

1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Draw a sphere in the model window.** Choose the sphere primitive from the Tool palette. Click once in the model window. Vision3d draws a sphere in the model window.

3. **Re-size the sphere.** Match the values shown in the following illustration of the Transform dialog box. Basically, you are stretching the sphere vertically so that becomes more egg-like in geometry.

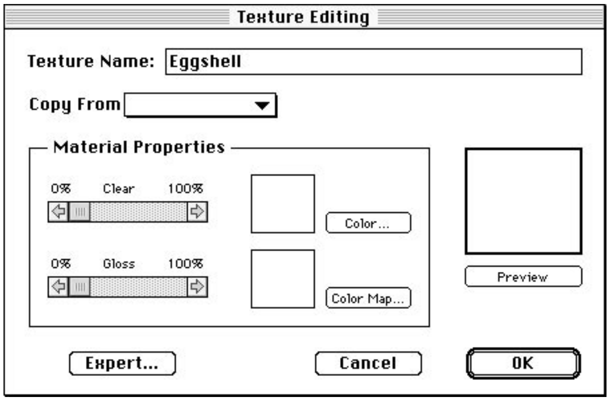


(inches)	X	Y	Z
Position:	0.0	0.0	0.0
Size:	6.75	8.5	6.75
Rotate (°):	0.0	0.0	0.0

● Absolute
○ Relative

If the re-scaled sphere is too large to fit in the model window, use the "Fit Views to All" command in the Windows menu. You may also want to look at the sphere from the Top view or a side view. However, if you change the view angle, be sure to return to the Front view before proceeding.

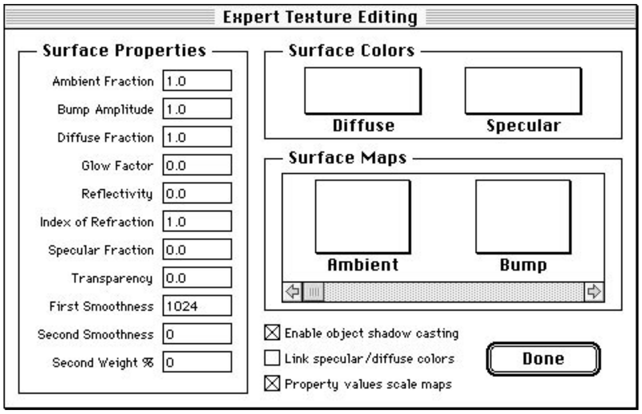
4. **Create a new texture.** Use the New Texture command in the Textures menu as you did in Exercise 1 of this chapter. The texture editing dialog box for creating a new surface-type textures opens. The first thing to do is name the new texture, "Eggshell."



Texture Name: Eggshell

Alternately, you can click on the New button in the Textures palette. This performs the same function.

5. **Click on the Expert button.** This is found in the lower left of the Texture Editing dialog. Vision3d displays the Expert-level Texture Editing dialog box.



Expert Texture Editing

Surface Properties

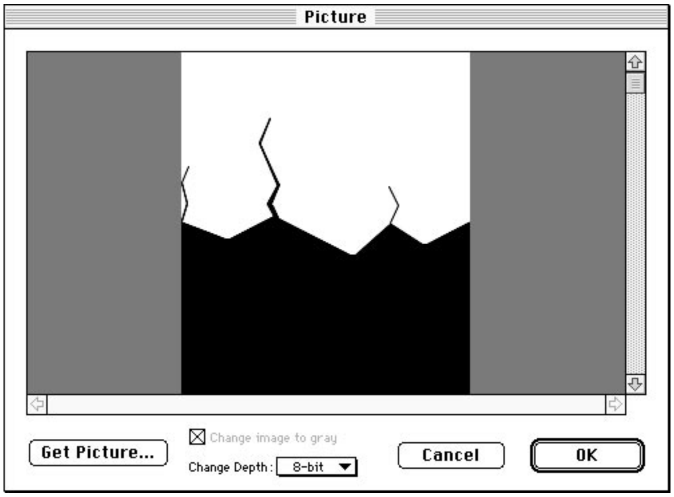
Ambient Fraction	1.0
Bump Amplitude	1.0
Diffuse Fraction	1.0
Glow Factor	0.0
Reflectivity	0.0
Index of Refraction	1.0
Specular Fraction	0.0
Transparency	0.0
First Smoothness	1024
Second Smoothness	0
Second Weight %	0

Surface Colors: Diffuse, Specular

Surface Maps: Ambient, Bump

☒ Enable object shadow casting
☐ Link specular/diffuse colors
☒ Property values scale maps

6. **Click once inside the Ambient field within the Surface Maps area of this dialog.** Vision3d responds by displaying the Picture dialog window. Through this dialog you are going to get the PICT image that will be placed in the Ambient surface mapping field.
7. **Click on the Get Picture button.** The standard file directory dialog opens. This dialog allows you to find and open the PICT files you need.
8. **Load the "Eggshell - amb/diff/spec" file.** This is in the Tutorial folder within the Vision3d application folder. This is a grayscale PICT.



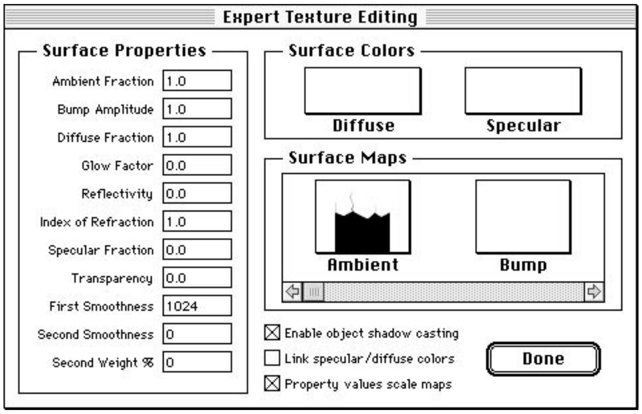
Picture

☒ Change image to gray

Change Depth: 8-bit

Vision3d loads the PICT and inserts it into the Eggshell texture as the Ambient surface map.

9. **Click on the OK button.** Vision3d returns you to the Expert Texture Editing dialog. You can see that the Eggshell - amb/diff/spec PICT has been placed in the Ambient field.



Expert Texture Editing

Surface Properties

Ambient Fraction	1.0
Bump Amplitude	1.0
Diffuse Fraction	1.0
Glow Factor	0.0
Reflectivity	0.0
Index of Refraction	1.0
Specular Fraction	0.0
Transparency	0.0
First Smoothness	1024
Second Smoothness	0
Second Weight %	0

☒ Enable object shadow casting

□ Link specular/diffuse colors

☒ Property values scale maps

10. **Load the Diffuse surface map into its field.** In the Surface Maps section of the Expert dialog, scroll to the right until you see the Diffuse field. Then, click on the Diffuse rectangle to insert the same “Eggshell - amb/diff/spec” image in this field. Follow the same steps used to place the map in the Ambient field. Go back to steps 6-9 to jog your memory, if needed.

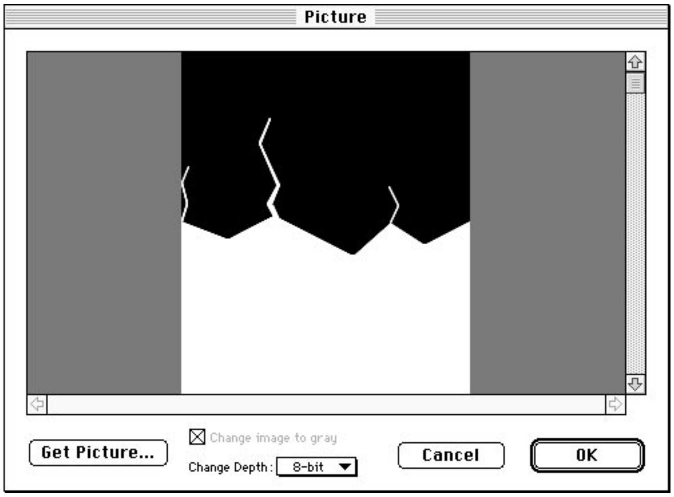
11. **Copy the map to the clipboard.** Before clicking on the OK button of the Picture dialog to insert the map into the Diffuse map filed, use the Copy command in the Edit menu to copy the image in this dialog to the clipboard. This is done to demonstrate an alternative method for inserting the same map in several fields.

12. **Load the Specular map field with the same map.** Of course, you could load the map into this field just as you did the other two fields…by now this is old hat to you. But, by copying the image to the clipboard, you won’t have to go to the hard-disk drive again.

    Scroll right in the Surface Maps section to find the Specular field (NOT to be confused with the Specular Color field). Click on the Specular field’s rectangle to open the Picture dialog as before. This time use the Paste command in the Edit menu to place the same image into this dialog, just as if you had loaded it from the hard disk.

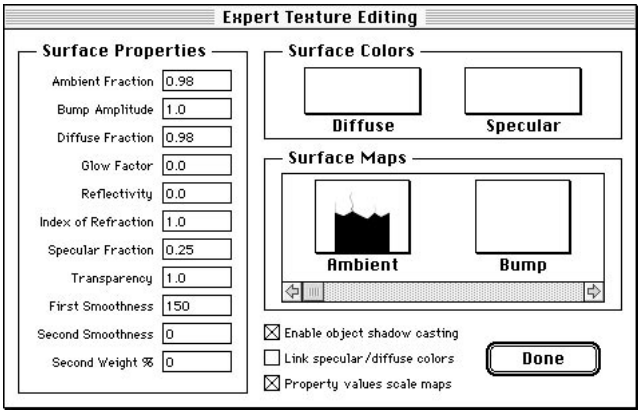
    Of course, all the rules of the clipboard are in force here, just like any other application. That is, only one item is stored on the clipboard at a time and a copy or cut command replaces what is on the clipboard with the new item. So, to be able to paste the same map into multiple fields, you cannot use the copy command to temporarily store something else and still retain the map you want pasted into the various texture fields.

13. **Load the "Eggshell - transparency" file into the Transparency field.** Use the same process as above. This is a different map, so it must be loaded from the hard-disk drive using the Get Picture directory dialog.



NOTE: *This map is inverted when compared to the first image used in the other fields. While all the fields work on the same principle — that of controlling the amount of a specific surface property with levels of gray in the desired pattern (as explained earlier in this chapter) — whereas the white area in the Ambient, Diffuse, and Specular maps is where you want light to reflect off the eggshell, the transparency map uses an inverted image because the white area in this map corresponds to that part of the sphere's surface you want to be invisible.*

14. **Set the Surface Properties fields to work with the Surface Maps fields.** Match the values in each of the surface properties to those shown in the illustration on the following page.



Expert Texture Editing

Surface Properties

Ambient Fraction	0.98
Bump Amplitude	1.0
Diffuse Fraction	0.98
Glow Factor	0.0
Reflectivity	0.0
Index of Refraction	1.0
Specular Fraction	0.25
Transparency	1.0
First Smoothness	150
Second Smoothness	0
Second Weight %	0

Surface Colors

Diffuse Specular

Surface Maps

Ambient Bump

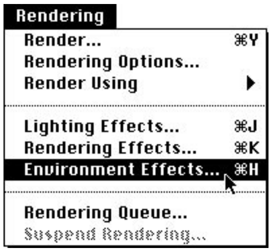
☒ Enable object shadow casting

☐ Link specular/diffuse colors

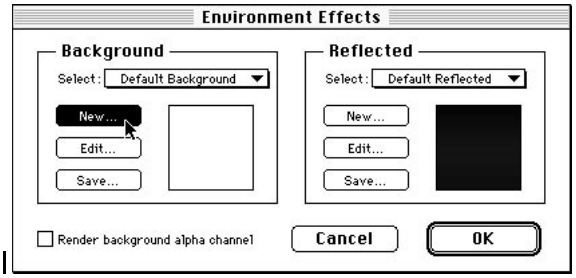
☒ Property values scale maps

Done

15. **Return to the model window.** Click on the Done button in the expert dialog, then on the OK button in the primary texture-editing dialog to return to the model window. Notice the Clear and Gloss slider settings in the Texture Editing dialog have changed to reflect the settings you entered in the Expert Texture Editing surface property fields.
16. **Make sure the Eggshell texture is applied to the sphere primitive in the model window.** Double check this using the same procedure you used in the previous exercise.
17. **Add a background environment to the model.** Pull down the Rendering menu and choose the Environment Effects command. Vision3d opens the Environment Effects dialog window.



18. **Create a new background.** On the left side of the Environment dialog box, click on the New button. Vision3d opens the New Environment dialog window.



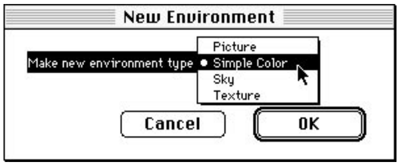
Environment Effects

Background
Select: Default Background
New…
Edit…
Save…

Reflected
Select: Default Reflected
New…
Edit…
Save…

□ Render background alpha channel
Cancel
OK

19. **Choose Simple Color as the new environment type.** Set the Make New Environment Type pop-up menu to "Simple Color."

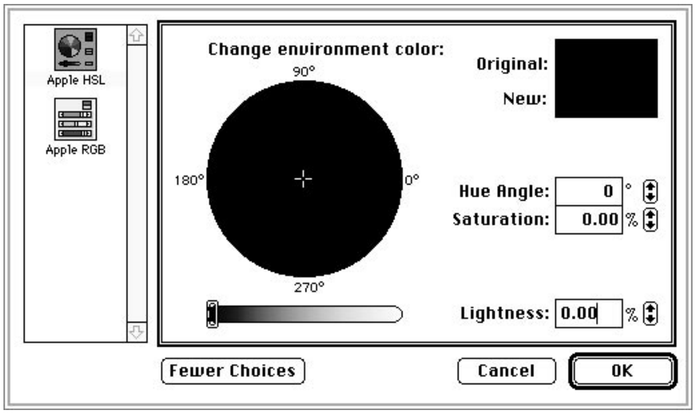


New Environment

Make new environment type
Picture
● Simple Color
Sky
Texture

Cancel
OK

20. **Click on the OK button.** Vision3d brings up the Macintosh color picker. The version displayed on your screen depends on the monitor's color depth.

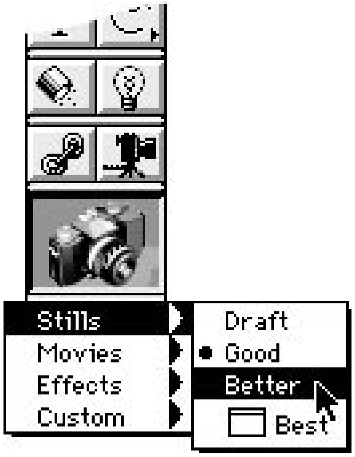


Change environment color:
Original:
New:
Hue Angle: 0 °
Saturation: 0.00 %
Lightness: 0.00 %
Fewer Choices
Cancel
OK

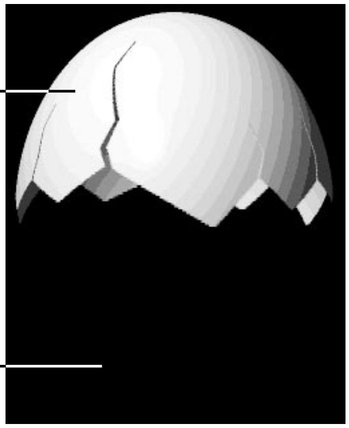
21. **Set the color to black and click on the OK button.** Vision3d returns you to the Environment Effects dialog. Notice the background environment preview box on the left side of the dialog is now black, showing the background environment color you chose in the color picker.

22. **Click on the OK button in the Environment Effects dialog.** Vision3d returns you to the model window.

23. **Render the sphere.** In Exercise 1, you used the Rendering Options command in the Rendering menu to set the rendering parameters. This time, click on the pull-down menu at the bottom of the Tool palette. Choose "Stills - Better" as the rendering parameters from the sub-menu, then release the mouse button. The Camera tool automatically selects. Move the cursor to the model window and click once to initiate a rendering.



The "Stills - Better" rendering setting will use the raytracer to render the sphere model. You will see successive passes of the renderer as it refines the image. This continues until the rendering is finished.



This cracked egg-shell started as a stretched sphere

The rest of the sphere is still in the model, but through the use of suface maps, it is invisible when rendered

Your finished rendering should look similar to the one shown in the above illustration. If not, re-check your texture's settings.

In this exercise you discovered how surface maps placed in the expert mapping fields work together with the surface properties values to create surface detail (or as in this case, the absence of surface through a transparency map) on an object in your model. This should help you better understand the power of surface maps to alter the appearance, and even the apparent existence, of an object or portion of an object.

Textures on light sources

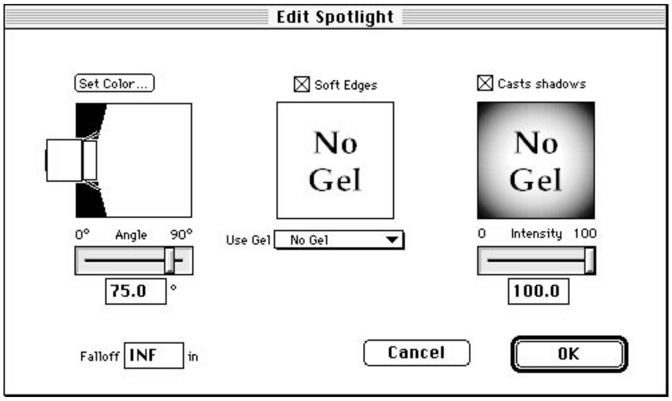
Lighting in a computer model in many ways is analogous to lighting for a theatrical play, i.e. — the scene is assembled, then lit for effect. A little drama in a rendered image can significantly improve its impact.

A light is a source of illumination. Light sources interact with surfaces (and the textures that define surface properties) to determine the amount and type of light that a surface reflects, as well as cast shadows on other surfaces.

The use of light sources s is discussed in detail in*Chapter 6: Lighting*. They are mentioned here as it relates to the use of textures on light sources.

Giving lights color…and more

All lights have color. The default color for light sources is white. A light source's color — whether it be a directional, spot, or point light — may be changed by opening the Edit Light dialog. For directional light sources, this is done by selecting the light source and clicking on the Edit button found on the Lighting palette.



Edit Spotlight

☒ Soft Edges

☒ Casts shadows

0° Angle 90°: 75.0 °

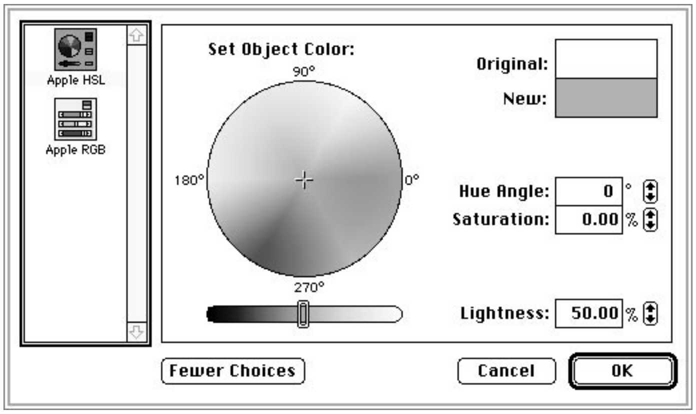
Use Gel: No Gel

0 Intensity 100: 100.0

Falloff INF in

For local light sources, you can select the light source's icon and choose the Edit Primitive command in the Modeling menu. Alternately, you can simply double-click on the light source. The following illustration showing the spot light's edit dialog is representative of the other light sources as well.

Lights can be any color. It is a simple matter of specifying a color in this dialog box for the light source. Clicking on the Set Color button opens the standard Apple Color Picker available in all applications.


Set Object Color:

Hue Angle: 0 °

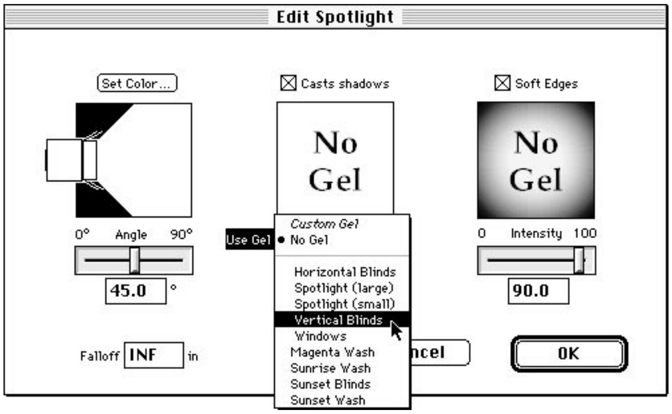
Saturation: 0.00 %

Lightness: 50.00 %

Of course, there are other controls in this dialog for intensity, falloff, soft edges, and more. These are explained in Chapter 6. Our focus here is with applying textures to light sources.

Using pre-built gels

Within the light editing dialog, you can also apply specialized "Gels" to a light source. This is done by clicking on the Use Gel pull-down menu. The gels listed in that menu are from the Gels folder in the Vision3d application folder.


Edit Spotlight

☒ Casts shadows

☒ Soft Edges

Angle 45.0 °

Intensity 90.0

Falloff INF in

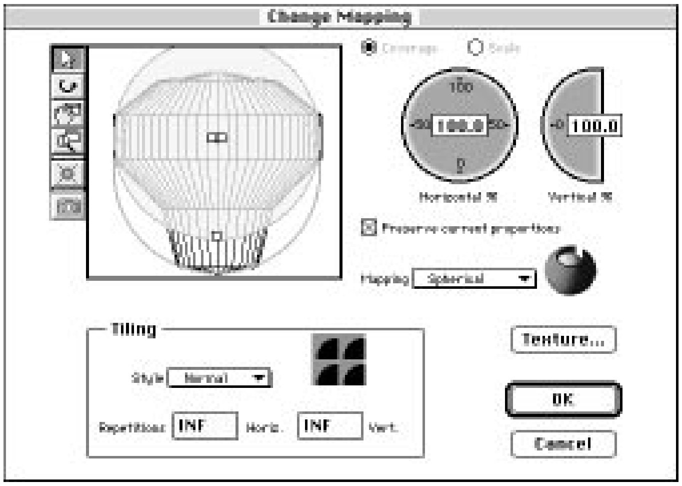
Gels are a type of texture, specially optimized for use with light sources. When a gel is selected, it is automatically loaded and applied to the light source with the proper orientation and scale so that it works correctly.

Making your own gels

Gels are the preferred method of using light sources as projectors of images. As noted, several pre-built gels are included in the Vision3d package. If none of the included gels fit your needs, you can create your own gels by selecting “Custom Gel” in the Use Gel field’s pull-down menu.

The familiar surface texture dialog box appears in which you can load any combination of surface maps you wish. While you have access to all the surface texture editing controls, some of the settings are not relevant to use on a light source. For example, since the texture is being projected instead of defining a surface, if you set a high gloss or reflectivity value it will just be ignored.

Once the gel/texture is defined, click on the OK button to return to the Light Edit dialog. The texture is mapped to the light source using a planar mapping style at 100% coverage of the icon, but you may change the mapping parameters by clicking on the Mapping button.



Change Mapping

● Coverage ○ Scale

Horizontal % 100.0

Vertical % 100.0

☒ Preserve current proportions

Mapping: Spherical

Tiling

Style: Normal

Repetitions: INF Horiz. INF Vert.

The also-familiar Change Mapping dialog appears in which you can adjust the mapping parameters to the specific needs of your custom gel. You may also re-edit your gel, if needed, by clicking on the Edit Gel button.

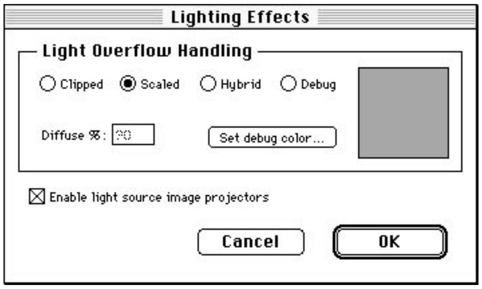
Using regular textures as a gel

Of course, you can also apply “non-gel,” standard textures manually to local light sources, just like any other object. You then set the mapping type, scale, and tiling method through the Change Mapping command.

Spot and Point light sources can accept textures to define the nature of their light, just as you can through the light editing dialog. However, the relevant characteristics a texture can impart to a light source only include color and the projection of color surface maps. Other settings like transparency, reflectivity, etc. (if included in the texture) are ignored for light sources. If the texture has a color defined, the illumination coming from the light source takes on that color. If the texture includes a color surface map, the light source can project the color map as a component of its illumination effect.

Whenever a surface map is used in a texture applied to a light source, that light source acts as a projector. Mapping type, scale and tiling parameters in the Change Mapping dialog are available to adjust the texture for that light source to obtain the projection desired. All mapping types are applicable with varying effects, i.e. — planar, spherical, cylindrical, etc.

Remember, as with objects, surface maps applied to light sources are direction-sensitive. That is, the orientation of the texture as it is projected from the light source is determined by the active view from which it was applied.


Lighting Effects

Light Overflow Handling

○ Clipped ● Scaled ○ Hybrid ○ Debug

Diffuse %: 20

☒ Enable light source image projectors

One more point, the “Enable Projectors” field (in the Lighting Options dialog) must be turned on for surface maps to work on light sources. The default is for this field to be turned on. If you have problems using this feature, you should check the “Enable Projectors” field first.

Directional Light Sources

The Lighting palette displays the directional light sources active in the current model. Directional lights are not the same as local light sources or geometric objects. They can only be edited through the Light Edit dialog.

Spot and Point Light Sources

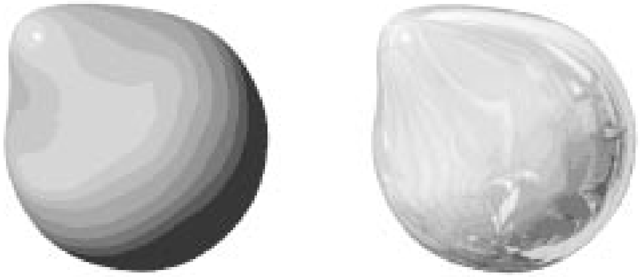
These light sources are inserted into the model as object "icons." That is, they are three dimensional in nature, but the scale of the icon has no affect on the size and intensity of the light source.

A point light source is omni-directional. A texture containing surface maps may be applied from any direction. Point lights would typically use a spherical mapping when applying your own textures manually.

Spot lights are directional in their illumination, like the global lighting, but local in their effect, like the point light source. Typically, textures using surface maps are applied from the front or back and set for Planar mapping. Applying the surface from the back projects the image as it is loaded into the texture. Applying the surface from the front reverses the image when projected.

Textures in environments

Vision3d's environment feature allows you to apply various images or patterns to an imaginary sphere surrounding the entire model. You can designate that the environment appears in the background or only reflects off the shiny objects in your model. Environments can add to the realism of a model by implying more detail than is actually modeled. An added benefit is that reflective environments also work with the Phong renderer.



Both images were rendered using Phong shading. Notice that for the right hand image the "Room Reflected" environment was loaded.

One of the three environment types supported allows you to use textures for the environment's image or color. Typically a texture-based environment includes a color surface map of some kind. Of course, you can use a texture for simple color in an environment if you wish, but that would be inefficient since solid colors can be implemented using the Simple Color environment type

Texture-based environments may use either surface or solid textures. Several texture-based environments are shipped with the product. You may use those provided, or create your own. When creating your own environments, you can use on of the textures provided or make a new one from scratch.

Applying a texture to the environment sphere is not the same as applying it to an object or light source in the model. It is important that you realize that environment textures have special considerations concerning size and tiling repetitions. The mapping style is always spherical, there is no option.

Obviously, every situation is different when making your own environments. To aid in understanding how to efficiently use this feature, you may wish to inspect one or more existing environments to see how large to make the map and how to set the scaling values to best fit the environment sphere so they appear as expected in the rendered image.

Additionally, once you have created an environment that you wish to reuse, you can also save environment files for use with other models.

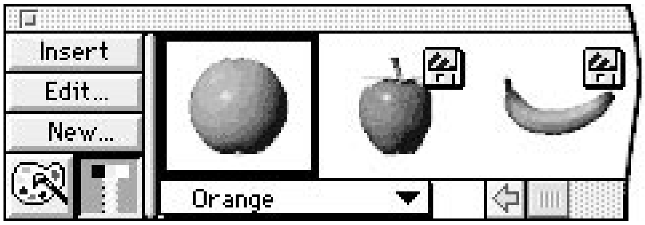
As you can see, you have total flexibility in the creation and use of environments. Textures are used for environments just like they would be used for objects, except that you cannot directly select the environment sphere. You must use the environment-editing dialogs to apply the texture to the environment and scale it appropriately.

Note that textures used for environment purposes become part of the model, just as when applied to objects. They are saved along with the other textures in the model and will be in place upon re-opening the model for a later session.

Textures Palette

The Shapes and Textures palettes share the same space on the screen. Clicking on the Shape button in the bottom left of this palette switches it to the Shapes palette; clicking on the Textures button switches it to the Textures palette.

Both Textures and Shapes use the same palette. The New and Edit buttons are reused while the Apply button changes to Insert.



Since so many of the issues involving shapes and textures are similar, you will find many aspects of Vision3d’s interface share this parallel or dual-purpose functionality for shapes and textures. Note also how the Shapes and Textures menus contain similar commands.

The Textures palette contains a preview image and a listing in the pop-up menu for each texture available in the Textures folder. At the beginning of a session, the Textures folder is scanned and those are the only textures displayed on the Textures palette. As you define new textures or load textures from other locations into the active model, they also appear on the Textures palette.

When a new texture is created in a model, it immediately appears on the Textures palette and can also have an image generated for the palette to help visually identify it. Texture files loaded from sources other than the Textures folder bring their own preview that automatically appears on the Textures palette.

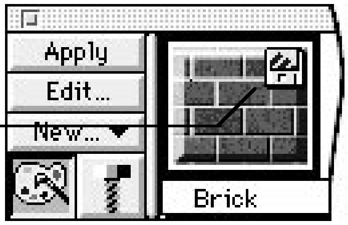
In addition, files imported into a model (which automatically creates a new shape workspace to contain the imported objects) may contain textures. Any textures that come with an object merged into the active model are given a space on the palette, even if it does not have a preview image for the palette. A preview is automatically created for a texture imported with an object.

If more than one model is open, switching active models causes the Textures palette to redraw. This is necessary so that it can accurately show the currently loaded textures for the active model.

Operation

Representations of all textures available to the current model are presented on the Textures palette. The available textures include those created in the model itself, those loaded into the model or those found in the Textures folder on the hard-disk drive upon start up of the application (but not loaded). A preview of these shapes may be viewed by scrolling their images horizontally in the palette window.

Notice the disk icon on textures that are not loaded into the current document.



When scrolling left to right on the Textures palette, the textures created in or loaded in the active model always appear first in the palette’s image preview area. Textures found in the Textures folder, but not loaded in the current model are identified by a floppy disk icon superimposed on its preview image. Loaded shapes do not have the floppy disk icon.

To select a texture on the palette, click once on its preview image. Only one texture can be selected at a time on the palette. A black rectangle appears around the preview. Alternately, you can use the pop-up menu to select a texture by name. Choosing a texture from the menu automatically scrolls the palette to show that texture's image and moves the selection rectangle to that texture.

To load a texture from the palette, double-click on its preview image. The floppy disk icon is then removed from that image, indicating that it is now loaded in the model. Note that this process only loads the texture, but does not apply it to any objects in the model.

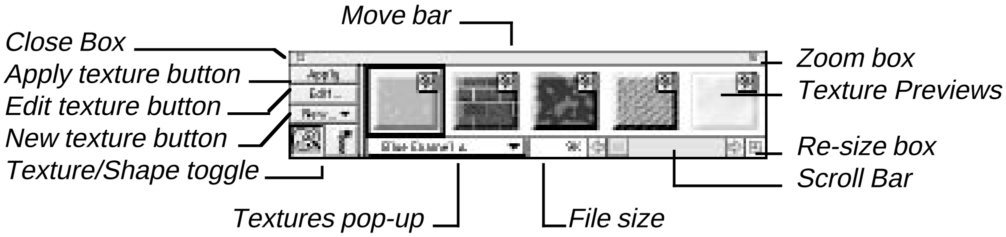
To apply a texture from the palette, select the object and the texture's preview on the palette, then click on the Apply button. Again, the active view determines the texture's application direction for mapping orientation.

The texture will be loaded into the model and applied to the object. As explained earlier in this chapter, the active view can be for the model space or a shape's workspace.

Even if the texture is already loaded, clicking on the Apply button still performs the apply operation. Additionally, with already-loaded textures, you can use the Set Selection command in the Textures menu to apply a texture.

Controls on the palette

Controls found on the Textures palette allow you to manage the textures represented there. Note that if more than one model is open, as with other aspects of the interface these controls have effect only in the active model.



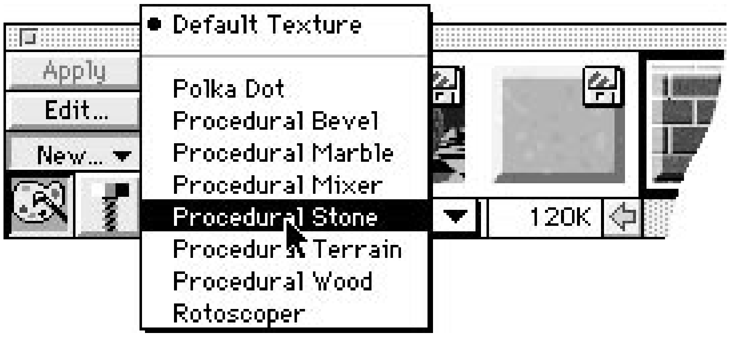
The Textures palette controls have been alluded to previously in this chapter while explaining certain operations. They are presented here item-by-item for reference. A discussion is also included so you can fully understand how useful this palette can be in managing textures.

- **Move bar.** Use this control to position the palette on the desktop. Vision3d remembers the position of the palette between sessions, and when showing the palette after it has been hidden.
- **Close box.** Click on this control to hide the Textures palette. This is the same as choosing the Hide Textures palette from the Palettes sub-menu in the Window menu, or clicking on the bottom right Hide/Show Palettes button on the Button bar.
- **Zoom box.** Click on this control to expand or contract the palette horizontally. Zooming out automatically expands the palette to the full width of the monitor on which it is displayed. Zoom in returns the palette's width to the size set by the re-size box. The palette does not expand vertically, even if there are more textures than can be displayed on the palette at full width.
- **Re-size box.** Use this control to manually set the width of the palette on the screen. You can drag the re-size control horizontally. The widest setting possible is the full width of the desktop (even across multiple monitors, if available).

  Note that the width you drag with this control "snaps to" the closest whole preview. You cannot set the palette's width to show only a partial preview image of a texture.
- **Scroll bar.** Use this control to scroll horizontally in the palette to view all the textures represented there. The scroll bar is active only when there are more textures than can be viewed at one time on the palette.
- **Textures preview area.** This area of the Textures palette contains a preview image for each texture available in the active model. Whether created in the model, loaded into the model, or still in the Textures folder on the hard-disk drive, each texture is given a place on the palette.
- **Apply button.** Click on this button to apply a selected texture to a selected object in the active model. The texture is loaded into the model, if needed, then applied to the object.
- **Edit button.** Click on this button to edit the texture selected on the Textures palette. A texture-editing dialog is opened for that texture. If the texture chosen to be edited is not currently into the model, it will be automatically loaded first. If the texture selected is already loaded, it is not re-loaded, but simply opens the texture's editing dialog box.

For those textures that are loaded in the active model, you can also use the Edit Texture command in the Textures menu to perform the same operation.

- **New button.** Click on this button to define a new texture. The new texture is created in the active model. By default, a surface-type texture-editing dialog box is presented in which you may enter the name of the texture and define surface properties for that texture. Clicking on the OK button accepts the texture settings and creates that texture in the active model. Clicking on the Cancel button discards the texture definition and no texture is created.



If you hold down the mouse button while clicking on the New button, a pop-up menu appears. This menu contains a list of the texture extensions from which you can create a new solid-type texture. Highlighting one of the texture extensions before releasing the mouse button opens a texture-editing dialog box appropriate for that type of solid texture. Each texture extension provides its own texture editing dialog containing the fields required to define a texture based on that texture extension.

The New button performs the same task as choosing the New Texture command in the Textures menu. Holding down the Option key while choosing the New Texture command provides the same access to the texture extensions list.

- **File size field.** This field indicates the size in Kbytes of the selected texture prior to loading. This is the size of the texture file as it is saved. The texture may actually take more space once loaded into the model, especially if any surface maps are used in that texture.
- **List of Textures (pop-up menu).** Use this menu to choose a texture on the palette by name. The menu lists any textures that are loaded in the active model first, with a divider between the loaded textures

and those not loaded but available in the Textures folder. Within each group, the listing is alphabetical.

- **Loaded/On-disk indicator (floppy icon).** The presence of this icon indicates that a texture is not loaded in the active model. It is removed from the preview images of textures loaded in the model.

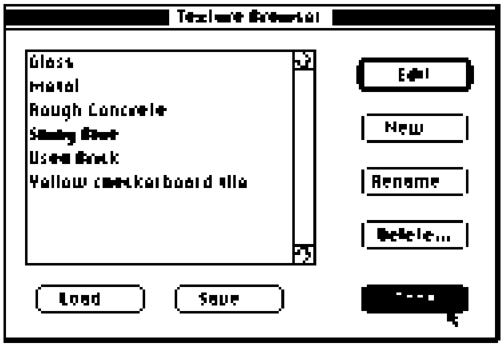
Commands and controls used with Textures

The following commands are considered pertinent to the handling of textures in the software. A detailed explanation is included under each command in the listing.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut to any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

Browse Textures command (Textures menu)

The textures browser is a specialized dialog box that provides several texture-handling functions in one location. The Browse Textures command is available when at least one texture is defined in the current model, either by creating a new texture directly or by loading a texture from the Textures folder or another location.



When the textures browser appears, a list of the textures in the current model is displayed for selection. One or more textures may be selected from the list at one time. Once a selection is made, the available operations that may be performed are indicated by which buttons in the dialog are active.

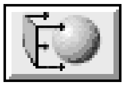
Some of the functions provided in the browser dialog are duplicates of other Textures palette buttons or Textures menu commands. They are also provided here for convenience. Some operations are available only through the browser dialog box, such as saving a texture as a separate texture file

- **Edit.** Clicking this button opens a texture dialog displaying a selected texture for editing purposes. Clicking on this button also closes the browser dialog box.
- **New.** Clicking this button provides the same operation as the New Texture command.
- **Rename.** Clicking this button allows the name of a selected texture to be changed. You may also rename a texture while in its texture editing dialog, but this can be more convenient at times.
- **Delete.** Clicking this button deletes the selected texture(s). Since this action is not reversible through the Undo command, an alert is displayed warning of the consequences before any texture is actually deleted. A Cancel button is provided as a way to exit gracefully without taking any action.
- **Load.** Clicking this button loads a texture into the active model. A directory dialog opens pointing at the Textures folder. You can select one of the textures located there or use the controls in the dialog to locate a texture file in another folder. The Load Texture dialog only recognizes files with the “VisA” file type.
- **Save.** Clicking this button saves the selected texture(s) as individual files used for textures libraries. This file has the “VisA” file type designation and is recognized by the Load Texture dialog.

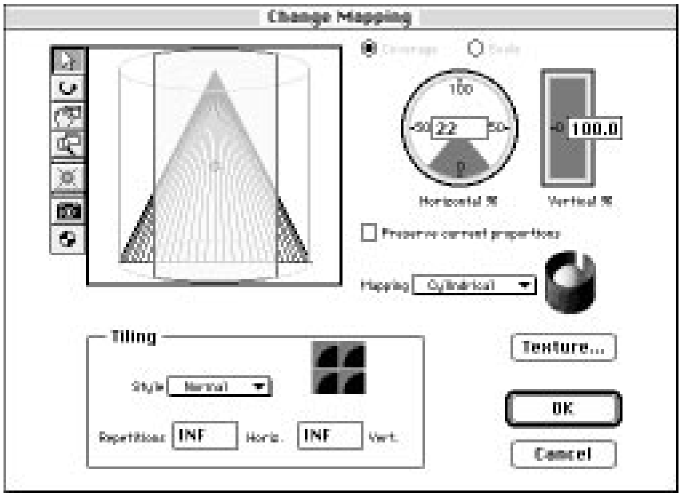
  During the save operation, Vision3d renders and includes a preview image of the texture which is used in the Load Texture dialog box and the Textures palette. To render an anti-aliased preview image, hold down the Option key while clicking on the Save button.
- **Done.** This button closes the browser dialog box and returns to the model.

Change Mapping command (Textures menu, ⌘M)

In addition to providing several ways to map surfaces onto an object, the Change Mapping dialog allows you to also set the scale and position of the surface map on the object. Further, controls are provided for setting the tiling style and the maximum number of repetitions a map makes across an object’s surface.



The Change Mapping command is available when a selected object has a surface-type texture applied containing at least one surface map. In addition, the button shown on the previous page also appears on the Button bar. If no maps are present in the surface-type texture applied to a selected object, both the menu command and the button are dimmed.



Change Mapping

● Coverage ○ Scale

Horizontal % 22

Vertical % 100.0

□ Preserve current proportions

Mapping Cylindrical

Tiling

Style Normal

Repetitions INF Horiz. INF Vert.

Only one object can be selected at a time to use the Change Mapping command. The mapping parameters are set in this dialog for each object, independently of the texture providing the map(s). That way, the same texture may be customized to fit different objects in different ways.

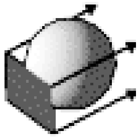
Mapping Type

The coverage or scale fields presented in the Change Mapping dialog vary with the mapping type selected. Surface textures may be mapped in one of the following ways:

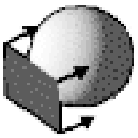
- **Planar.** Pushes the texture straight through the object from the direction it was applied. All faces of the object show the texture, even the inside and back.
- **Decal.** Applies the texture to only one side of an object. The pattern is similar to Planar, but is visible only on the surface(s) facing the active view at the time the texture is applied.
- **Cubic.** Places a complete representation of the texture on the six faces of a cube. If the object is not a cube, so that there is no corresponding geometric faces to receive the texture, the texture is still applied from six directions and shows on the object as they intersect the surface.

- **Cylindrical.** Wraps the texture around the object in a cylindrical fashion. If the texture is longer than the object (top and bottom), the texture smears to the center of the object on the top and bottom end caps.
- **Cylindrical II.** Wraps the texture around the object in the same cylindrical pattern as Cylinder does, except that the end caps are not smeared. Rather they receive their own face similar to the way cubic would work.
- **Spherical.** Wraps the texture around the object in spherical fashion, with the texture gathering together at the top and bottom "poles."

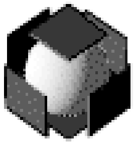
Vision3d tries to determine the most appropriate mapping type for the given object type, and will use that mapping type unless over-ridden by you. Any mapping type may be applied to any object. For best results, however, the mapping type should usually be matched as closely as possible to the object's geometry.



Planar Mapping

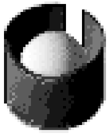


Decal Mapping



Cubic Mapping

Cylindrical Mapping

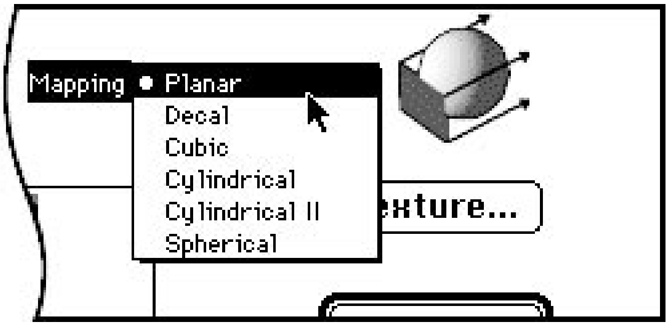


Cylindrical II Mapping

Spherical Mapping

Solid (Non-map and procedural textures only)

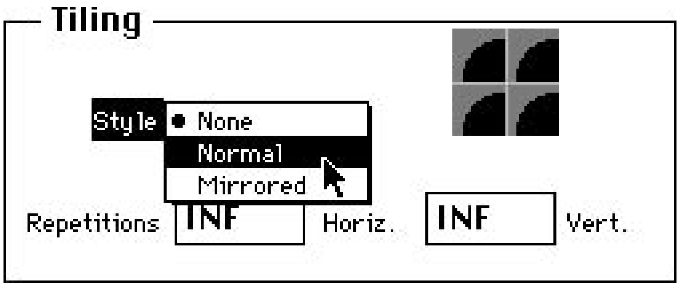
Textures are applied to the object from the direction of the active view. This is important when orienting surface maps. Each map in the texture is centered on the texture origin point in the active view, then wrapped around the object from that point in the pattern specified by the mapping type. Manipulating the texture origin point is explained later in this section of the chapter.



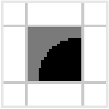
To chose the mapping type, click on the pop up menu of possible choices. Each type of mapping applies the chosen texture containing surface maps in a different way.

Tiling Style

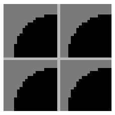
In combination with the horizontal and vertical scale settings, a surface map can be repeated over the surface of the object in two different patterns.



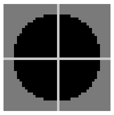
Repetitions INF Horiz. INF Vert.



No tiling (None)



Normal tiling



Mirrored tiling

You may choose from two ways of tiling, or no tiling at all. If no tiling is chosen, the image is mapped once at the indicated scale and/or coverage. The map is centered on the texture origin point facing the view in which the texture was applied to the object.

Normal tiling repeats the map in a straight manner beginning over each time the previous iteration of the pattern ends; mirrored tiling alternates the image end-for-end with each repetition across the surface.

Tiling allows you to keep surface maps to a manageable size, yet easily cover objects of widely-varying scale.

If you plan to use images of your own creation as surface maps, and they will be tiled, you must anticipate the need to match the pattern on each edge to avoid tiling seams.

Map Repetitions

The number of times a map will be repeated is also adjustable in this dialog. The default setting is “INF” or infinite in both the horizontal and vertical directions. The infinite setting insures that an object will be completely covered no matter how large or small the map is scaled relative to the object.

A value less than infinite in these fields repeats the map a specific number of times horizontally or vertically across the surface or until the edge of the object is reached, whichever occurs first.

Edit Selected command (Textures menu, ⌘‘ [apostrophe])

This command allows you to open a texture for editing based on the selected object. Edit Selected is available when a selected object has a texture applied. If no texture is applied to the selected object, or the selection is an instantiation of a shape (and the texture was applied at a different level in the hierarchy of that shape), the command is dimmed.

If more than one object is selected and all have the same texture applied, the command is available for opening the single texture they have in common. If the objects have different textures applied, the command is dimmed.

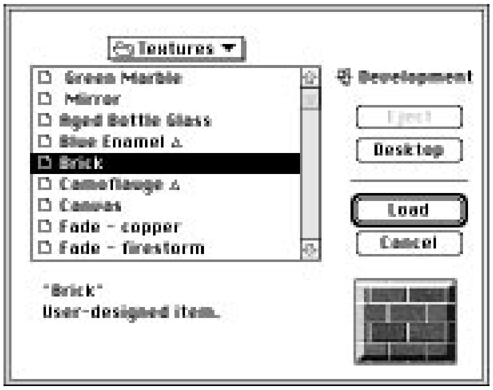
Edit Texture command (Textures menu)

Use this command to edit any texture currently loaded in the active model. The texture is chosen for editing from a sub-menu that appears when this command is highlighted in the menu. The texture’s edit dialog appears.

Only loaded textures can be selected for editing with this command. If you wish to edit a texture not currently loaded, you can use the Load Texture command first. Or, if the texture you want is represented on the Textures palette, you can select it and click on the Edit button.

Load Textures command (Textures menu)

The Load Texture command reads a texture file from any mounted volume or disk into the current model and includes it in the textures list that appears in the several menus throughout the interface. This command is available when a model is open.



Vision3d initially looks for textures to be loaded in a hard disk-resident Textures folder. The default data path is automatically set, but may be adjusted as needed by directory controls in the dialog to look for texture files outside that folder.

The loaded texture may be applied, edited, renamed or deleted again as desired using the other texture-handling commands.

Note: *Only texture files with a "VisA" file type are recognized by this Load Texture directory dialog. Non-texture files that may also be in the same location do not appear within this load dialog and cannot be loaded through this command. Other file formats and models containing textures may be merged with the active model using the Cut/Copy/Paste commands or the Import command.*

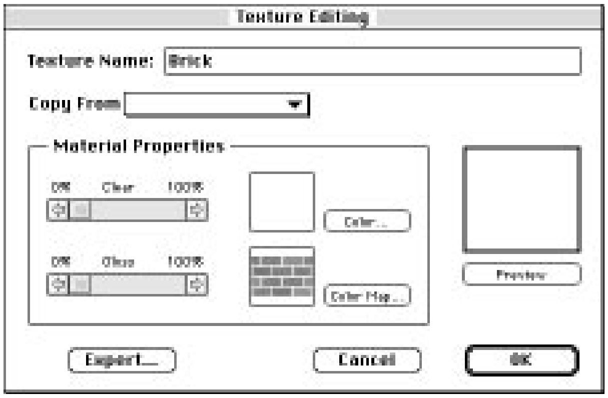
New Texture command (Textures menu)

The New Texture command initiates the creation of a new texture in the current model. New Texture is available whenever a model is open. This command has the same function as the New button on the Textures palette.

Choosing this command normally opens the dialogs for creating a surface-type texture. Holding down the Option key allows you to create solid-type texture.

Surface texture — Primary level

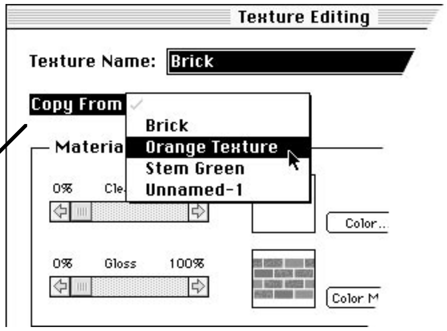
When called, the command displays the Texture Editing dialog. This dialog allows you to set various surface properties to describe a unique texture. A field in the dialog also provides for naming the texture, which is used in other dialogs and the Textures palette to identify the surface.


Texture Name: Brick

If one or more objects are selected at the time this command is called, the texture defined in the dialog box is automatically applied upon completion. Otherwise, the texture is available for manual application to objects through the Set Selection command.

The fields presented in this dialog, with their resulting effects, are:

- **Texture Name:** This is the name the texture will be cataloged under in the textures list. It is usually best to use a descriptive name that indicates the type of texture represented.
- **Copy From:** This field allows copying settings from another texture to be used in creating a variation for a new texture. Use of this field is optional when creating a texture.


Texture Name: Brick

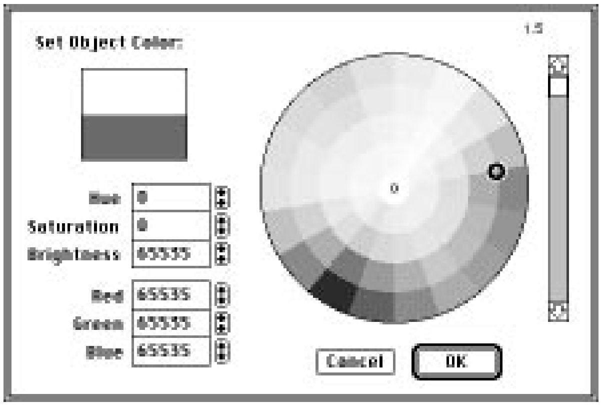
Pull down menu shows previously defined or currently loaded textures to use as a start for a new texture.

To use, position the cursor within the text box and hold down the mouse button. A pull-down menu of all textures currently available on the Textures list will appear. Drag the cursor downward to highlight the desired texture you wish to copy. Release the mouse button to execute the copy into the new texture.

- **Material Properties:** Two slide bars are used to set the general properties of the texture. These provide a convenient, graphical way to implement the most commonly used texture properties. “Clear” sets an object’s transparency. “Gloss” sets the polished appearance. Gloss is not the same as reflectivity. The Gloss slide bar inter-connects the smoothness, reflectivity, diffuse and specular properties of a surface.
- **Color:** This field allows you to set the native color of objects using this texture. An object’s native color may be altered by other fields accessed through the “Expert” button, but it will always be a factor in creating the rendered image.

  To set the color, click the cursor in the color box. This displays the Apple Color Picker. The resolution of the color palette presented is dependent on the colors available for the monitor where it is displayed.

  Position the cursor over the color desired and click the mouse button. Holding down the mouse button while moving the cursor will continuously update the values until the button is released. A color may also be set by editing the numerical values of its components. As numbers are entered, the resulting color is displayed in the box above the fields.



Set Object Color:

Hue 0
Saturation 0
Brightness 65535

Red 65535
Green 65535
Blue 65535

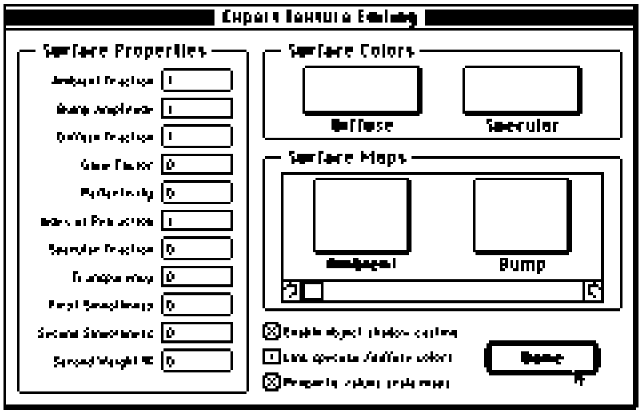
The color chosen will be automatically transferred to the Texture dialog by clicking “OK” or pressing the Return key.

Selecting “Cancel” will exit the color palette without altering the previous color.

- **Preview:** Clicking on this button will generate a representation of the texture on a generic object.
- **Expert:** Click on this button to display a secondary dialog with more fields relating to texture creation. This display should only be used to add properties to the texture not available in the primary dialog.

Expert level

The expert-level dialog offers greater capability for editing texture settings. The parameters entered in these fields influence the texture in more exacting ways than the primary dialog's sliding bars settings.



Expert Texture Editing

Surface Properties

Ambient Fraction	1
Bump Amplitude	1
Diffuse Fraction	1
Glow Factor	0
Reflectivity	0
Index of Refraction	1
Specular Fraction	0
Transparency	0
First Smoothness	0
Second Smoothness	0
Second Weight %	0

☒ Enable object shadow casting
□ Link specular/diffuse colors
☒ [illegible]

A limited explanation follows on what effect these fields have on the surface definition. However, this is not meant to be a definitive discussion of surface mapping. Once you are comfortable with this software, you should seek other reference materials to fully understand the concept of surface mapping and the effect of these fields on the rendering capability that has been built into Vision3d. Some reference materials are listed in the Appendix section of this manual

Surface Properties

Renderings of computer-generated three-dimensional graphics deal not only with the geometry of objects, their position and size, but also with the properties of their surfaces. The description and treatment of the surface of an object is as important as the geometric description of the object. Often, surface maps are a more elegant (and efficient) approach to the challenge of putting realistic details into a model than trying to create explicit geometry.

- **Ambient Fraction.** How much ambient light a surface reflects controls the apparent brightness of an object in situations where it is not directly illuminated with a light source, i.e. how visible is the surface when in the shadows? A zero value produces a surface that goes dark where shadows fall across it. A value of one results in a surface as bright as the ambient light level set in the Lighting palette.

- **Bump Amplitude.** This field sets the depth of depressions proscribed by the bump map during the rendering process. The higher the value, the deeper the depressions appear. However, the value must grow exponentially to achieve a constant increase. Put another way, a value of two is twice as deep as a value of one, but a value of four is required to double the depression again.

  Unlike the other fields in this dialog, the bump amplitude field must use a bump map to have any effect on the surface. If no bump map is present, no bump amplitude can be applied.

- **Diffuse Fraction.** All surfaces reflect light. Otherwise you would not be able to see the objects. Reflected light has two components — specular and diffuse. This field determines how much of the total light reflected from a surface is diffuse in nature. Diffuse light is light that scatters in all directions. Surfaces with predominantly diffuse reflections appear dull or matte in nature.

- **Glow Factor.** Glow determines the amount of luminescence a surface emits, without reflecting light from an outside source. The luminescence of a surface with a glow factor is not considered a light source and does not illuminate other nearby surfaces. The color of a glowing surface is determined by the color set in the texture, not the color of the light falling on the object's surface.

  Glow is used to give a neon-like appearance to objects or in special situations with heavy shadows, to ensure that certain surfaces will not get too dark.

  Glow is expressed as a value from zero to one. Zero produces no emissivity, there is no maximum setting, but a value of one (1.0) will usually "blow out" or overpower any other surface characteristics.

- **Reflectivity.** Reflectivity is available in Phong, Raytracing and Raydiosity. Phong's reflectivity is limited to interaction with the current reflected environmental map. Raytracing supports inter-object reflectivity, as well as environment maps. Raydiosity supports both specular and diffuse inter-object reflections, plus environment maps.

  Reflection is the bouncing of light off a surface into the area around it and toward the viewpoint. When a computer model is illuminated, the color of the light, the reflective properties of an object's surface — as well as the native color of the object — determines the color for that surface.

Reflectivity is a special case of specular light. This field determines the level of mirror-like reflection a surface may have. Reflectivity calculations must take into account the directionality of the surface — called the surface normal — at any given point. The amount of reflectivity observed varies directly with the angles between the light source, the surface normal, and the viewpoint.

The reflectivity setting is established as a value between zero and one. A value of zero produces a non-reflective surface. A value of one is fully reflective.

- **Index of Refraction.** Refraction is available in Raytracing and Raydiosity images. Light not only interacts with surfaces, but with volume as well. One volumetric light property is refraction, which causes light to bend or change direction whenever it travels through one medium to another, i.e. from air through glass through water and back to air.

  Refraction can be modeled in a Vision3d texture as a single constant — the refractivity of the material — and is used to simulate curved or optically transparent surfaces such as flat glass, lenses, crystals, gems, etc.

  A value of one produces the refraction index of air. A table of some of the more common materials with their index of refraction is in the Appendix section.

- **Specular Fraction.** This field determines the amount of specular light reflected from a surface. Surfaces that reflect mostly specular light appears shiny or glossy. Where diffuse reflections scatter in all directions, giving a dull or matte appearance, specular reflection is light that is reflected in the direction of the angle of reflection. Specular reflections create highlights or “hot spots” on an object.

  The first smoothness field also plays an important part in the appearance of specular reflections. Smooth surfaces reflect light with less distortion, which appear on the object as pinpoint highlights. Rough surfaces with high specularity scatter the reflection such that the hot spot is more dispersed or spread out.

  The appearance of the specular light and the ratio between diffuse and specular reflections in the light coming from a surface provides important visual clues about the materials composition of an object.

- **Transparency.** This field describes the amount of light that passes through a surface in the form of reflected light from other surfaces. A transparent surface transmits light without appreciable scattering, so that surfaces behind it are visible.

Transparent objects are treated differently depending on the rendering technique used. In wireframe, no surfaces are indicated, thus everything is transparent, so to speak. Hidden line and Quick shaded algorithms do not support transparency at all. Flat shading, Gouraud, and Phong algorithms use a “screen door” type of transparency in increments of zero, 25%, 50%, 75% and 100%, and use the increment that is closest to the value set in this field. Raytracing and Raydiosity can render continuously variable amounts of transparency matching the field’s setting.

Transparency is expressed as a real value between zero and one. A value of zero describes an opaque surface. A surface with a value of one is fully transparent and therefore invisible except for its effect on the objects behind it, such as color and refraction. However, fully transparent objects can also have ambient and diffuse values which cause the surface to become visible, and if set too high can obscure objects behind it.

- **First and Second Smoothness.** These two fields are used together to more accurately simulate the complex properties of metallic surfaces. A metallic surface is often both rough and smooth at the same time. Rough because of the microscopic faceting in its crystalline structure. Smooth due to the machining process used to create the shape.

  In addition, metals reflect light differently than other types of material. Metallic surfaces also have a high specular coefficient. Thus, even though a metallic surface may not have mirror-like reflectivity, it can still have strong specular highlights. These fields provide information to the rendering algorithms about how rough or smooth a surface is.

  Both fields have a range from zero to 32,767. A value near zero produces a less smooth appearance. The maximum value yields a completely uniform, machined surface. Values between zero and 1,000 are most often used in describing common metals. You should not think of these fields as a way to make surfaces rough-…only less smooth.

  Non-metals should use only the first smoothness field or have a second weight percentage of zero.

- **Second Weight %.** This field determines how much the Second Smoothness field contributes to the surface definition. A value of zero ignores the second smoothness field completely. A value of 100% uses this field exclusively and ignores the First Smoothness value. This is not a recommended technique; the First Smoothness field should always be the primary setting.

Surface Mapping

The fields contained in this area of the Expert dialog provide access to all the different mappings that may be used on an object.

Diffuse color and specular color are the only maps that can be color. All the other maps must be gray scale. Gray-scale maps are sampled at a certain frequency set in the rendering options dialog box. The value of each pixel in a map determines the level of effect for that property at that pixel in the rendered image.

For example, if a reflectivity map is applied to an object that has a 50% reflectivity setting, the most reflective areas of the map (those that are white) will not exceed 50%. To obtain full reflectivity in the surface map, the Reflectivity field must be set to its maximum value of one.

A texture may employ one or more surface maps in its definition. These maps correlate one-for-one with the Surface properties fields described here. The value in each surface properties field sets the ceiling for the amount of impact each surface map can have on a surface definition. In effect, they scale the range of the surface maps. If no map is present for a particular surface property, the value in the field for that property is applied evenly across the surface. If a map is present for a property, the values in the map are applied pixel-by-pixel in the rendered image, but do not exceed the ceiling set in the field for that property.

To access a map area, click within the desired box to display a dialog into which you may import or paste an image.

Surface Map Dialog

Surface mapping is a powerful feature of Vision3d. It is the process of using two-dimensional, bit-mapped images and treating them as if they were the surface of a two- or three-dimensional object. A surface is mapped differently depending on the type of map applied to it. The user need only create an image — either by digital photo capture or using a paint program — then map it onto an object. Images can be modified in a paint-type application to achieve the patterns and contrast desired for the specific mapping effect.

Surface maps are used in three ways:

- **Bump:** This map alters an object’s surface normals based on the map’s gray-scale values to reflect light in different directions; simulating the way light would be reflected if the object had the actual geometry described by the bump map.
- **Diffuse Color and Specular Color:** applies the color values of the map to form a detailed colored image or pattern on the surface.

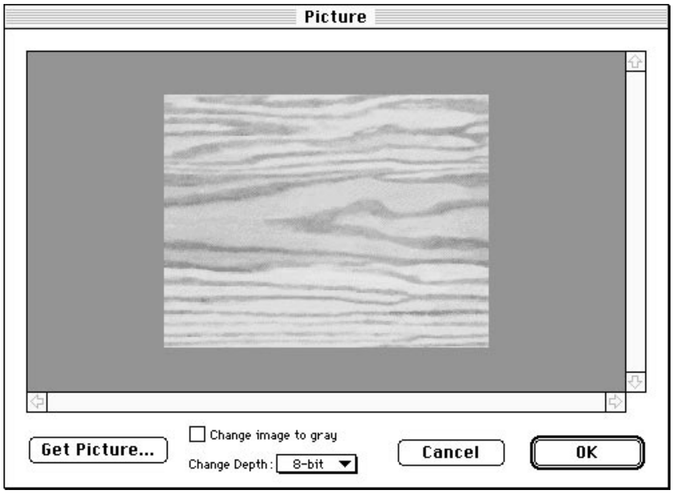
- **Glow, Reflectivity, Specular and Transparency:** Uses the lightness and darkness of a gray-scale image to alter the surface. The map is sampled pixel-by-pixel and the level of property represented by the map is adjusted up or down at the corresponding location on the object. White represents the maximum value of the surface property (up to the ceiling set in the properties field), black disables it. Values of gray fall within the two extremes.

Surface mapping works with the properties fields to create subtle deviations that more accurately simulate the detailed way light interacts with a surface. Mapping provides a way to include photograph-like detail — like labels, fabrics, or wood grains — to alter surfaces and give objects a more realistic appearance than is possible using geometric modeling.

For example, to create a realistic image of a brick wall, you would use a color image of the bricks for the diffuse color map and a gray-scale image for the depressions of the bricks in the bump map. The result would be a rendered image that gives a very realistic “look and feel” of brick.

NOTE: *At its simplest, mapping places a flat image on a flat plane. Other features of surface mapping allow wrapping images onto curved surfaces, such as a label wrapped around a can. In the case of a can or cylindrical shape, the image is not distorted. But a surface mapped onto a sphere or multi-sided object usually must be distorted or stretched to fit.*

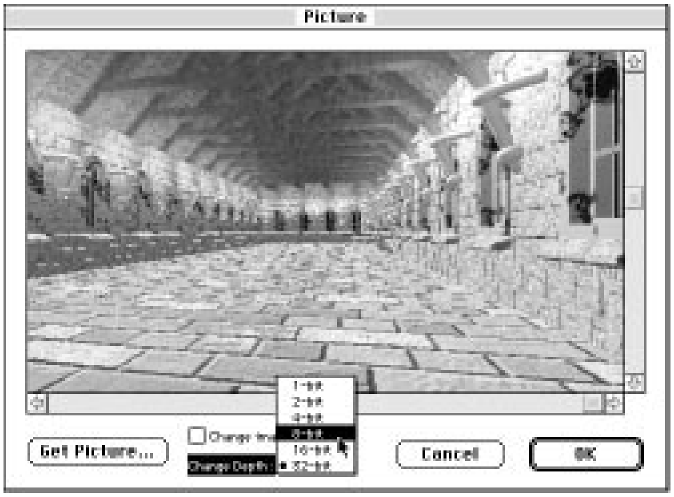
Whenever one of the mapping fields is selected, the Picture dialog appears to select and load the map. A mapping field is selected by clicking on its rectangle within the Surface Mapping area.



When this dialog is open, you may load a PICT file into the dialog to be used as a map. This is done using the "Get Picture" button. You may also use the Cut, Copy and Paste commands from the Editing menu on the image in this dialog. The clipboard and Scrapbook desk accessory available through the System are compatible with the operation of this surface map dialog.

Adjustable Color Depth of Surface Maps.

Control for setting color depth is provided in the "Get Picture" dialog, which is used for all mapping fields. Images can also be converted directly to gray-scale in this dialog, if needed.



Reducing a map's color depth can save memory. For example: a 300 pixel by 300 pixel map in 32-bit color requires 264K to store as part of a texture within a model, but the same size image in 8-bit color requires only 88K. Changes to the image occur only when the "OK" button in this dialog is clicked. Once they are made, you must re-load the map into the texture again to revert to its original color depth.

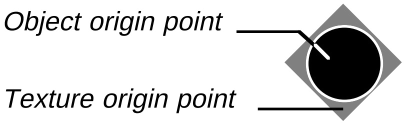
While some maps need to be in full color, many others work just as well at reduced color depth. Reduced-color maps can be used successfully when the colors are simple and contain large areas of solid color. Even though the map may contain simpler colors, the results can be just as rich. Renderings are still in 32-bit color and contain all the subtle shading and lighting-induced color shifts that give Vision3d images their incredible realism.

Reduced-color maps are automatically given an optimized palette to produce colors as close to the original as possible. Note that the image is not dithered. On occasion, you may see some banding or “artifacts” of colors in the map, especially below the 8-bit color depth. This is preferable, however, to getting pattern-aliasing from the dithering in the finished rendering.

Origin points

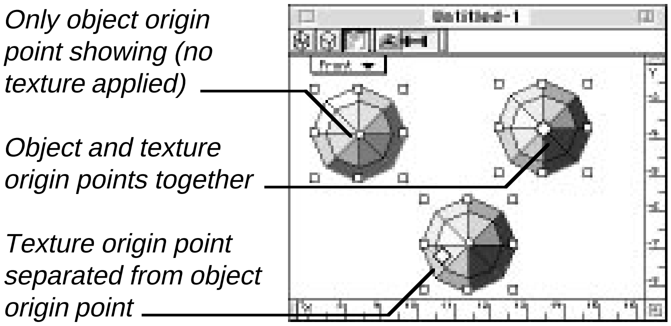
This is not a command in a menu, but rather a feature of the handles and object manipulation mechanisms that become available when an object is selected.

Each object has both an object origin point and a texture origin point. The object origin point for an object is always visible when it is selected. These points appear respectively as a small round dot and a small diamond whenever an object is selected. When these two origin points are both visible and are coincident (together), the object origin (dot) always appears in front of the texture origin (diamond) in all views.



The texture origin point is normally not visible when an object is selected. If the Show Texture Center command is enabled in the Textures menu AND a texture is applied to the object or shape instance at the current level, a diamond-shape handle appears behind the object origin point. This is the texture origin point.

NOTE: *The Show Texture Center command is disabled whenever a new session begins, even if it was enabled in the previous session. Since it is rare to move the texture origin point, you must consciously decide to show texture origin points in order to manipulate them on the objects.*


Only object origin point showing (no texture applied)
Object and texture origin points together
Texture origin point separated from object origin point
Untitled-1
Front

The texture origin point is used to position the center of surface map(s) in a texture when applied to the object. Moving the texture origin point moves the center of the map(s). This may be done in the model window or in the Change Mapping dialog.

Usually, both origin points are located in the geometric center of an object. However, they may be shifted in relation to the object, but ONLY IF the Show Animation Paths command is disabled in the Animation menu. This is an important protocol. If the Show Animation Paths command is enabled, moving the origin points always moves the object with it.

On new objects these two origin points are coincident. They can be separated, if needed, using the Control key. Either point may be re-positioned independently if the circumstances require it. For most situations, they will usually remain together. If the two points are together, they remain together when dragging to a new position.

Origin points that have been moved from the geometric center can be returned manually, if necessary. However, the Re-Center command in the Modeling menu is the easiest method for putting them back in the geometric center of an object.

While the default location of the origin points in objects created by Vision3d is at the geometric center, objects imported from other applications may not locate the origin points in the same manner.

Object origin point

The position of the object origin point (●) determines how a selected object responds to certain operations using the object manipulation tools:

- **It sets the axis for object rotation.**
- **It is where a child object is linked to its parent.**
- **It locates the object on its animation path.**

Texture origin point

The position of the texture origin point (◆) determines how surface maps contained in applied textures are mapped to objects:

- **It is where the center of a surface map is placed on an object.**
- **It sets the axis for surface map rotation.**

Moving origin points

Moving the texture origin point moves the surface maps in the applied texture. It is a technique for fine-tuning the mapping of surface textures. Moving a texture in the Change Mapping dialog is the same as moving the texture origin point in a model window, except that the texture origin point can be moved on all three axes — a feature that the Change Mapping dialog does not offer.

Normally, you cannot move the origin points independently of the object. Because this has such a far-reaching effect on the appearance of the object, as well as animation behavior, there is a specific protocol established to allow moving them. To move the origin points relative to the object, but maintain their spatial arrangement relative to each other, follow these steps:

1. **Turn off the Show Animation Paths command in the Animation menu.** This is necessary even if no animation has been scripted for the object.
2. **Select the Pointer tool.** While any of the three object manipulation tools can select objects, only the Object Move tool (a.k.a., Pointer tool) can move origin points.
3. **Position the cursor over the object origin point.** If the selected object has a texture applied, you will see two origin points, one behind the other. If no texture has been applied to the object, only the object origin point is visible.
4. **Hold down the Command key.** Without the Command key, dragging the origin point(s) also moves the object (even with the Show Animation Paths command disabled).
5. **Drag it to the new position.** The two origin points will move as one, leaving the object stationary.

Separating origin points

To separate the two points, hold down the Control key while dragging either of them. If both origin points are visible, the object origin point is in front and may be moved by simply clicking and dragging the dot. The diamond shape of the texture origin point always appears behind the object origin point when they are together, but may be selected by clicking and dragging one of the visible diamond points. With the Control key down, the chosen origin point will be moved while leaving the other point, as well as the object itself, stationary.

Just as the object itself can be moved any direction in three-dimensional space, an origin point can be relocated in any direction, on any axis, and at any distance from the object's geometric center. They can only be moved horizontally or vertically in the plane of a particular view. To move the point on the depth axis, use a perpendicular view of the object.

Modifier keys that apply to positioning object and texture origin points:

A. **Shift key** — Constrains the cursor to drag the origin points horizontally or vertically only. Unconstrained, the cursor can move diagonally.

B. **Command key** — Moves the origin points (both object and texture, if visible) without moving the object. Without the Command key, the object moves with the origin point(s). With the Command key down, the origin points move together, in their current spatial arrangement, without moving the object.

C. **Control key** — Allows independent movement of origin points. Without the Control key, dragging one object origin point moves the other as well. With the Control key down, either the object origin point or texture origin point (if visible) can be moved with moving the other point or the object itself.

Replace Selection command (Textures menu)

This command replaces the texture currently applied to a selected object with one chosen from its sub-menu. This is similar to the Set Selection command, but with one important difference. The mapping parameters set in the Change Mapping command for the first texture are retained for use with the second texture, including direction of application. Conversely, using the Set Selection command to apply or replace another texture causes all mapping parameters to revert to their default settings.

If no texture has been applied to the selected object, this command is dimmed. You must use the Set Selection command for applying a texture the first time.

Set Selection command (Textures menu)

This command applies the texture chosen from its sub-menu to the currently selected object. The texture is applied from the direction of the active view. If no object is selected, the command is dimmed. If more than one object is selected, the texture is applied to each of them.

If the selected object has a texture currently applied, that texture will appear in the sub-menu with a check mark before its name. If no texture is applied, the check mark appears before the “none” entry in the list. An object without a texture renders with an opaque, dull-white surface.

Even for objects with a texture already applied, using this command is like applying a texture for the first time. The mapping parameters for those object(s) are reset to the default values described earlier in this chapter, even if set for a previous texture.

This command has the same function as the Apply button on the Textures palette.

Show Texture Centers command (Textures menu)

Use this command to make the texture origin point visible. This command toggles on or off each time it is chosen. It defaults to off each time a new session is begun.

When enabled, the texture origin point appears whenever an object is selected that has a texture applied. The texture origin point defines the center of surface maps on an object.

Note that shape instances that have had textures applied at a different level show their texture origin point only at that level. Thus, a shape may have a texture applied at the shape level and its instance may render with a texture, but that texture is not controlled at the instance level.

Show/Hide Textures Palette command (Windows menu)

The Show/Hide Textures Palette commands occupy the same location in the Palettes sub-menu of the Windows menu; toggling as needed to display or hide the textures palette. The Show command is available when the textures palette is hidden. Hide Textures Palette is available when the textures palette is showing. Choosing the Hide Textures Palette command is the same as clicking on the close box of the textures palette or clicking on the Textures palette show/hide button on the Button bar.

Vision3d displays only one textures palette, even if multiple model documents are open. This palette (and all other palettes) always refers to the active document and its current state. Switching to another model causes the textures palette to redraw showing the status of the second model, as it becomes the active model.

The Textures palette's show/hide status when a new session is started is determined by the previous session. If the palette was showing when the application was quit the last session, it will be showing when a model is opened in the current session. If it was hidden at quitting last time, it will be hidden on start up this time…the status is retained from session to session.

CHAPTER 6: MANAGING LIGHTING

In Vision3d, light sources are used to provide illumination. Using dramatic lighting effects adds impact and realism to rendered images — for both still images and animations.

Objects are only visible when they are illuminated. As illumination from the light sources interact with object surfaces, the renderer determines the amount and types of light reflecting from those objects. The nature of that reflected light provides important visual clues about the materials used to make those objects, as defined by the textures applied to them.

The three different types of light sources allow you to customize the lighting for the specific needs of your project. Tools and controls are provided in the software to handle the light sources just as you would regular geometric objects.

There are three types of light sources:

- **Directional.** These are also called Global lights, which is a reference to their illumination effects.
- **Spot Light.** Shines in one direction only, with control for setting the cone angle.
- **Point Light.** Shines in all directions from the point of illumination.

Light sources may be added to a model in any number and combination. However, as the number of light sources used in a model increases, the time necessary to render the scene also increases.

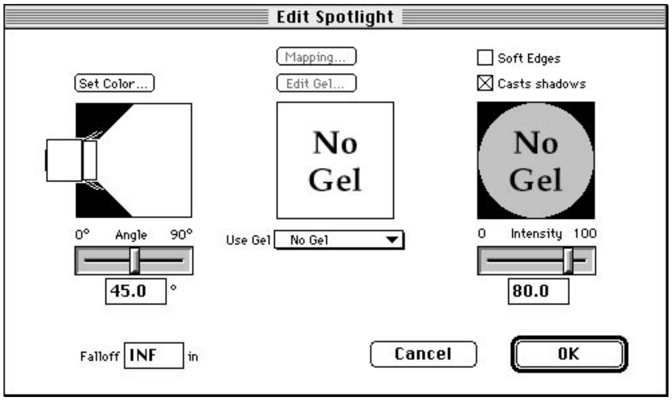
Adding multiple lights in a model also increases the overall level of illumination, with the risk of over-exposure. An automatic over-exposure control is built into Vision3d. This feature allows for some amount of illumination latitude, but can only handle so much concentration before being overpowered. Lighting over-exposure is explained later in this chapter.

Properties of lights

Two of the light sources, Spot and Point lights, are drawn in the model just like objects. When a spot or point light source tool is selected, the cursor changes appearance to that tool whenever it is within the boundary of the active view. They are considered local light sources and are represented by icons that can be manipulated in the model space with the Object Move, Object Rotate, and Object Scale tools.

Directional light sources, on the other hand, are considered global in nature. They are an infinite distance away from your model, much like sunlight striking the earth. Directional lights are added, adjusted for position, and deleted through the Lighting palette.

The light source itself is a finite point contained within the icon. Moving and rotating the icon moves and rotates the light source inside. This applies to all three types of light source.



☐ Soft Edges

☒ Casts shadows

Angle 45.0 °

Use Gel No Gel

Intensity 80.0

Falloff INF in

However, scaling the icon of local light sources larger or smaller DOES NOT increase or decrease the light source’s intensity. That illumination control is reserved for the Edit Primitive command in the Modeling menu. Depending on the type of local light source, the Edit Primitive dialog provides control for setting light intensity, attenuation distance, cone angle, etc.

Spot and Point light sources do not show directly in the views. Their position is represented by three-dimensional icons in the modeling views, which indicate the light source type and position.

Directional light sources, by definition are not IN the model in the same sense as the local light sources. More accurately, they are placed AROUND the model.

Renderings that include a local light source within the image will not show any light beam or glare at the point where the light is positioned. Only the light sources’ illumination on objects show in rendered images. Depending on the rendering options chosen, those effects may include smooth shaded surfaces, specular highlights on shiny surfaces, reflections on polished surfaces, shadows projected onto other objects, or any combination of these.

The three types of light sources provided in Vision3d have one or more of the following properties in common:

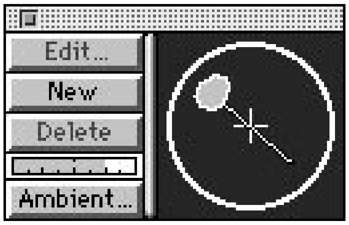
- **Position** — Lights may be positioned at infinity, like a directional light source. Or they may be local sources with a specific location in 3-D space, like the spot and point lights.
- **Intensity** — The quantity of light being emitted from the source. Intensity settings are a relative index with a range of zero to 100. There is no comparison in units that can be related to light in the real world.
- **Color** — Expressed either as a hue, saturation and brightness or a red, green and blue triplet. A light source's color is not calculated in wavelength internally.
- **Gel** — Specialized surface textures can be applied to light sources to project onto the surface of one or more objects. You may use a predefined gel or create your own.
- **Directionality** — Lights can be omni-directional, like Point light sources or directional like Spot and Directional lights. While the position and orientation of a local light source's icon determines its direction of illumination, the size or proportion of the icon does not control or even affect the distribution of the illumination coming from that light source.
- **Spread** — The cone angle of a spot light can be adjusted. Only the spot light source has this control. You may also set a hard or soft edge on the edge of the cone where the illumination ends.
- **Falloff** — The proximity of the light to an object determines how brightly the object is illuminated when the Falloff field is not set at "infinite." When set to infinite, there is no attenuation of a light source's intensity. The directional light does not have an attenuation control; only the local light sources have this feature.

Intensity, color, cone angle and attenuation are adjusted for each light source individually through the Edit Primitive command in the Modeling menu.

Additionally, light sources in Vision3d can accept textures — either as a Gel (within the Edit Light dialog, or as a texture applied to the light source as you would any other object (in the model window). Applying volumetric maps (in solid textures) or surface maps (in surface textures) to light sources causes them to be projected into a scene, providing realistic environmental lighting effects like multi-color washes, venetian blinds, window frames, curtains, etc.

Global/Directional lights

When a new model is started, a single directional light source is automatically present in the Lighting Palette. Its position is set so that the illumination is coming from the upper left front quadrant. (In other words, over your left shoulder as you face the screen).



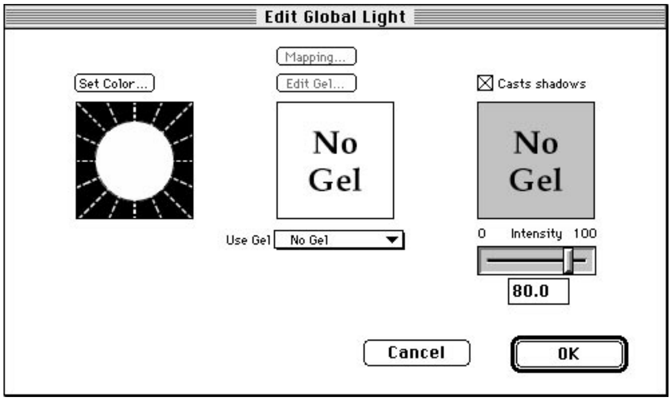
Of course, you can add, move, change intensity, or even delete the directional light sources using the buttons provided on the Lighting palette. If you remove all directional lights, the model will appear very dark unless you insert one or more local lights to take the place of the directional light.

The intensity of the default directional light is set at 80%. As you can see from the light meter on the Lighting palette, a single directional light at 80% intensity is sufficient to fully illuminate a model. The light meter measures only cumulative amount of all directional lights in the model and the ambient light level.

You can add as many directional lights as you wish, although this is not a typical use of this type of light. Multiple directional light sources can over-expose a model very quickly, so reducing their individual intensity is highly recommended.

The cumulative effect of multiple light sources on the lighting intensity in a model must be considered in the overall appearance of a rendered image. Ideally, the total of all light reflected from objects in a model should fall in an internal range between zero and one. Of course, that means nothing to you. What does matter is, if the total illumination exceeds the limit, the result is over-exposure. This can be avoided by judicious attention to how multiple light sources are positioned and adjusting their lighting intensities accordingly.

Directional light sources shine “toward” the model from a direction outside the model space. As mentioned, a directional light source is intended to simulate sunlight striking the earth. Its rays of “light” are parallel to each other. Thus, even in images rendered with algorithms that support soft edges on shadows, directional lights cast hard-edged shadows.



☒ Casts shadows

Use Gel: No Gel

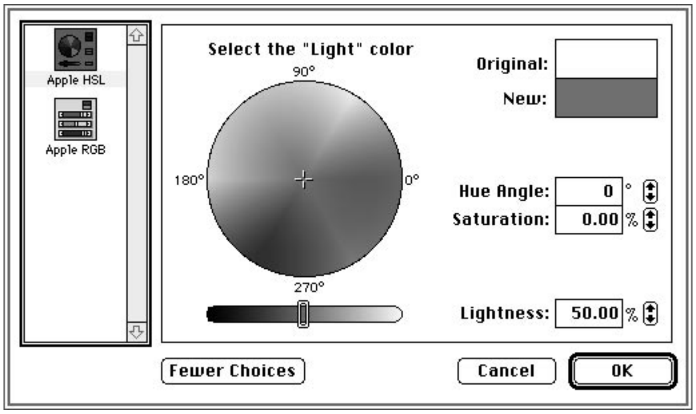
Intensity: 80.0

The illumination parameters of a directional light source are set or adjusted through the Edit Global Light dialog. This is opened by selecting the directional light on the Lighting palette, then clicking on the Edit button on the left side of that palette.

The fields in the Edit Light dialog for directional light sources are explained in the following topics under this section of the chapter

Setting Color.

The directional light source(s) present in the Lighting palette display in the color given in the Edit Light dialog box. Normally, directional lights are white, but may be any color you want.



Hue Angle: 0 °

Saturation: 0.00 %

Lightness: 50.00 %

The color for a directional light is set with the standard Apple Color Picker dialog box. The color picker dialog is opened by clicking on the Set Color button in the Edit Light dialog. In configurations where there are multiple monitors present on a machine, the color picker dialog ALWAYS opens on the monitor with the best color depth.

NOTE: *Even if a monitor is turned off, but the video card is installed in a NuBus slot on the machine, the operating system still thinks that monitor is present as part of the desktop extents. Thus, if the monitor that is off has a higher color depth than the other(s), the color picker actually opens but you will not see it and the software may appear to freeze up.*

The number of colors presented for selection in the color picker dialog depend on the color depth of your monitor.

Once a color is selected, clicking on the OK button returns you to the Edit Light dialog. The chosen color is then represented in that dialog. Clicking on the OK button again in the Edit Light dialog returns you to the model.

Setting intensity.

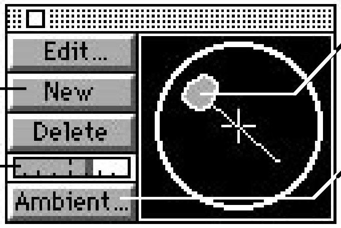
A directional light's intensity can be adjusted graphically with the slider control or numerically in the field below the slider. As expected, a setting of zero produces no illumination, 100 is maximum illumination.

If you need more than 100% illumination, you can add multiple directional lights in the same location on the lighting sphere.

Lighting palette

Any discussion of directional light sources must include the Lighting palette upon which this type of light resides and is accessed. The Lighting palette presents controls for handling directional light sources, as well as ambient lighting.

The Lighting palette is open by default whenever a session is begun. The Lighting palette may be hidden or shown by using the "Show/Hide Lighting Palette" command in the Palettes sub-menu of the Windows menu or by clicking on the Lighting palette's button in the Button bar. Alternately, you can click on the palette's close box, which has the same effect as using the hide command.



Use these buttons to adjust, add or delete a light source

LIght meter

Drag here to position directional light around model

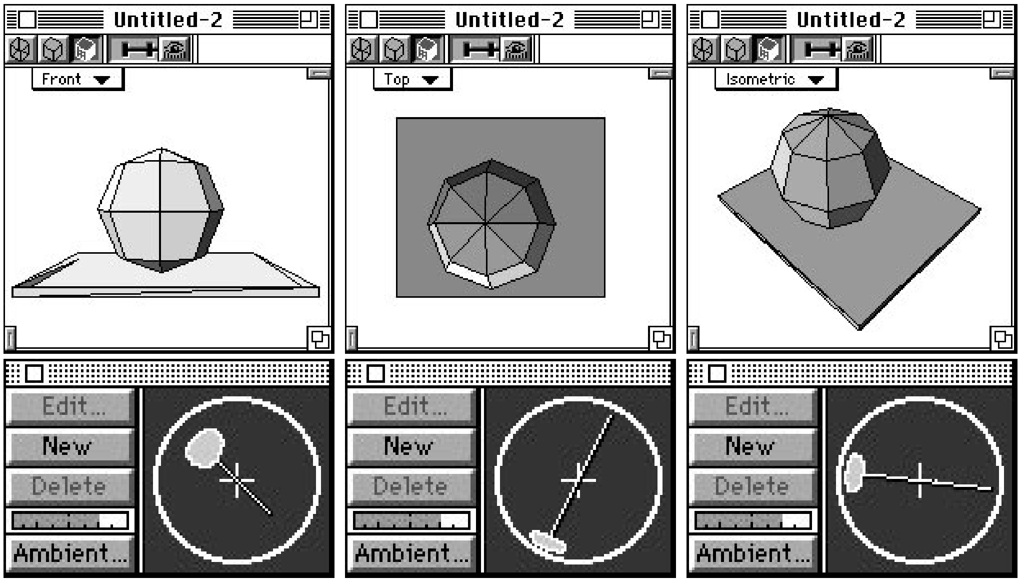
Click here to set ambient light and color

As mentioned, directional light sources illuminate a model with parallel rays of light, similar to the way sunlight illuminates objects on Earth. They are considered the primary light source for a model. In comparison, ambient lighting is a non-directional general illumination that fills shaded or shadow areas of a model so that details not directly illuminated by light sources are still visible.

Positioning a directional light on the lighting sphere

Think of the area on the palette containing directional light sources as a sphere surrounding the model. Directional light sources may be added or deleted to the sphere as needed. Each light may be selected and moved anywhere on the sphere to illuminate the model from that direction.

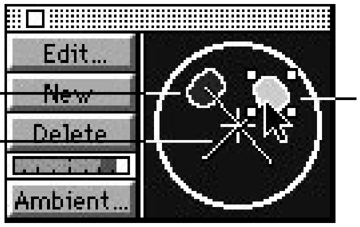
A directional light's position may be adjusted by dragging the light source around the lighting overlay sphere. A pointer is attached to each directional light showing what direction the light from that source is striking the model .



The lighting sphere displays the position of directional light sources relative to the currently active view. If the active view is changed, the position of the light(s) shifts on the sunlight palette to correspond with the new view.

Toggling directional lights between hemispheres

To move a light from the front hemisphere to the back (or vice versa), based on the active view, hold the Control key down while dragging the light source. The light source changes appearance to indicate its new global position relative to the model. Releasing the mouse button before the Control key keeps it on the new hemisphere. The light reverts to its original hemisphere if the Control key is released before the mouse button.



Appearance of light source icon when on back hemisphere

Appearance when on front hemisphere

Handles appear when a light source is selected on the palette

Once toggled to a hemisphere, you may move the light without holding the Control key down. Usually, directional lights on the back hemisphere are given a reduced intensity. Back lighting a model with a directional light may be useful for bouncing a highlight off some of the objects. It may take a couple of adjustments to get the highlight just right, but does add to the quality of an image.

Light meter

The Lighting palette also displays a simple light meter to warn of potential over-exposure problems. This light meter measures the cumulative light coming from the directional light sources and the ambient light level. Local light sources are not considered in the meter’s level.

The meter is intended to alert you to the possibility of over-exposure only, not necessarily that it will occur. Often, the placement of a directional light at half intensity on the back hemisphere (relative to the active view) can provide a needed fill which will not over expose.

Over exposure is usually first noticed in surface maps that appear washed out. If you are not sure how and where excess illumination levels are affecting the scene, you can set the lighting control to “Debug” through the Lighting Effects command in the Rendering menu. In that mode, taking a preview rendering causes the debug color to appear wherever the illumination levels are too high. You will then be able to manually adjust the light sources as needed to compensate.

Additionally, Vision3d provides several automatic exposure control methods to help control excess lighting. These are also accessed through the Lighting Effects dialog box. These lighting options are discussed later in this chapter under the Lighting Effects command's description.

Modifier keys that apply to Directional light sources:

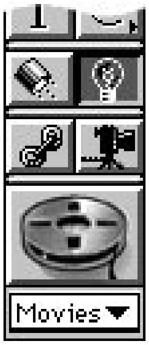
A. **Shift key —** Extends the selection of light sources on the palette. Multiple light sources may be selected at one time for moving in unison, setting parameters or applying textures.

B. **Control key —** Toggles the hemisphere orientation of a Directional light source as it is being dragged around the model. The front/back hemisphere orientation in the lighting palette is always relative to the active view.

C. **Option key —** Leaves a copy of the light source in the original position as the original is dragged to a new location. This is an alternative method to clicking the New button on the palette, but is most useful when you need to duplicate a light source's parameters for another light.

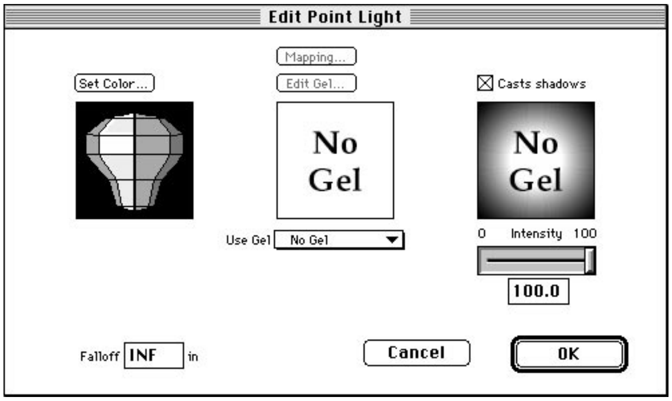
Point light sources

A Point light is a local source of illumination. It shines in all directions from a single point contained within the point light icon. A point light source is inserted into the active model using the Point Light tool.



When a new point light is inserted into a model, the Edit Light dialog box automatically appears so you can change its color, intensity, attenuation, gel, etc. from the default settings.

This is the same dialog that opens if you choose the Edit Primitive command in the Modeling menu when a point light source is selected. If you make changes to the light's settings in this dialog, clicking on the OK button accepts those changes and applies them to the light source. If you click on the Cancel button, no changes are made to the light source, but the light source is still inserted into the model. To remove the point light source, simply select it and push the Delete key. Alternately, you can choose the Delete command in the Edit menu.



Edit Point Light

☒ Casts shadows

Use Gel No Gel

0 Intensity 100

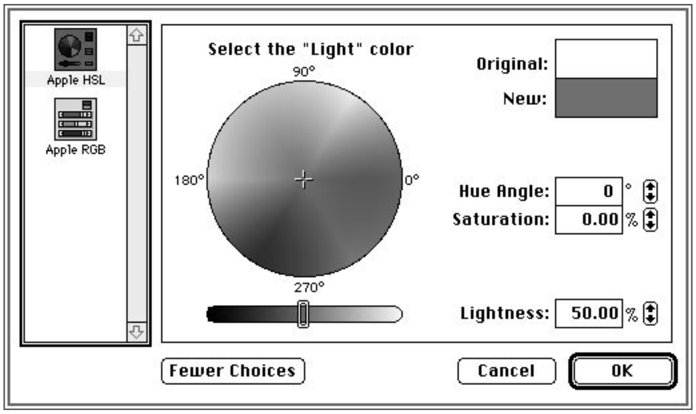
100.0

Falloff INF in

If you DO NOT want the Edit Light dialog to open for a light source upon insertion, hold down the Control key before clicking or click-dragging. The light source is inserted into the model with its default settings.

Setting Color.

The color for a point light is set with the standard Apple Color Picker dialog box. The color picker dialog is opened by clicking on the Set Color button in the Edit Light dialog.



Select the "Light" color

Hue Angle: 0 °

Saturation: 0.00 %

Lightness: 50.00 %

Once a color is selected, clicking on the OK button returns you to the Edit Light dialog. The chosen color is then represented in that dialog. Clicking on the OK button again in the Edit Light dialog returns you to the model.

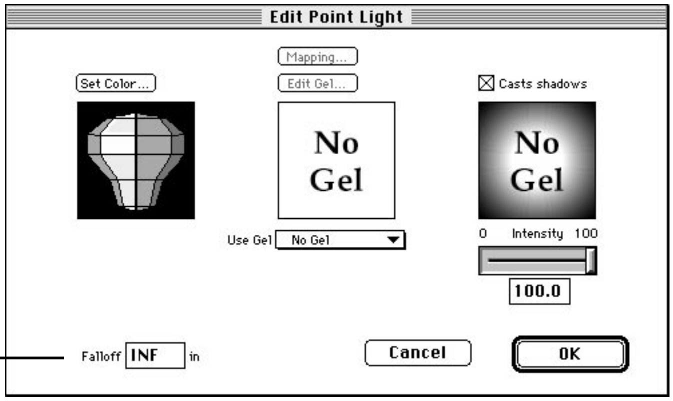
Setting intensity.

A point light's intensity can be adjusted graphically with the slider control or numerically in the field below the slider. As expected, a setting of zero produces no illumination, 100 is maximum illumination for that light source.

Inserting multiple light sources is effectively the same as increasing the intensity above 100% for one light source. Neither method is recommended under normal circumstances because of over-exposure concerns. If you want an object to appear extra bright, a better technique is to use a glow factor in its applied texture.

Setting Attenuation

By default, point light sources are inserted in a model set to an infinite distance of illumination. For most purposes, this setting is desirable. Entering an illumination distance in the Edit Light dialog for a point light source causes its illumination to decrease over distance. This can be useful when setting up special or dramatic lighting effects in a local area of a model or scene.



Falloff INF in

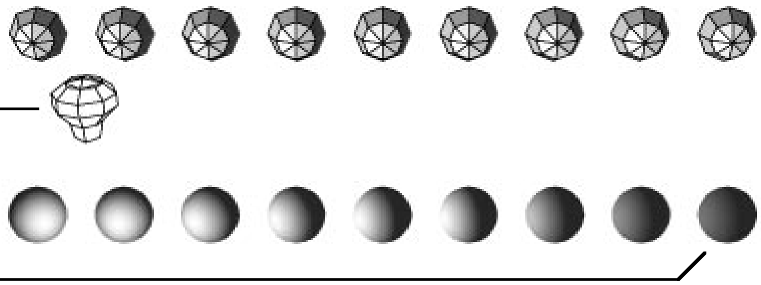
100.0

This field sets the maximum distance at which a Point light source will illuminate objects

The Falloff field indicates how far the illumination travels. This value is the absolute maximum distance that an object will be illuminated by that light source. Note that an object positioned just inside that maximum distance may not APPEAR illuminated even though it is, because the amount of light is so low.

Reducing the intensity of the light source will not cause the attenuation to fall off any quicker. The falloff value is a specific distance — meaning that if the light source is reduced in intensity, it will still illuminate objects in the model to that distance, but the closer objects will be less bright.

Light attenuation



Example model

Falloff for this light source was set to 12"

Rendered image shows illumination falloff, with last sphere at 13.5" not illuminated at all

There is no upper limit to the value, unless you consider “infinite” a limit. The infinite setting (INF) disables any attenuation effect, i.e. — no fall off occurs for that light…ever. This field uses the same units defined in the Set Rulers dialog box, which is the units of scale for the model being assembled. The value in this field is the distance at which a light source’s illumination is no longer visible; the distance at which the falloff of the illumination from that source is complete.

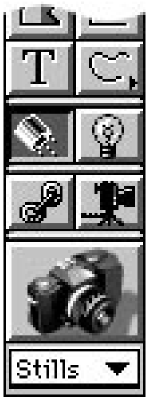
Modifier keys that apply to point light sources:

A. **Shift key** — Constrains the tool to draw the icon with the same dimensions on all axes.

B. **Option key** — Draws an object from its center, instead of the corner.

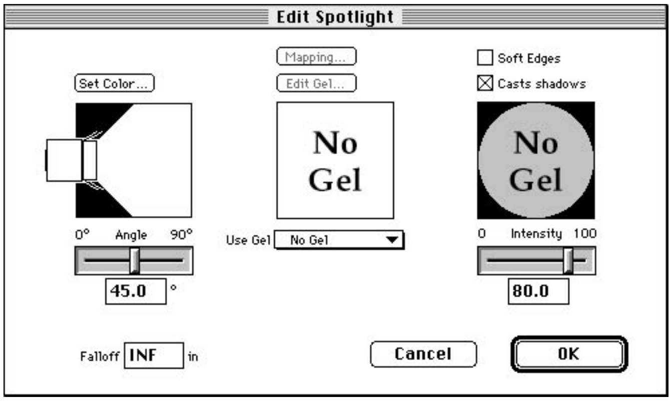
Spot light sources

A spot light is also a local source of illumination, like the point light. However, it shines in only one direction, like the directional/global light. The illumination comes from a single point contained within the point light icon, so its light rays are not parallel like the directional light.



Just like the point light, a spot light source is inserted into the active model using the Spot Light tool. This is located just under the Text tool on the left side of the Tool palette.

When a new spot light is inserted into a model, the Edit Light dialog box automatically appears so you can change its color, intensity, cone angle, attenuation, gel, etc. from the default settings.



Edit Spotlight

☐ Soft Edges

☒ Casts shadows

Angle 0°–90°: 45.0 °

Use Gel: No Gel

Intensity 0–100: 80.0

Falloff INF in

This is the same dialog that opens if you choose the Edit Primitive command in the Modeling menu when a spot light source is selected. If you make changes to the light's settings in this dialog, clicking on the OK button accepts those changes and applies them to the light source. If you click on the Cancel button, no changes are made to the light source, but the light source is still inserted into the model. To remove the spot light source, simply select it and push the Delete key. Alternately, you can choose the Delete command in the Edit menu.

If you DO NOT want the Edit Light dialog to open for a light source upon insertion, hold down the Control key before clicking or click-dragging. The light source is inserted into the model with its default settings.

Setting intensity.

A spots light's intensity can be adjusted graphically with a slider control on the right side of the dialog. You can also set the intensity numerically in the field below the slider. As expected, a setting of zero produces no illumination, 100 is maximum illumination.

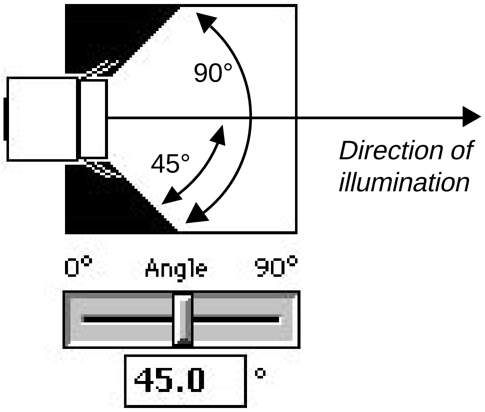
If you need more than 100% illumination, you can add multiple spot lights in the same position. This is not recommended, because of overexposure concerns.

Setting cone angle

Only the spot light source has a cone angle setting. A spot light's cone angle or degree of spread can be adjusted graphically with a slider control on the left side of the dialog. You can also enter the cone angle numerically in the field below the slider. The range of from 0° to 90°. Of course, a setting of zero shuts off any illumination coming from the spot light, regardless of the intensity setting.

Measuring cone angle

A setting of 45° for the cone angle in a Spot Light's Editing dialog, corresponds to a total included angle of 90°; 60° corresponds to 120°; 90° corresponds to 180°; etc.



The cone angle is measured as the amount that the cone of illumination diverges outward from the direction of illumination. This is different than the total angle of the cone of illumination. That is, a setting of 45° has an effective total angle of 90°, a setting of 90° is effectively 180°, and so on.

Obviously, the size of the cone of illumination striking objects in the model depends on both the cone angle and the spot light's distance from the object. Just as in the real world, a spot light can be moved closer to or farther away from an object to adjust the size of the spot of light.

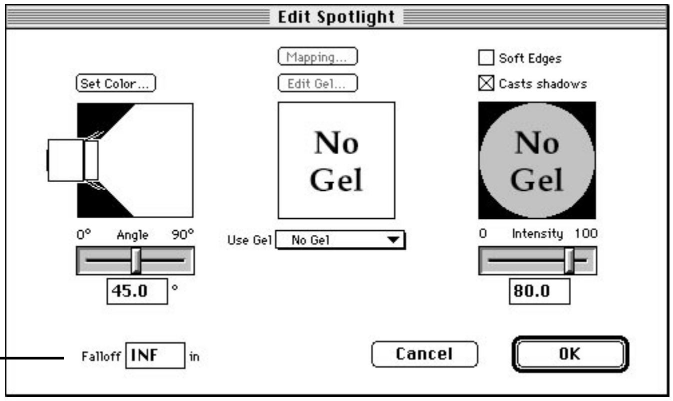
Setting soft edges on the spot light

Connected with the cone angle, there is a checkbox in the spot light's editing dialog that allows you set hard or soft edges on the cone of light as it strikes objects. This is located in the upper right corner of the dialog.

The default is to illuminate with hard edges for the spot of light. Turning on soft edges requires more calculations and can slow the rendering speed, especially if many spot lights are used in a model.

Setting Attenuation

By default, all light sources are inserted in a model set to an infinite range of illumination. For most purposes, that setting is desirable. Setting a falloff distance in the Edit Light dialog for a light source causes its illumination to decrease over distance. The ability to do this can be useful when setting up special or dramatic lighting effects in a specific model or scene.



Edit Spotlight

☐ Soft Edges

☒ Casts shadows

Angle 45.0 °

Use Gel No Gel

Intensity 80.0

Falloff INF in

This field sets the maximum distance at which a Spot light source will illuminate objects

The Falloff field sets the maximum distance a light source will illuminate objects. As mentioned in the point light source discussion, reducing the intensity will not cause the attenuation to fall off quicker. The attenuation distance is a specific measurement, meaning that if the light source is reduced in intensity, it will still illuminate objects in the model to that distance, but the interim objects will be less bright. An object positioned at or near that maximum distance may not appear to be illuminated, because the intensity is so low. If you want a faint illumination effect, you can simply set the Falloff distance higher to get the light to travel a little farther.

There is no upper limit to the value, unless you consider “infinite” a limit. The infinite setting (INF) disables any attenuation effect, no fall off occurs for that light. This field uses the same units defined in the Set Rulers dialog box, which is the units of scale of the model being assembled. The value in this field is the distance at which the light’s illumination is no longer visible; the distance at which falloff of the light from that source is complete.

Modifier keys that apply to spot light sources:

A. **Shift key —** Constrains the tool to draw the icon with the same dimensions on all axes.

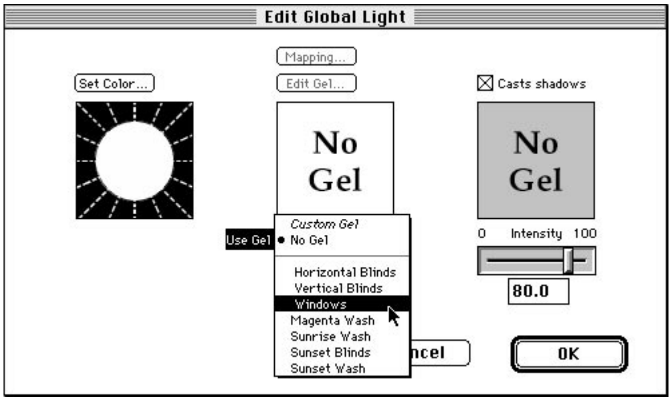
B. **Option key —** Draws an object from its center, instead of the corner.

Applying Gels (all light sources)

Special textures may be applied to any light sources. These are called "Gels." When applied, the light source projects the pattern or image present in the gel onto the object(s) that light source is illuminating.

Built-in Gels

Several gels are included with the Vision3d package. They are stored in the Gels folder within the Vision3d application folder. Each time Vision3d is launched, this folder is scanned (along with the Extensions, Textures, Shapes, and Primitives folders) so a list of available gels can appear in a pull-down menu within the Edit Light dialog.

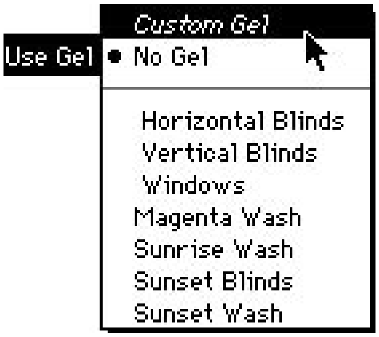


To apply a gel to a light source click on the Use Gel pull-down menu. The gels available during this session appear in this menu. Choosing one from the list automatically loads the gel and applies it to the light source. When you render the model, that light will illuminate with the gel you applied.

As mentioned, a gel is a specialized texture. It contains not only the pattern or surface map to be projected, but also the mapping and scale information that fits the texture to the light source in a manner so it will work correctly the first time. In other words, all the work has been done for you, you just have to use them through the Edit Light dialog.

Custom Gels

There will be times when you want to have a lights source project an image of your choosing. Choosing "Custom Gel" in the Use Gel pull-down menu allows you to create a new surface texture to be used as a gel for the selected light source.



The familiar surface texture editing dialog opens. Typically you would load a color map into this texture to be projected by the light source. You can also use the expert mapping fields to also affect the illumination coming from the light source. The same principles discussed in Chapter 5 — using grayscale maps control surface properties on objects — is also applicable to gel creation. Thus, diffuse and specular maps can also be an effective method of changing illumination intensity in specific patterns that simulate how a light would cast shadows as it shines through a window, door, translucent panel, or other object. Or perhaps you want to project a logo or graphic design onto an object with another texture applied, as shown in the following illustration.

STRATA
VISION3d

Of course, you can get the same result by creating a new texture through the New Texture command (or New button on the Textures palette) and applying it to the local light source manually in a model window. The advantage of using the Use Gel field in the Edit Light dialog is that the texture is automatically applied from the appropriate direction to be projected by the light source as expected. Manually applying a texture to the light source icon requires that you set the application direction yourself, as you would with any other object.

The Edit Light dialog's "Custom Gel" feature allows you to use only surface textures as gels. If you want to use a solid texture on a light source, you will need to create or load the texture and apply it to the light source manually. An example of using a solid texture on a light source would be to apply a RotoScoper texture to play a PICS file or QuickTime movie so the light would act as a movie projector in an animation.

In any case, you would still have to set the mapping style and texture scale. If you do apply a texture in the conventional way, note that when you open the Edit Light dialog for that object, that texture will be represented there as a custom gel.

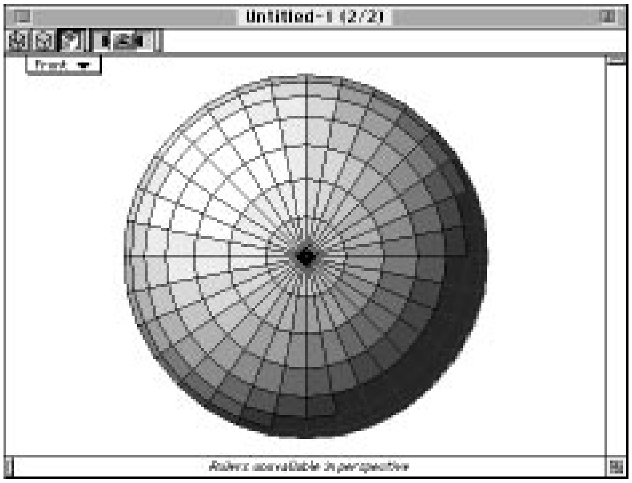
Quick Tutorials

The object of these Quick Tutorial exercises is to help you understand how light sources are placed into a scene and edited, and how different types of lights and lighting techniques can be used to achieve the effects you want to see in a finished rendering.

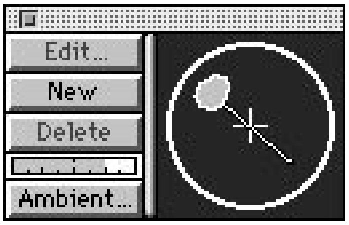
Exercise 1: handling lights

In Exercise 1, you will learn how to place light sources in a scene and how to position them. You'll also see how directional light sources and spot light sources illuminate a scene differently— the first is a global light source whose light falls on all the objects the same way; the second is a local light source with illumination that impacts only a specific area of the model. You will also see how ambient light provides "fill" lighting to soften shadows and dark areas not directly light by light sources.

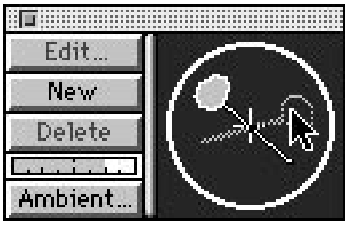
1. **Create a new model.** As before, insure that the lens is set to orthographic and the view is set to Front.
2. **Draw a large sphere in the model window.** The size is not important, just make it a comfortable fit in the model window.
3. **Change the three-position lens control to Normal.** Vision3d redraws the sphere in perspective at a slightly smaller size. Use the "Fit Views to All" command to fit the sphere to the model window again.
4. **Change the Display Method to Shaded Surfaces.** Vision3d again redraws the sphere, but this time with shading. Notice how with the Shaded Surfaces display method you can see light from the default directional light source shining on the top left portion of the sphere.



Look at the Lighting palette (usually positioned in the lower left corner of your screen). There you see the single default light source is positioned to shine down on your model from the "10 o'clock" position.

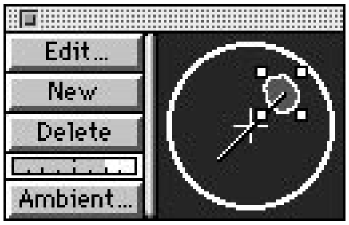


5. **Reposition the light source.** Place the cursor on the light source as shown. Click-and-drag to move the light source to a new position within the lighting circle. When you release the mouse button, Vision3d redraws the scene with the light source illuminating the model from its new position. Reposition the light source several times. Notice how each time you reposition the light source, the lighting in the model changes with the new position.



The circle on right side of the Lighting palette represents the outline of an imaginary sphere that encompasses your model. By positioning a directional light source at a specific point on this imaginary sphere, you indicate to Vision3d from what direction the light source should illuminate your scene in the model window.

6. **Reposition the light source in the rear hemisphere.** Hold down the Control key while you drag the light source. Notice how this moves the light source to a corresponding position behind the object relative to the model window.



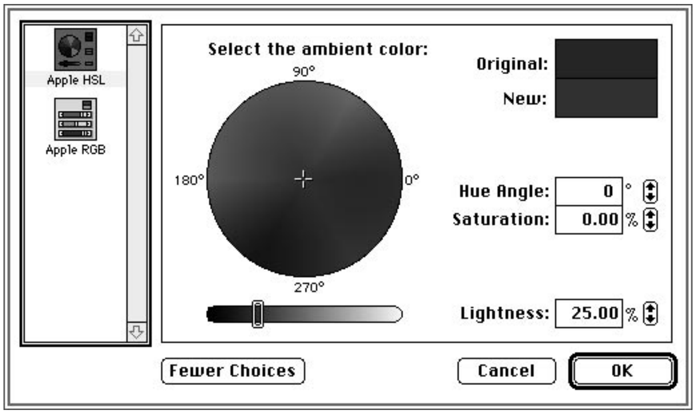
Holding down the Control key while repositioning a directional light source toggles between the front and rear hemispheres of the lighting sphere, relative to the active view. Now reposition the light source again, but this time don't hold down the Control key. Notice how the light source remains in the rear hemisphere. It will stay there until you toggle back to the front by holding down the Control key while dragging the light source.

For now, leave the light source in the rear hemisphere. Notice how the front of the sphere appears dark, since no light source is currently illuminating that part of the sphere.

7. **On the Lighting palette, click on the Ambient button in the bottom left corner.** Vision3d opens the Apple Color Picker dialog box and displays the current color of ambient light in the scene.

The actual layout of the color picker displayed differs with the color depth of your monitor, but they all work essentially the same.

Notice that the ambient light brightness is quite low, but not black. The setting is for a neutral, dark gray color, as displayed in the upper left corner of the dialog window. *This is the recommended setting for ambient light.*



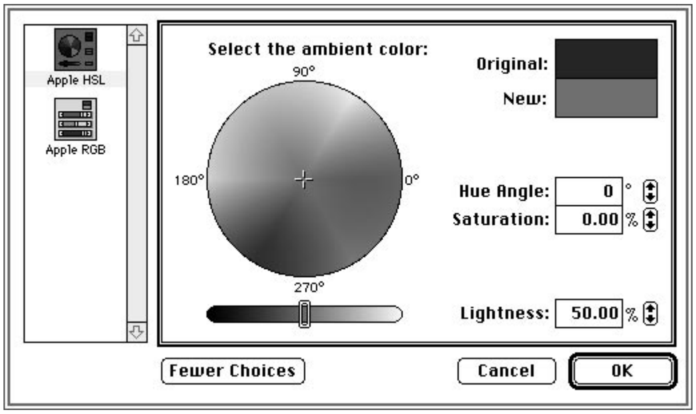
Select the ambient color:

Hue Angle: 0 °

Saturation: 0.00 %

Lightness: 25.00 %

8. **Move the brightness slider in the color picker dialog until it is positioned about midway between the black and white.** Notice the new color you've specified is significantly lighter than the default setting.



Select the ambient color:

Hue Angle: 0 °

Saturation: 0.00 %

Lightness: 50.00 %

Ambient light is non-directional, general illumination that provides "fill" light for dark areas of a model. Ambient light has no source; it's just present in the model. The level of ambient light is controlled through this color picker — the brighter the color, the brighter the ambient lighting in the model.

You may also pick a color other than different levels of gray for the ambient light. This can be useful for special lighting effects in the shadow areas of a rendered image.

9. **Click on the OK button.** Vision3d redraws the model window with the new level of ambient light you've defined for your model.

Notice how your model appears brighter now than it did when the ambient light was set to the darker default color.

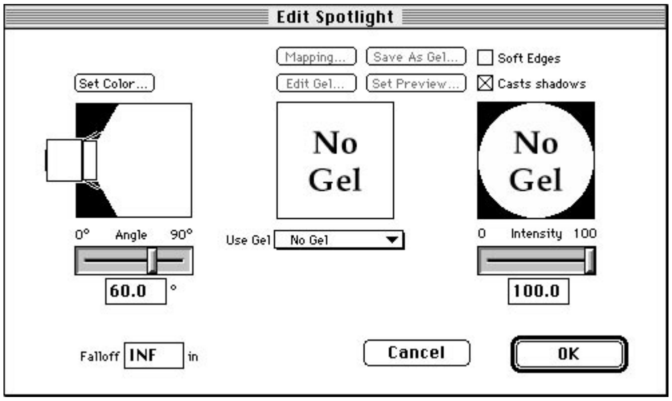
Notice also that the light meter on the Lighting Palette is now into the red, warning of possible over-exposure, i.e. — too much light in your model. To avoid possible overexposure, you must reduce the level of ambient light back to what is was before.

10. **Return the ambient light level to approximately the default.** Refer to steps 7 and 8 if needed to help you recall how to edit the ambient light level. Move the slider back to a position near the default setting and click on the OK button. Vision3d will redraw the scene again with the re-edited ambient light level.

11. **Replace the directional light source with a spot light source.** To do this, first delete the directional light source. On the Lighting palette, click on the directional light source to select it. You know it's selected when four selection handles appear around the head of the directional light. With the directional light source selected, click on the Delete button on the lighting palette. Vision3d redraws the model without the directional source.

    Notice how dark the model now appears.

12. **Place a spot light in the scene.** To do this, choose the Spot Light tool from the Tool palette. Notice how the cursor icon changes to a spot light icon when moved into the model window. Position the cursor near the center of the sphere and click once to place the spot light into the scene.



Edit Spotlight

☐ Soft Edges

☒ Casts shadows

Angle 60.0 °

Use Gel No Gel

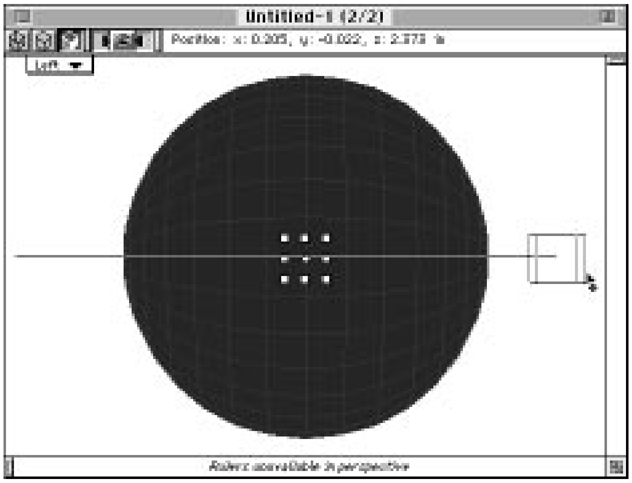
Intensity 100.0

Falloff INF in

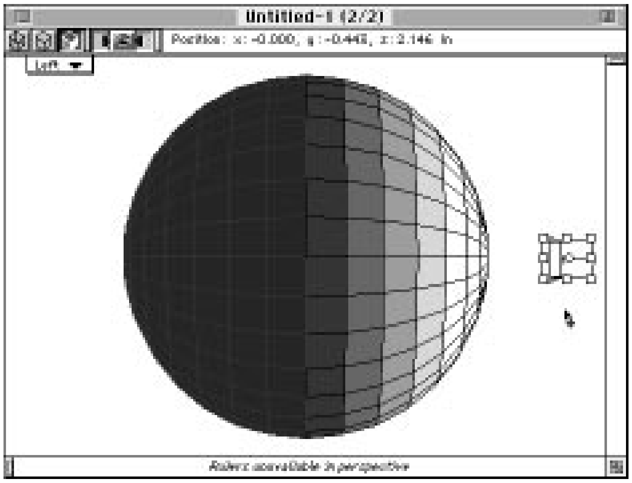
Vision3d opens the Edit Spotlight dialog box. This is where you can set the cone angle, color, intensity, and other parameters for the spot light. Click on the OK button to accept the spot light's default settings.

If you wish to re-open this dialog for a selected spot light, choose the Edit Primitive command from the Modeling menu. Vision3d re-draws the model with the spot light's icon visible in the model.

13. **Reposition the spot light source.** With the spotlight source selected, switch to the Left view. Notice the light source was inserted at the same depth from the Front view into the model. It is occupying the same space as the sphere.


Untitled-1 (2/2)
Position: x: 0.205, y: -0.022, z: 2.373 in
Left
Rulers unavailable in perspective

Choose the Object Move tool from the Tool palette, then click on the right center selection handle and drag the spot light to the right edge of your model window. When you click on the selection handle, a line from the spot light indicates which direction the light is shining.


Untitled-1 (2/2)
Position: x: -0.000, y: -0.448, z: 2.146 in
Left
Rulers unavailable in perspective

After the light source is repositioned, Vision3d redraws the scene again. You can now see the light from the spotlight source illuminating the front of the sphere, as viewed from the left side. By changing back to the Front view, you can see how the illumination of the spot light is casting a circle on the sphere.

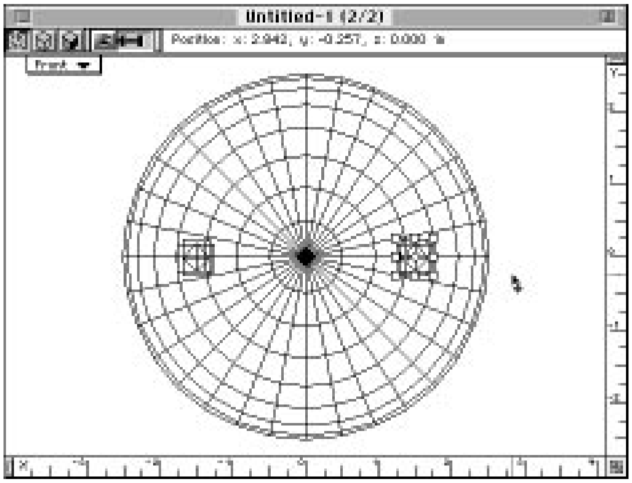
This exercise was intended as a simple introduction to handling light sources. No rendering was required, because the effects of moving light sources was obvious enough in the model window. More advanced features of the light sources in Vision3d may not show until the model is rendered. Some of these advanced features are presented in the next exercise.

Exercise 2: using gels

In this exercise you'll learn how to apply colors, textures and gels to light sources and how to use light sources to project a pattern onto the surface of an object. You'll also learn how to edit light intensity and other lighting parameters.

1. **Create a new model.** As before, make sure the lens is set to Orthographic and the window is displaying the Front view.
2. **Draw a large sphere.** Use the Transform command to set its absolute size to 5" in diameter. Also set the sphere's position to absolute zero on all three axes while in this dialog. Use the "Fit Views to All" command in the Windows menu, if necessary so you can see all of the sphere in the view.

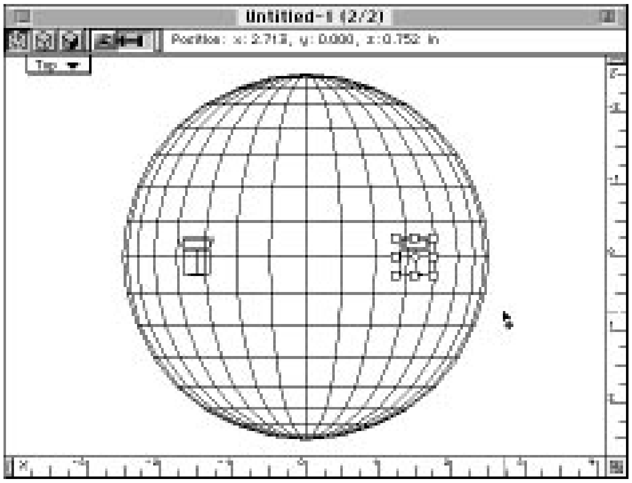
   In this exercise, size is important because multiple light sources will be illuminating different portions of its surface. It must be large enough to minimize overlap of the illumination.



3. **Place two spotlight sources into the model.** Position them on each side of the sphere, as shown in the illustration on the previous page. Refer to the previous exercise if you need a reminder on how to insert spotlights into a scene.

   When you insert each spotlight, Vision3d opens the Edit Light dialog box. Just click on either the OK or Cancel buttons both times when this dialog opens. The spot lights are inserted using the default settings.

4. **Reposition both spot lights outside the sphere.** If you switch to the Top view, you will see that the spot lights were inserted in the model at the same depth as the sphere, so they are actually inside the sphere.



Position: x: 2.713, y: 0.000, z: 0.752 in

NOTE: *You will have difficulty selecting either or both of the spotlights, because they are inside the sphere. There are several ways to select objects inside or behind other objects. You may want to review the sections in chapter 1 on commands provided in the Selection menu (or its sub-menu in the Edit menu on smaller monitors) and in chapter 3 concerning changing selection criteria in the mouse filter dialog.*

One method of selecting objects inside or behind other objects is by changing the selection criteria. Choose the Mouse Filter command from the Selection menu. Vision3d opens the Mouse Filter dialog box. Make sure the Edge checkbox is enabled, and the Surface checkbox is NOT enabled. Click on the OK button. Vision3d returns you to the model window. Now click on the edge of a spotlight to select it. By adjusting the selection parameters, *you can define alternate selection criteria for the cursor, making it easier to select objects that are positioned very close to, or inside of, other objects.*

Another method is to use the Select Next command. Click on the spot light icon. The sphere will be selected because its surface is the first encountered. Choose Select Next from the Selection menu or use its keyboard equivalent, Command-tilde (⌘~) to ask Vision3d to look deeper on the Z axis where you clicked. The spot light should then become selected.

Whatever method of selection you use, remember that you must drag with the selection handles to keep the object selected when it is inside or behind another object. If, after selecting the second object, you click anywhere on that object's surface instead of a selection handle to drag it, the first object will be selected again and it will be moved instead.

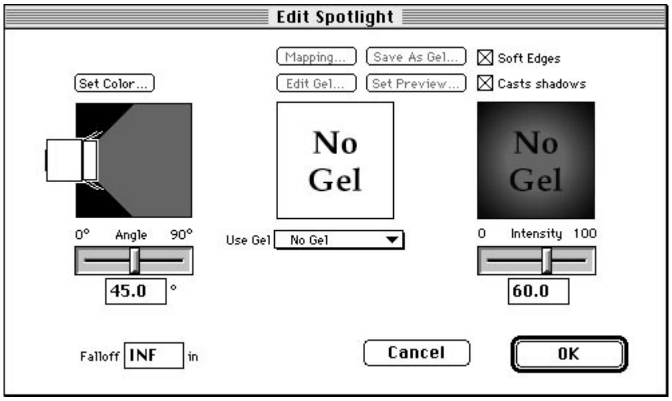
There are several ways to move the spot lights, once selected. The most obvious is to select them in the Top view and drag them downward until they are approximately 1" to 1.5" from the sphere. A second way would be to use the Transform command and simply reposition them numerically an appropriate distance on the positive Z axis.

BUT, YOU WILL LEARN YET ANOTHER WAY HERE.

To move the spot lights for this exercise, first switch back to the Front view if not already there. Select each spot light in turn, then position the Object Move tool's cursor on either top corner handle, then click and drag downward. Watch the feedback area at the top of the model window as you drag until the Z axis reads +3.500 (or as close to that location as the tool can position them). This pulls the spot lights toward the Front view plane on the Z Axis. Refer to the previous exercise if you need help. The discussion in chapter 3 on dragging objects with their selection handles would also be useful

5. **In the Front view, select the left spotlight.** This time the spot light will select as expected because it is outside, or in front, of the sphere.
6. **Choose the Edit Primitive command in the Modeling menu.** Vision3d opens the Edit Light dialog for the selected spot light.
7. **Set the spot light's color to red.** Do this by clicking on the Set Color button. The Apple Color Picker appears. Choose a bright red color and click on the OK button. Vision3d returns you to the Edit Spotlight dialog, where you can see the color you chose for the light represented.

8. **Further edit the light source's illumination parameters.** Do this by matching the settings shown in the following illustration, then click on the OK button. Vision3d returns you to the model window.



Edit Spotlight

☒ Soft Edges

☒ Casts shadows

Angle: 45.0 °

Use Gel: No Gel

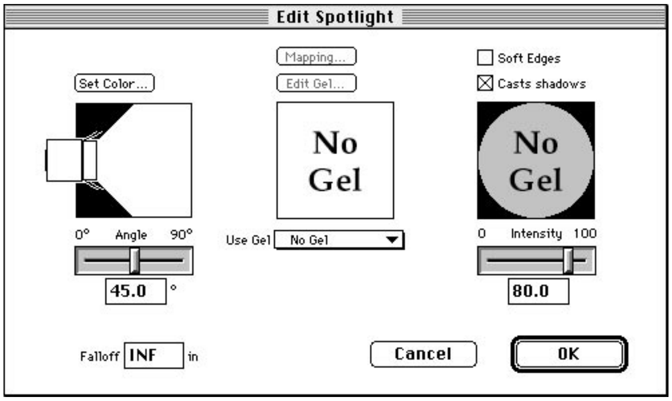
Intensity: 60.0

Falloff: INF in

Change the Angle slider to 45 degrees. This control determines the spot light's cone angle — that is, how wide the field of illumination will be for the light. Enable the Soft Edges check box. Notice how the preview area in this dialog changes from a hard-edged shadow to a soft, feathered edge. Finally, adjust the intensity setting to 60 percent.

9. **Select the right spot light.** Open its Edit Light dialog, again this is done by choosing the Edit Primitive command in the Modeling menu. Alternately, you can double-click on a light source to open the Edit Light dialog box.

10. **Edit the spot light's illumination parameters as shown in the following illustration.** This time, set the Angle to 45° and the Intensity to 80%. Click on the OK button to return to the model window.



Edit Spotlight

☐ Soft Edges

☒ Casts shadows

Angle: 45.0 °

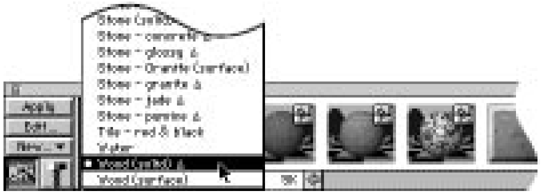
Use Gel: No Gel

Intensity: 80.0

Falloff: INF in

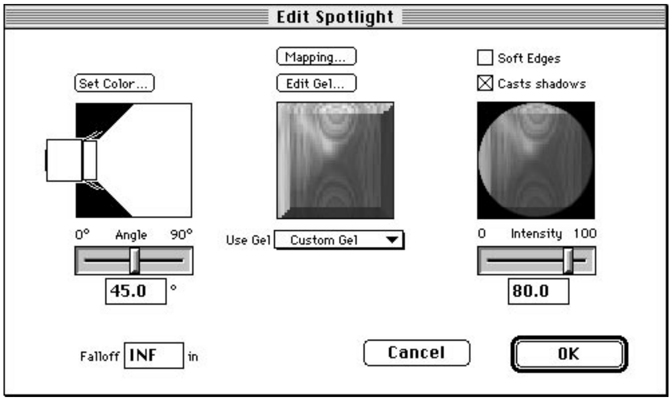
Note that you did not change this light source's color, use a gel, or enable soft edges. You will be applying a texture manually.

11. **Apply the Wood (solid) texture to the spot light.** While the spot light is still selected and the Front view is the active view (so the texture will be applied from the back of the light source), use the Texture palette's pop-up menu to locate the solid Wood texture. Selecting the Wood (solid) texture from the pop-up menu causes the preview images to scroll on the palette until that texture is visible and a black rectangle appears around its preview to indicate selection.



By convention, solid textures provided by Strata Incorporated are marked with a delta symbol (hollow triangle). With this texture selected on the Textures palette (a black rectangle is visible around its preview image), click on the Apply button. The selected spot light-icon changes color in the model window to indicate the color of the texture that has been applied.

12. **With the right spot light still selected, choose the Edit Primitive command.** Vision3d opens the Edit Light dialog box again for this spot light.


Edit Spotlight

□ Soft Edges
☒ Casts shadows

0° Angle 90°: 45.0 °

Use Gel: Custom Gel

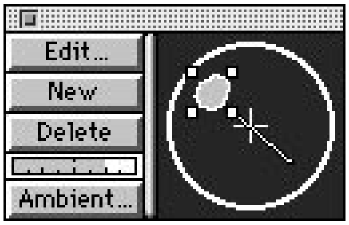
0 Intensity 100: 80.0

Falloff INF in

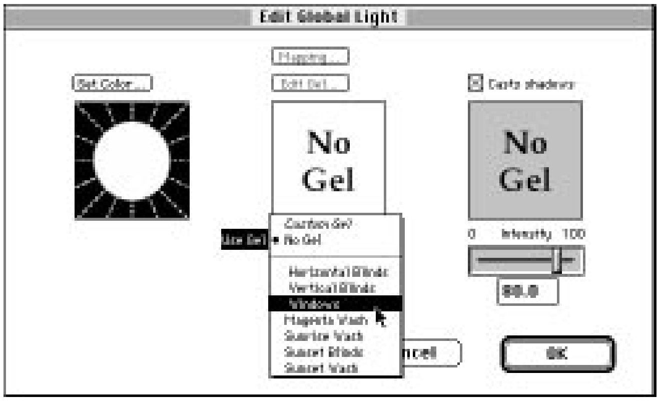
Of course, the previous cone angle and intensity settings are still in effect. But notice that you can see the Wood (solid) texture that was applied in the model window now displayed as a custom gel in this dialog. Click on the Cancel button to exit this dialog. No changes need to be made.

As discussed elsewhere in this chapter, manually applying a solid texture to a light source in the model window is the only way to use a solid texture as a light source gel.

13. **Next, select the default directional light source.** On the Lighting palette, click once on the head of the directional light source displayed there. Four selection handles should appear around the head.



14. **Click on the Edit button on the Lighting palette.** Vision3d opens the Edit Light dialog box for the directional/global light sources. This looks similar to the Edit Light dialog for spot lights, but does not have the cone angle control, the Falloff field, or the Soft Edges checkbox.
15. **Click on the Use Gel pull-down menu and choose the "Windows" gel.** Vision3d displays the Windows gel in the dialog.



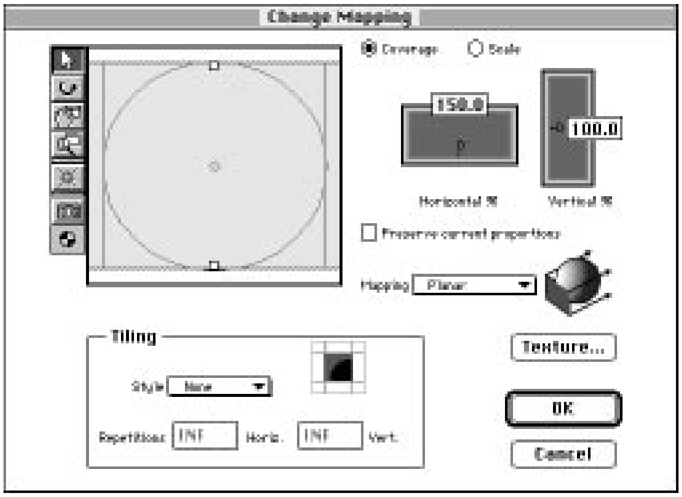
A gel is a special type of texture that is designed for use only to light sources. Gels do not appear in any other menu or list throughout the interface. Several gels are included to provide you with pre-configured special lighting effects.

Of course, you can make your own gels to achieve custom lighting effects. Gel files can include color as well as shadow effects, but most are simple grayscale maps, like the Windows gel listed here in the Use Gel menu in the Edit Light dialog.

16. **Click on the Mapping button at the top of the Edit Light dialog.** Vision3d opens the Change Mapping dialog. This is the same dialog used to edit regular texture maps applied to geometric objects in your model. However, it is used here to edit the gel applied to the directional light source.

    NOTE: *The Camera tool in the Change Mapping dialog is not available when working with a light source. As discussed in chapter 5, this tool is provided to preview how a texture will appear on the selected object so you can make interactive adjustments while in the dialog. It would not be valid to render a texture on a light source, since the texture is intended to be projected FROM the light source, not be visible ON its surface.*

17. **Match the settings shown in the following illustration.** Set Tiling Style to None. Set Mapping to Planar. Set Horizontal % to 150 and Vertical % to 100.



Change Mapping

● Coverage ○ Scale

Horizontal %: 150.0

Vertical %: 100.0

□ Preserve current proportions

Mapping: Planar

Tiling

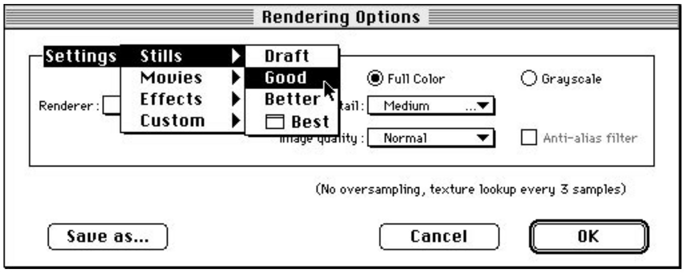
Style: None

Repetitions: INF Horiz. INF Vert.

When the settings are in place, click on the OK button to get back to the Edit Global Light dialog window, then click on that dialog's OK button to return to the model window.

19. **Set up the rendering parameters.** It's time to see the results of your lighting efforts. Because the display in a modeling window is not of sufficient quality to see the lighting effects you have created, it is necessary to render an image of the sphere.

Of course, you have to tell Vision3d how realistic you want the rendered image to be. Since this is just a preview image, you don't want to spend time rendering a better image than you need. Typically, you will render a two or three preview images, fine-tuning the details as you identify problems, then invest time on a final image with all the high quality options chosen.



Rendering Options

Settings: Stills ▶ Draft / **Good** / Better / Best

Movies ▶
Effects ▶
Custom ▶

Renderer :

● Full Color ○ Grayscale

Detail : Medium

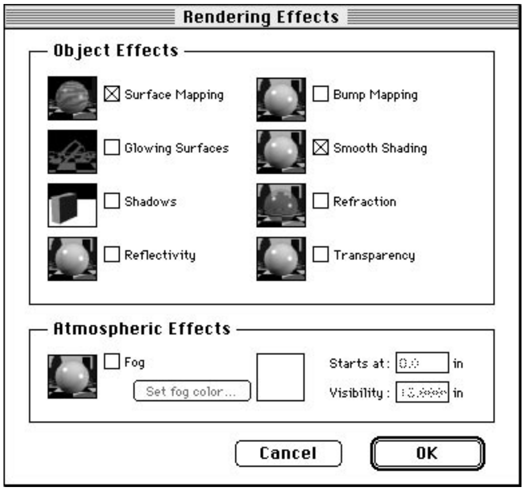
Image quality : Normal

☐ Anti-alias filter

(No oversampling, texture lookup every 3 samples)

Pull down the Rendering menu and choose the Rendering Options command. Vision3d opens the Rendering Options dialog window. From the Settings pull-down menu choose "Stills•Good" as your rendering setting. Then click on the OK button. Vision3d returns you to the model window.

21. **Select your Rendering Effects.** Pull down the Rendering menu and choose the Rendering Effects command. Vision3d opens the Rendering Effects dialog box. Match the settings shown in the following illustration, then click on the OK button. Vision3d returns you to the model window.



Rendering Effects

Object Effects

☒ Surface Mapping ☐ Bump Mapping
☐ Glowing Surfaces ☒ Smooth Shading
☐ Shadows ☐ Refraction
☐ Reflectivity ☐ Transparency

Atmospheric Effects

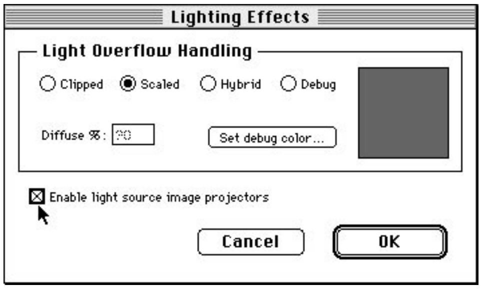
☐ Fog

Starts at : 0.0 in

Visibility : 10.000 in

This dialog is where you decide which effects you want to see in the rendered image. In this case, you want only the Surface Mapping and Smooth Shading effects to be enabled. There is no need to ask for any of the additional effects, since none of them will appear in the rendering.

22. **Set the Lighting Effects.** Pull down the Rendering menu and choose the Lighting Effects command. Vision3d opens the Lighting Effects dialog box. Click on the "Scaled" radio button, and make sure the "Enable Light Sources as Image Projectors" checkbox has an "X" in it, so it is enabled. Then click on the OK button to return to the model window.



Lighting Effects

Light Overflow Handling

○ Clipped ● Scaled ○ Hybrid ○ Debug

Diffuse %: 20

☒ Enable light source image projectors

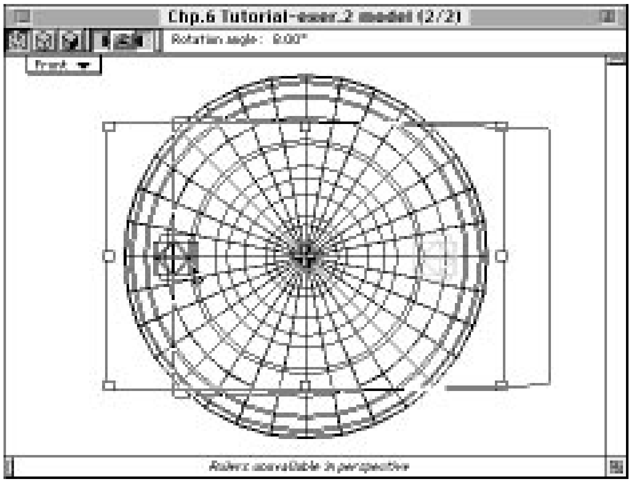
When you have multiple high-intensity light sources shining on one object, such as you do in this model, the scaled over-exposure control method does the best job. It is not as fast as the clipped method, but is much more accurate.

Also, if the checkbox that allows light sources to project images is turned off, none of the effects you have set-up in this model would appear during rendering. Normally, this checkbox should be left on at all times.

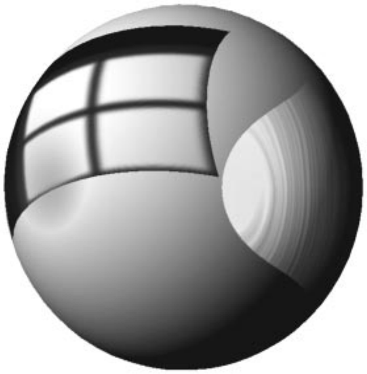
23. **Set the view position.** The final step in preparing for a rendering is to pick a viewing position from which to render the image.

First, set the view to the Front view, if it is not already chosen. Next, switch the lens to Normal perspective. Use the "Fit Views to All" command again to redraw the view at a scale that displays the sphere to maximum size in the view. Finally, rotate the view slightly to the right so you can better see the wood grain projected by the right spot light.

Do this by choosing the View Rotate tool on the Tool palette, then dragging the left center handle on the view management overlay 8° to the right (watch the feedback area at the top of the window as you rotate the view).


Chp.6 Tutorial-exer.2 model (2/2)
Rotation angle : 0.00°
Front
Rulers unavailable in perspective

24. **Render your model.** If not selected already, select the large camera icon from the tool palette. Click once in the model window. Vision3d begins rendering the model.



The finished rendering should look similar to the example shown here, with a spot of red light from the left spotlight with a soft edge, a wood grain pattern projected from the right spotlight with a hard edge, and the window frame pattern projected from the directional light source…all on the same object.

This second exercise has introduced you to gels and their effects on lighting a model. Of course, the rendered image is nothing special, it doesn’t look like anything you’ve seen before. But it does demonstrate the possibilities. Hopefully, you can use the principles learned in these exercises, along with some creative experimentation of your own, to help you achieve nearly any conceivable lighting effect for you own projects.

Commands used in lighting control

The following commands are considered germane to the management of lighting functions in the software. A detailed explanation is included with each command listed.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

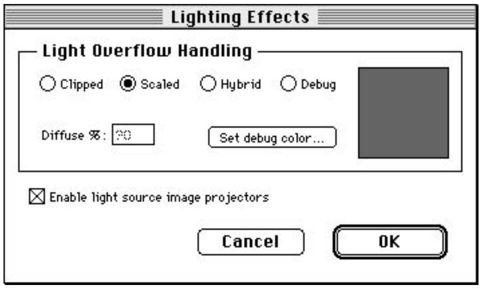
Edit Primitive command (Modeling menu, ⌘B)

Use this command to edit the illumination parameters of a light source. While this command provides some type of editing controls for each type of object, it is especially important for light sources. This command is the primary mechanism provided for adjusting color, light intensity, and many other light source settings.

Each of the three types of light sources has a slightly different set of controls when the Edit Primitive dialog opens, as is the case with all objects. These differences are discussed in detail at the beginning of this chapter.

Lighting Effects command (Rendering menu, ⌘J)

The Lighting Effects command is used to set the method of handling lighting overflow during the image rendering process, as well as enabling or disabling using light sources as image projectors. The default overflow handler is Clipped; the default for using light sources as image projects is enabled. If you want different settings, you can change them in this dialog, but your choice is not remembered between sessions.



Lighting Effects

Light Overflow Handling

○ Clipped ● Scaled ○ Hybrid ○ Debug

Diffuse %: 20

☒ Enable light source image projectors

An explanation of the fields contained in this dialog follows. You may not alter these settings often, but it is important to understand.

Enable Projectors

Click this option to allow projection of gels or other textures applied to light sources. Clearing this checkbox turns off the projection feature globally. Depending on the nature of the model, disabling it may permit Vision3d to execute faster renderings.

Lighting Overflow Handling

Automatic exposure control is a built-in feature of Vision3d. The fields in this section of the Lighting Effects dialog box determine how excess lighting intensity in a model is resolved during the rendering process.

It is possible to over-illuminate an area of the rendering, just like you might see in a real world photograph. This may occur for different reasons:

- **By directing multiple light sources on certain objects**
- **By setting the ambient lighting brightness too high**
- **By giving object(s) surfaces high reflectivity, diffuse and ambient fractions through their applied texture(s).**

The result is a “white out” of the image, similar to an over-exposed photograph. Vision3d has four ways to deal with light overflow problems. Only one of the four choices may be active at one time, with the following result:

- **Clipped:** Ignores all lighting intensity over 100%. This is the fastest method of overflow handling. However, it may cause color shifts in the rendered image. The quantity of light falling on a specific area is based on the total of the red, blue and green components of the light. Excess light is clipped as individual components, then recombined, which tends to affect each of the color components differently. If exact color is critical, clipping the excess light off is not the optimum choice.
- **Scaled:** This method scales total lighting intensity down so highest intensity is equal to 100%. It does not adjust the lighting if no area in the scene exceeds 100%. Scaled is the least likely to cause color shift problems, but takes a bit longer to render.
- **Hybrid:** If total lighting intensity exceeds 100% in a specific area of the model, this method divides the light striking the object(s) surfaces into its diffuse and specular components. It then scales back the diffuse light based on the value in the “Diffuse %” field, and recombines the diffuse and specular components into a recalculated light level. Again, color shifts can occur during re-mixing of the lighting components, but this method gives you a greater degree of control in the final product.

- **Debug:** If any lighting exceeds 100%, this option will show the overflow areas in the chosen debug color when the scene is rendered. You may then adjust the lighting components manually to reduce these “hot spots.” To select the color, click on the set debug color button to display the color palette dialog.

Show/Hide Lighting Palette command (Windows menu)

The Lighting palette is always showing whenever Vision3d is launched. You may have a need to hide or re-show the Lighting palette during the course of a Vision3d work session.

The Show Lighting Palette command displays the Lighting palette. Show Lighting Palette is available when the Lighting palette is hidden. Hide Lighting Palette is available when the Lighting palette is showing. These two commands toggle in the same location on the Palettes submenu of the Windows menu.

Choosing the Hide Lighting Palette command is the same as clicking on the close box of the Lighting palette. Alternately, you can also show or hide this palette by clicking on the Lighting palette button on the Button bar. If needed, refer to *Chapter 1: General Operation* for the location of this button.

CHAPTER 7: SCRIPTING ANIMATIONS

Several textbooks have been written on the techniques of creating conventional animations. Those concepts apply equally well to computer-based 3-D models, though by their nature computer-generated animations are usually assembled from individual frames, rather than using the more traditional cell-based method.

Animation is the perception of motion over time, made possible by a marvelous way our brains process images from our eyes. The effect is called "persistence of vision." In animations, the perception of motion is achieved by presenting a series of finite images in rapid succession where the objects in motion are offset a small amount from one frame to the next. We see the motion because a persistence or residue of the previous image is still fading from our minds as each new image in the sequence is presented.

Vision3d's method for generating a series of frames to make an animation sequence starts with "event markers" placed at intervals along a time line. Based on the spacing of the event markers and the number of frames per second used to create the movement, Vision3d calculates how each in-between frame should render to achieve a fluid motion to the movement of objects within the model. Calculating the in-between frames is called "tweening." The more frames per second, the smoother the animation, and the longer it takes to create.

An important point to remember here is that the eye is very forgiving of lack of detail while an object is in motion. While a single frame of the model might require raytracing to produce an interesting still image, an animation can often be rendered in one of the lesser-quality rendering methods with excellent results.

NOTE: *An example is presented at the first of this chapter to illustrate the effective use of hierarchical animation. It demonstrates recommended techniques for scripting complex motion, using the elementary sun, earth and moon orbital mechanics. It is also used as a vehicle to explain the operation of the Animation palette. You will want to read and follow along to get the most from this chapter. Other tutorials and a control-by-*

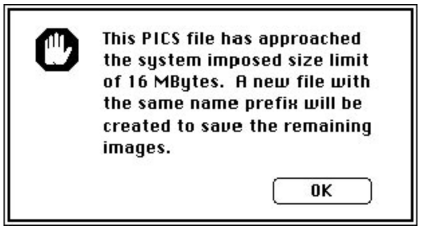
control explanation of the Animation palette are included later in the chapter for review, but this is the most comprehensive description.

Hardware limitations

When creating an animation, the intent is usually to output it to video tape. This may be in the form of raw footage that can be edited and mixed conventionally with other video and sound to arrive at a finished product. Or you may want to keep it in digital form longer to put the finishing touches on your project with multi-media software before outputting. Either way, the ultimate goal is to get it on video tape where you can show it in full motion.

Theoretically, there is no upper limit to the number of frames per second or the finished length of an animation. There are some very real physical limitations, however, related to the current state of the Macintosh hardware and released system software and extensions:

- **PICS Files cannot contain more than 16 megabytes.** Saving a file with more than 16 megabytes of resources to disk can be disastrous to existing files on the same volume. PICS files containing multiple PICT images of an animation can easily exceed that size, because ALL the images are stored in the resource fork. That is the way the PICS format works. For this reason, Vision3d will not assemble an animation sequence in PICS format larger than 16 megabytes.



This PICS file has approached the system imposed size limit of 16 MBytes. A new file with the same name prefix will be created to save the remaining images.

The obvious work around is to save long PICS-formatted animations in segments of less than 16 megabytes. In the process of rendering and saving an animation, if Vision3d detects that a PICS format will exceed the 16 megabyte limit with the next frame, a directory dialog appears asking for the name of a second file.

- **QuickTime Movie files can spool faster from the hard disk because they are compressed.** There are several compression methods available; most of them use a “lossy” format. That is, they discard data from the images to achieve the compression. There is trade off between the picture quality and compression ratio, which are generally inversely proportional. A high quality setting will achieve low compression. Alternately, image quality is usually sacrificed to achieve high compression. While different compression schemes may produce varying incremental changes, the general rule is that quality goes up as compression ratio (and thus playback performance) goes down, and vice versa of course.

- **The maximum number of frames per second that can be played is highly dependent on the frame size, the color depth, and the hardware architecture.** Obviously, a sequence of smaller one-bit (B&W) images will spool from disk onto the screen faster than a set of larger 32-bit (full color) images.

CONSIDER: *A simple stream of moving color images can quickly become a torrent of bytes that brings most Macintosh hardware to its knees. For example, to play an animation made of 32-bit, 72 d.p.i. color images the size of a 13 inch screen at 30 frames per second requires data to pour in from the hard disk at a rate of 20 to 36 megabytes per second. Even the lower end of this range is more than can be handled by any but the fastest Macintosh hardware. It is faster than can be handled by the standard SCSI connection; It is faster than Color QuickDraw can display it on the screen; It is even faster than the data can be transferred to video cards through the NuBus architecture. And if you want to add sound to the animation, the discrepancy in capacity gets even more absurd. Without specific hardware support, full-size, full-color, full-motion video is not possible in real time.*

Obviously, faster platforms will be built and quicker transfer rates will be provided to better support the development of true multimedia production. There are hardware alternatives available now — such as CPU accelerators, real-time data compression/expansion cards, QuickDraw video accelerators, faster storage/retrieval devices, SCSI-2 ports (with both Fast and Wide capabilities), separate I/O handlers, etc.

This discussion about limitations is presented to give you a more realistic expectation about what you can do in your “cinematic debut.” Vision3d’s role in producing the finished animation is to assemble 3-D models and render raw footage of them in motion or with various animation effects. No tools are provided to edit the animation once it is rendered. Those are available in other applications intended for that purpose.

A more economical software approach is to buffer the full animation in RAM before playing, but that requires memory, and lots of it. Playing from buffer usually cannot play at full-size, either.

While digital animation-editing tools are improving all the time, professionals usually prefer to output an animation sequence one frame at a time to video tape, then edit the finished animation on video. This requires a “frame accurate” video controller and compatible tape deck. While still relatively expensive, they have become more affordable in recent years.

Do not expect to become the electronic equivalent of John Ford overnight. That will take some work and several projects. Besides, anything you can do on a Macintosh has probably been done better by George Lucas and ILM (Industrial Light & Magic, Inc.) using high end, specialized equipment with LOTS of people and multi-million dollar budgets. That said, use your ingenuity and prove us wrong!

Yes, you can create broadcast-quality animations on the Macintosh; you will be able to amaze your friends; and you will have lots of fun (along with some frustrations) in the process. Understand that it takes more software than Vision3d to produce finished animations (with sound, special effects, A-B rolls, etc.). And even though animations can now be done entirely on computer, quality work still requires skills in the traditional areas of storyboarding, cinematography, and scene direction. *But if it was easy, everyone could do it and you couldn’t make any money!*

Concept of actions in creating motion

Complex motion does not happen by accident. It must be planned and deliberately scripted. Executed ineffectively, at best the result is often just a busy screen. At worst, the objects are not well controlled and the project is usually abandoned before completion because “it can’t be done.” Done well, however, the animation tells your story and you are commercially rewarded.

Ironically, the best tool you have for creating award-winning computer animations is not on the computer…it’s paper and pencil. Investing the necessary time BEFORE you begin a project to write down (or draw) a storyboard and work out all the details, including special notations and time indexes, can be your greatest ally and help you keep focused when you are in the thick of the project.

Actions that are time-dependent

In animation terminology, “action” refers to motion. The motion may be either with the objects, the camera, or both. An action is a single movement over a specific sequence of frames. Vision3d measures the following actions as it interpolates, or ‘tweens frames for an animation sequence:

- **Birth and death.** An object may appear and disappear at specified times,and as many times as you wish.

- **Position.** An object may move from one location in 3-D space to another over time. Directional and local light sources may be moved the same way as geometric objects. The speed of movement is determined by the distance and the time given for it to move. The path or action may be linear or spline.
- **Rotation.** An object may rotate on any axis or combination of axes over time. The speed of the rotation is determined by the angle specified between event markers and the time given for the rotation. The rotation action may be linear or spline (while a little harder to conceptualize, the same tweening parameters applicable to position also apply to rotation).
- **Scale.** An object may be scaled larger or smaller over time. The speed of a scale change is determined by the distance and the time given for it to occur. The action may be linear or spline.
- **Texture.** An object or light source may change its texture over time. It may change as often as desired. The timing of the transition(s) are controlled by the placement of event marker and their type (linear or spline).
- **Camera focal length.** A camera object may change its window's focal length over time. Again, speed and path parameters apply to this action just as they do to movement, rotation, and scale.
- **Light intensity.** Light source objects may change intensity over time. This includes directional/global lights as well as the local light types.
- **Environment.** The environment may be changed over time by using the Rotoscoper solid texture extension to "play" an animation as the image shown or reflected. While both the background and reflected environments are aware of time, note that you cannot simply change textures over time as you can with an object or light source. You must use the Rotoscoper extension. Please refer to *Chapter 5: Using Textures* for more information on the Rotoscoper extension.

Multiple object and camera actions may occur simultaneously during the same frame sequences. You may use as many of these action as needed in the sequence to create the animation effects you desire.

The following parameters are used in an animation, but not 'tweened, i.e. — they are manually set for the entire sequence:

- **Frame size of rendered animation.**
- **Viewing position (in model windows).**
- **Object geometry.**

From the first list, you can see that Vision3d is able to monitor quite a few actions in its tweening calculations to produce full object and/or camera choreography. However, no time dependencies are attached to the items in the second list. They are considered static and must be set before each rendering session.

Minimum requirements for animation

The following points are presented as suggestions for how to accomplish some of the basic effects available in Vision3d

- **A minimum of two event markers at different points along the time line are required for animation.** There is no limit to the number of event markers that may be defined, as memory allows. During preview playbacks or the high-level rendering of each frame in an animation, Vision3d then moves each animated object from one event marker to the next, yielding as many positions between the event markers as requested by the frames-per-second setting.
- **Rendered animation sequences may be generated from any view.** The active view determines the viewing position used for the rendered frames of an animation. You may use a view in a modeling window or the view from a camera window. In the case of a modeling window, no view movement is possible, since the viewing position in a modeling window cannot be moved over time; it is fixed. Camera objects are provided for this purpose.

  To add camera motion to an animation, first insert at least one camera object in the model and give it an animation path just like any other object. Then, use the camera window attached to that camera object as the source of your animation's images. Since the camera window shows the view of a model from the selected camera object, the viewing position of the animation will change as the camera moves. You may wish to do the same animation sequence from several different viewing positions — from static views and from moving cameras — then combine the rendered sequences in an animation-editing application for the finished "movie."
- **Life of object.** If an object is added to the model while the time pointer is at a position other than time zero on the time line, that object is assumed to have existed in that same position from time zero. This is the default for each object. However, you can designate birth and death event markers for an object, causing an object to appear and/or disappear .during the animation.

  Another method is to create and apply a fully-transparent texture to an object. The texture applied to an object may be changed over time. Thus, it would be possible to hide an object by making it completely transparent, then change to another texture that would cause the object to become visible. This is useful if you need the object to be moving at the moment of birth. Birth event markers do not allow you to have any action scripted prior to an object's birth.

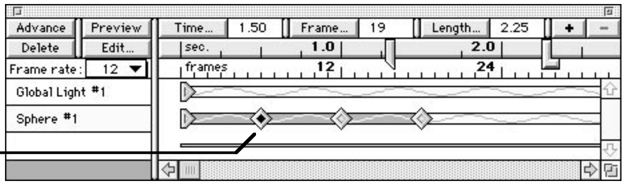
- **Editing textures during surface metamorphosis.** Textures may be changed on an object over time. During the transition from one texture to the next, there will be some frames in which the object will actually have two textures applied to it at the same time. This can be a useful technique, even for still images. But, it also means that the "Edit Selected" command in the Textures menu will not work for the object while in metamorphosis. Basically, it does not know which texture to edit. When that situation exists on the time, an alert will appear.

  You can, of course, still edit any of the individual textures involved in a metamorphosis through the standard texture-editing commands, including those found in the Textures browser. All Textures are considered to exist in their current definition from time zero. Their use is time-dependent, but the textures are not.

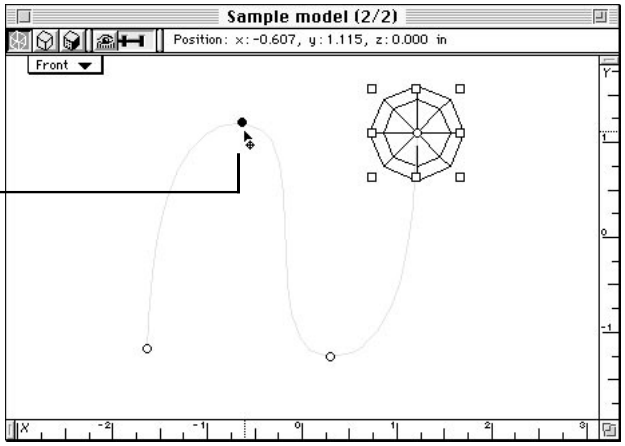
Event marker control options

Vision3d calculates the motion of objects by the control of their event markers as they follow a motion path. The setting for all event markers is to construct a spline path, with smooth transitions between each marker. However, event markers may be individually edited to alter their control of the line as it passed through that "event."

Editing an Event Marker



1. *Double-click on the marker itself*

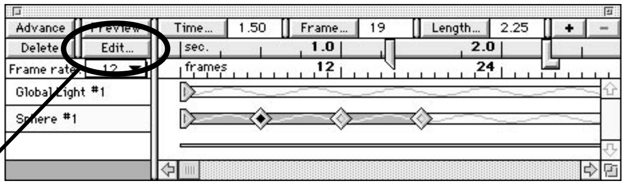


2. *Or double-click on the corresponding anchor point on the motion path*

   (The Show Animation Paths command in the Animation menu must be enabled)

An event marker may be edited by double-clicking directly on the event marker visible on the Animation palette or by double-clicking on the corresponding anchor point visible on the motion path (when Show Animation Paths is on) in the model window.

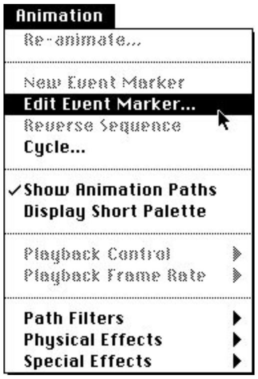
3. *Or select an event marker, then click on the Edit button on the Animation palette*



Alternately, you can select the event marker then click on the Edit button in the upper left corner of the Animation palette. You can also select the event marker or anchor point and choose the Edit Event Marker command from the Animation menu.

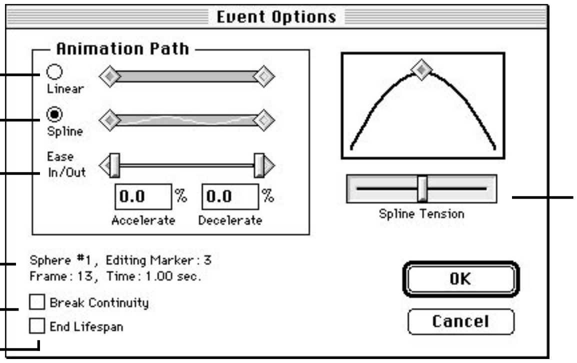
Notice that selecting the event marker automatically selects the appropriate anchor point in the model window, and vice versa.

4. *…Or select the event marker, then choose the Edit Event Marker command in the Animation menu*



Any one of these methods opens the Event Options dialog box. It contains controls for altering the behavior of an event marker. Each event marker is customizable in this manner to achieve the desired object behavior along the motion path.

NOTE: *Motion paths are obvious and easy to understand when an object moves through 3-D space. A line can be seen showing exactly where and how the object is moving. The concept of a “path” is just as valid for actions such as rotation, scale, texture morphing, focal length , light intensity, etc., even though they do not leave a visible path in the model windows.*



Event Options

Animation Path

○ Linear

● Spline

Ease In/Out

0.0 % Accelerate

0.0 % Decelerate

Spline Tension

Sphere #1, Editing Marker: 3
Frame: 13, Time: 1.00 sec.

□ Break Continuity

□ End Lifespan

Linear path button

Spline path button

Ease in/ease out

Object scripting information

Continuity checkbox

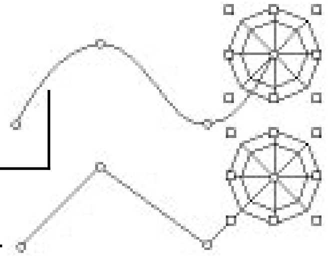
Death checkbox

Spline tension slide control

The settings of an event marker control the motion path between this marker and the next one on the time line. The effects of these controls on an object's motion are as follows:

- **Spline/Linear path.** Use to make the path through the event marker a smoothly flowing motion or a straight, angular movement.

Paths are generated based on settings in event markers:



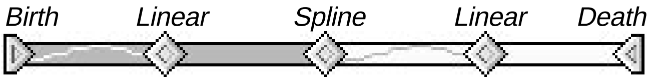
Spline event markers

Linear event markers

The type of marker — spline or linear — is also echoed on the Animation palette. Spline paths are indicated by a wavy line between event markers, as set in the Event options dialog. Linear paths are indicated by the absence of a wavy line between event markers.

Four main types of event markers

…and variations on their path control

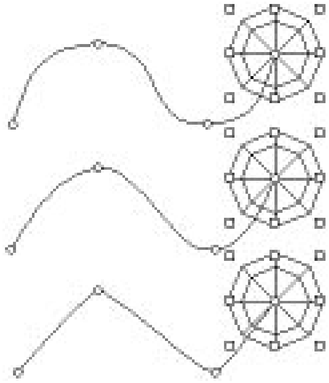


Dk. green: change	*Dk. green: change*	*Lt. green: no change*	*Lt. green: no change*
Wavy line: spline path	*No line: linear path*	*Wavy line: spline path*	*No line: linear path*

The color between event markers is closely-related to the type of marker used. The above illustration identifies time-dependent changes in an object by a dark green color. If there is no "tweenable" changes in an object from one marker to the next, a light green color appears. Thus, depending on the action scripted and the type of event marker used, the behavior of the object may be different. (On grayscale monitors, the colors become dark gray and light gray. Monochrome monitors show black and white. The wavy line is inverted in the black so it can still be seen.)

- **Spline Tension.** Use to adjust how taut a spline path flows from one event marker to the next. If the event marker is designated a linear marker, this control is dimmed.

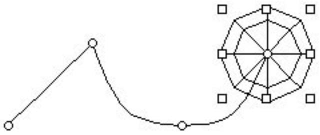
Spline Tension



- **Ease in/Ease out.** Use to control the acceleration and deceleration "curve" of an object's motion between event markers. In a typical situation where an object is starting and stopping, the first event marker will use most of the time between it and the next event marker to accelerate. The in-between markers will not use any ease in/ease out. Then the next to last event marker will use a lot of deceleration as the object comes to a stop at the last event marker. This imparts a stronger sense of mass to the object and inertia to its motion.
- **Break Continuity.** Use to break the smooth flow of a spline path when you want the object to make an instantaneous change of direction. This checkbox has the effect of making that event marker a hinged anchor point on the motion path.

Continuity control

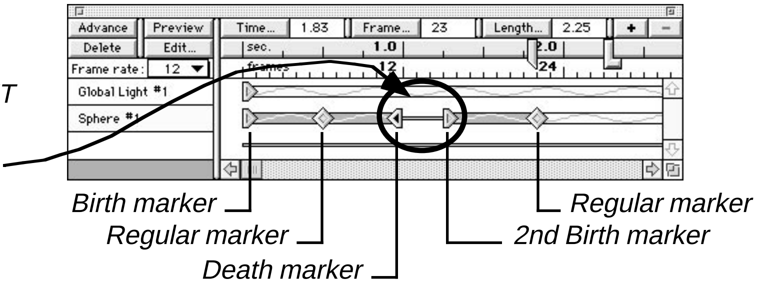
Second marker on this spline path has "Break Continuity" field on



- **End Lifespan.** Use to make the selected event marker a death marker. The object disappears from the model after this event marker.

Lifespan control event markers

OBJECT DOESN'T EXIST DURING THIS INTERVAL!



When an object ends, its spot on the time line appears empty. While in the model space, it is as if the object has ceased to exist, internally the object is still in the model's database. Notice that an object can die and return to life again. The scope of this manual does not address the philosophical and religious meaning of this feature.

Using hierarchical scripting

Individual event markers in Vision3d can accept multiple actions for an object, such as changing position and rotating at the same time. This is often how animations are developed. However, it can sometimes be difficult to go back later to alter or change one of those actions without affecting any other actions that may also be part of the same event markers. That is when animations become unwieldy and difficult to get right.

The trick is to identify the separate actions making up each object's motion, complex or otherwise, then script those actions in a coherent manner. It would be much more preferable if you can script the individual actions separately, so that if the need arises you can edit them confident that you will not impact other aspects of the animations that are right. Vision3d provides a mechanism for doing just that, but it requires some planning.

Once you have the animation on paper, it is usually a simple matter to break down the motion for each object into a series of simple actions. Do you remember the discussion in chapter 4 about the ability to model complex geometry using Shapes and instances of shapes? How you could build complex objects and/or scenes by assembling simpler objects that you could work on in their own isolated workspace? That same concept can also be applied to animation actions. You can use various shape levels to script simple actions; adding them together to form the desired complex motion.

An example:
scripting simple actions at different hierarchical levels

Suppose you want to create an animation illustrating the orbital relationships between the Sun, Earth, and Moon. The motion of the moon must work so that as it rotates on its own axis, it also rotates around the earth AND moves with the earth as it rotates around the sun. Even such a simple example as this can be rather intimidating if you try to do it all in the model space. But using different levels of shape workspaces to script the rotation of the moon and earth can make the project much simpler.

To keep things simple, this project will not address the application of textures. After the model is done and a first rendering of the animation is completed, you can go back and apply textures for a more complete look. For the moon and earth, this would be done in their root shapes. The sun is just a sphere in the model, so that’s simple enough.

One more point. Since the actions you will script in this exercise are essentially all rotations about an axis, there will not be any visible animation paths. As discussed earlier, the paths are there, but since the origin points about which the objects rotate or orbit do not move, the paths do not show. The exercise is still valid, and you can still apply the concepts to your own projects where object motion will leave a path.

The planning

At the beginning, you must decide the size of the objects. This is required so you can then decide the diameter of their orbits. In this example, the sun will be ten inches in diameter, the earth will be three inches in diameter, and the moon will be one inch in diameter. Further, the earth’s orbit will be scripted at 50 inches and the moon’s orbit set to 15 inches in diameter. Obviously, these objects and orbits are not to scale, but then what example of the Solar System ever is? It would be possible in Vision3d to build this project to scale, since there is no boundary to its space. You might have some concern with the software’s math precision at that scale, however. And you definitely would have trouble seeing the earth and moon from any comprehensive viewing position and scale setting.

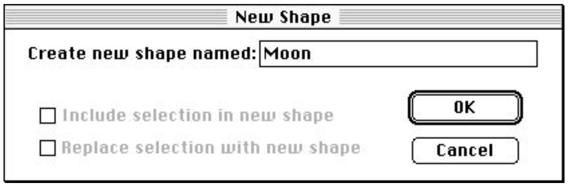
Next, you must decide how long to make the animation so you can set the rotational and orbital periods to complete within that time frame. It would be ideal if you could render the animation so that it loops without a seam. That would allow you to play it continuously; giving the effect of a very long animation.

In this example, the animation will be nine seconds. During that interval, the earth will rotate once around the sun in that time (one year…talk about compressing time!). The moon will rotate six times around the earth or once every 1.5 seconds. Okay, so the moon actually orbits the earth a little more than thirteen times a year, but in this project, whipping the moon around the earth that many times in nine seconds will cause temporal aliasing (or jumps in the motion). Slowing things down a bit will make a smoother, more pleasing, and more understandable animation. (Maybe you could script a version that is more accurate after you see how this is done.)

Although not necessary, you may also want to make a conscious decision to center all orbits at 0,0,0 in their respective workspaces, so you can know the offset for the radii of the object's orbits. It will make the animation scripting a little easier if you are consistent in this respect.

The Moon

Begin by creating a new model. Within the new model, create a new shape. Enter the name of this new shape as "Moon-rotating." This is a root shape. At each instance (or level change) of the moon, you will add more action; building up the complex motion that the moon will be required to perform.


New Shape

Create new shape named: Moon

□ Include selection in new shape

□ Replace selection with new shape

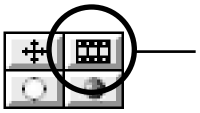
Within the Moon-rotating shape, switch the view to the Top, then draw a sphere one inch in diameter. Position it at the center of the workspace (X=0, Y=0, Z=0). Notice that the sphere is rotated 90° so that the poles are oriented top and bottom, when looked at from the Front view. Please refer to chapter 3 if you need help doing this task. (Here's a hint: Use the Transform command in absolute mode. If you don't know how to do something as simple as positioning an object in space, perhaps this is not the chapter you should be reading right now!)

So far so good. This root shape level is where you will give the moon rotation on its own axis. From your Junior High science class, you remember that the moon rotates on its axis in exactly the same period as it takes to orbit the earth. (That's why it was not until the 1960s that humans were able to get pictures of the back side of the moon to see what it looked like, because only one side was ever visible from the earth!)

To script the moon's rotation:

1. **Show the Animation palette, if not already present on the desktop.** This can be done through the Palettes sub-menu in the Windows menu. Alternately, you can click on the Animation palette's show/hide button on the Button bar.

Clicking on the Animation palette button in the Button bar toggles it to show or hide

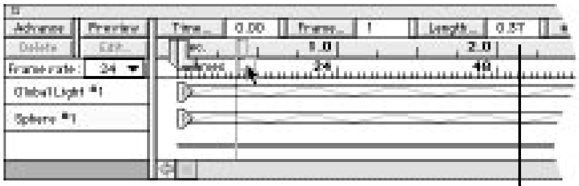


Animation palette's show/hide button

The show or hide status of the Animation palette is retained from session to session. If it was showing when you quit Vision3d the last time, it will be showing at the start of the next session, and vice versa of course.

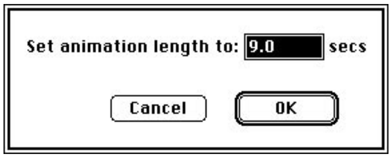
2. **Click and drag the Animation Length Pointer to nine seconds.** The palette automatically scrolls horizontally as you drag the length pointer out to the nine second mark.

To set the animation length for the document, drag the L-shaped pointer



Notice how the Length field at the top of the palette indicates the position of the Length pointer on the time line

This sets the animation length for the entire document. Notice how the Length field at the top of the Animation palette indicates the position of the length pointer as you drag and release.



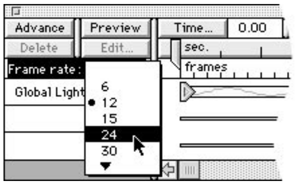
Set animation length to: 9.0 secs

Cancel OK

Alternately, you can click on Length button and enter the animation length numerically. The length pointer moves to the value set in the dialog box.

3. **Set the frame rate to 24 frames per second.** Click on the Frame fields pull-down menu on the left side of the palette and choose 24 from the list of options. The default frame rate for a new model is 12 frames per second — usable for previewing, but not smooth enough

for finished work, or situations like this project where a lot of motion is packed into a short duration.

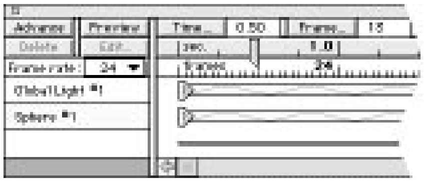

Time... 0.00

Notice how the frame count changes along the time line on the palette to correspond to this new setting. For nine seconds of animation, you are now working with 216 frames that must be rendered to create the animation images. Of course, the higher the frame rate, the smoother the motion.

NOTE: *If you were intending to eventually output this animation to video tape, you would need to set the frame rate to 30 frames per second, or 270 total frames for the nine seconds specified in this project.*

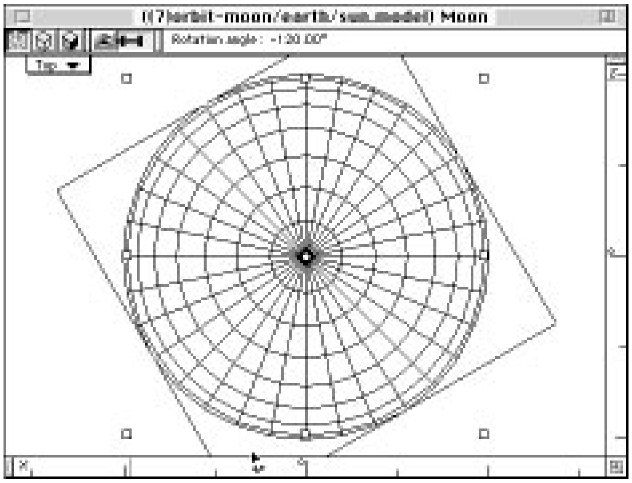
4. **Select the Object Rotate tool from the Tool palette.** Please refer to chapters 1 and 3 concerning the location and operation of the Object Rotate tool.

5. **Move the current time pointer to the half-second mark on the time line.** The time pointer will snap to each major tick mark on the time line as it gets close. You can check the time field, which indicates the pointer's position as you drag and release.


Time... 0.50
Frame... 12
Frame rate: 24

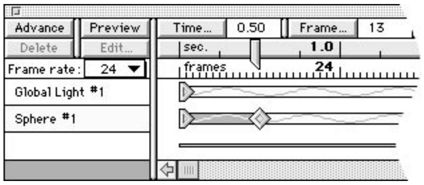
Alternately, you can click on the Time button to enter the desired current time directly. The time pointer moves to the value set in the dialog when you click on the OK button. There is also an Advance button on the top left of the palette. Clicking on this button moves the current time pointer to the next whole second on the time line, which is too far for this step but may be useful in other situations.

6. **Rotate the sphere 120° counter-clockwise in the Top view.** Do this by positioning the Object Move tool's cursor over the top left corner selection handle of the sphere. Click and drag downward until the object is rotated 120°. Watch the feedback area at the top of the window as you drag.



Rotation angle : -120.00°

Releasing the mouse button executes the rotation and causes the view to redraw and an event marker is automatically inserted on the Animation palette for the sphere object at the current time.

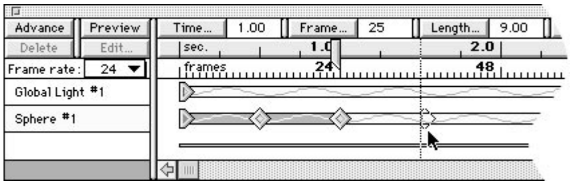


Time... 0.50
Frame... 13
Frame rate : 24

The area between the event markers at time zero and at .5 seconds changes to a dark green (on color monitors, of course) to indicate there is some action taking place. A light green color means there is no change in the object between adjacent event markers.

7. **Repeat the rotation another 120°.** This means moving the time pointer forward another half-second (to the full second mark), then rotating the sphere 120° again in the same direction. A third event marker appears.

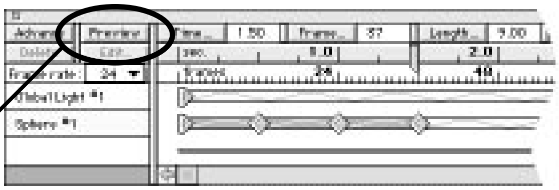
8. **Duplicate the event marker at time zero and position it at 1.5 seconds.** This is done by holding down the Option key while dragging the time zero event marker. The original marker is left where it is and a copy is dragged to the new location.



Time... 1.00
Frame... 25
Length... 9.00
Frame rate : 24

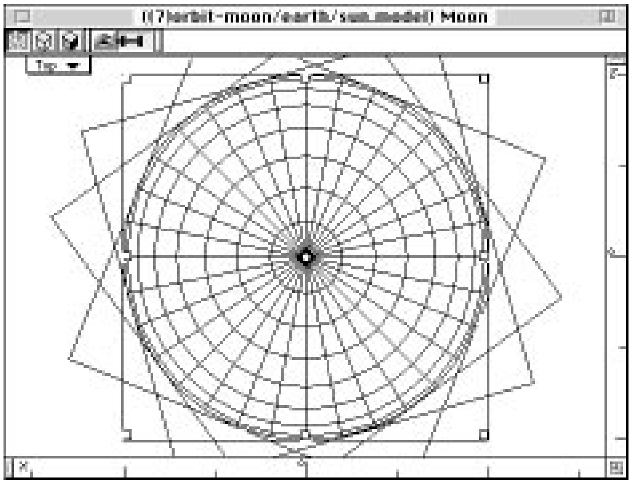
Releasing the mouse button drops the duplicate event marker onto the time line. Notice that you can drag event markers past the current time pointer, but not the Length pointer. Also notice that the time pointer does not automatically move to the time of the newly-inserted event marker.

Why use a duplicate of the first event marker? Yes, you could have simply rotated the sphere another 120° to complete the 360° rotation, and it probably would have worked just fine. However, by using a copy of the event marker at time zero, which is the position of the sphere BEFORE it starts rotating, you can guarantee that the sphere's rotation will return to EXACTLY the same location. That further guarantees that the rotation will be seamless, without a hitch or hesitation.



Clicking on preview button causes a bounding box run through of the animation script

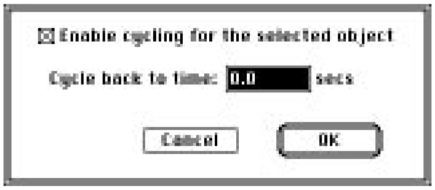
To double-check the rotation script, click on the Preview button at the top left of the palette. You should see a bounding box rotate around the sphere in the shape window.



It should rotate smoothly for one full 360° interval, then stop as the preview proceeds along to the nine second limit. Any time during a preview run through — in this case, wait until after it has passed the

1.5 seconds mark — you can press the Command-period (⌘.) key combination to stop it.

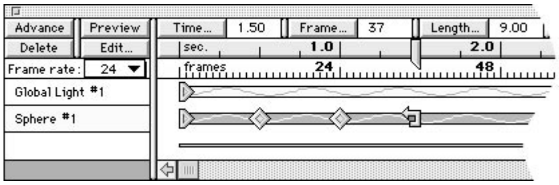
9. **Move the time pointer to 1.5 seconds, then choose the Cycle command in the Animation menu.** Click on the “Enable cycling…” checkbox and set the cycle back field to 0.00 seconds, if not already there, and click on the OK button.



☒ Enable cycling for the selected object

Cycle back to time: 0.0 secs

The last event marker will change appearance to indicate cycling is enabled for this object. You can disable the cycle feature, if needed, by choosing the Cycle command again and turning off the checkbox.



Time... 1.50 Frame... 37 Length... 9.00

Frame rate: 24

Notice how the dark green “action line” now continues past the last event marker, providing further visual feedback of the scripting in place for this object.

Cycling is a very useful feature that eliminates the need to manually script a repetitive action, such as rotation, for the full length of the animation. By laying down the event markers for a single iteration of the action, then turning on cycling, the object will repeat the action *in perpetuity*, or the animation length whichever comes first. If you make the animation length longer, there is no need to re-enable the cycling.

Clicking on the Preview button now shows that the sphere continues to rotate AFTER the last event marker for it, just as if you had scripted the rotation manually.

This concludes the work you need to do in this shape workspace. You can close the moon’s shape window to make room for other windows. Of course, the shape is still part of the model, even if its window is not open on the desktop.

Now would be a good time to save your model, if you have not already done so. As with any application, it is always a good idea to save often, especially after completing a major phase of the project.

The Earth

The next phase starts by setting up a new shape for the earth object. Title it, "Earth-rotating." This, too, will be a root shape where you will script the rotation of a sphere on its own axis. Notice that the event markers entered for the moon object are no longer visible. In fact, the sphere 1 object is not even listed on the Animation palette. While some of the controls on the Animation palette are global and have effect for the entire document (such as the Animation Length pointer and the Frames per Second setting), other controls are applicable only to the current level of the document. Because you are now in a different workspace, the work you did in the other workspace or level is not displayed here. Be assured that the work is still in place, and would reappear on the palette if you were to open the Moon-rotating shape window again.

Before you do anything in this shape workspace, move the current time pointer back to time zero. THIS IS IMPORTANT! If the time pointer is at another point on the time line when you draw the sphere, the object is automatically assumed to have existed in that state as it was drawn from time zero. Then you adjust its size and position to begin the rotation scripting. The event marker when you added the object will change, but not the one at time zero. It is important that you begin factoring time into your modeling techniques or you will get little surprises that will be hard to recover from unless you understand the "big picture."

As expected, draw a sphere in the earth's shape workspace. As discussed at the beginning of the project, make this sphere three inches in diameter and position it at 0,0,0 in the workspace. Using EXACTLY the same steps outlined above for the moon's rotation, script a 360° rotation for the earth. The one exception is that you want the earth to rotate once every second. Thus, you must set your event markers for 120° rotation each at .3333 second, .6667 second, and one second, respectively. This corresponds to frame 8, frame 16, and frame 24 of the animation. Of course, the cycle should be enabled from one second back to zero second. (This means the earth will only rotate nine times in a full orbit, again for visual clarity at the expense of accuracy.)

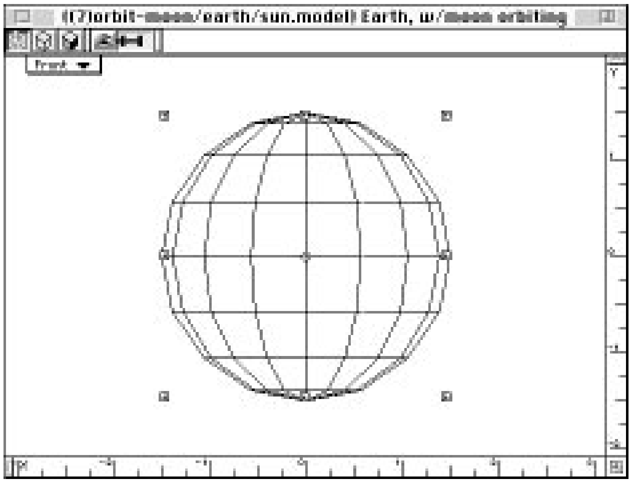
When you are satisfied everything is working as it should, i.e. — just like in the moon workspace only with a slightly faster rotation, close the window. Be sure to save the model periodically, and before moving to the next part of the project.

The moon's orbit

In this phase, you will bring the earth and moon together, then lay down event markers that describe the moon's orbit around the earth. This is a second action added to the previously-scripted rotation. This is where things begin to get interesting. They can be hard if assembled wrong, but ridiculously easy when done correctly.

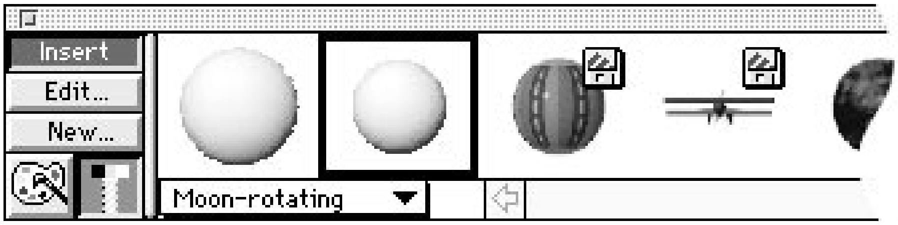
To script the moon's orbit around the earth:

1. **Begin by creating another new shape workspace.** Title it, "Earth, w/moon orbiting." A new shape window opens. As before, be sure to return the time pointer to time zero BEFORE putting anything into this workspace.
2. **Insert the earth into this shape's workspace.** This is done by choosing the Insert Shape command from the Animation menu. Highlight "Earth-rotating" in the sub-menu and release the mouse button. Move the "loaded" cursor to the active shape window, position it at X=0/Y=0 and click once. This inserts a copy of the earth shape at the same size. Just to make sure, use the Transform command to fine-tune the object's position and size. Notice that the earth object is inserted in the same orientation as it was in its own workspace, even though you are in the Front view.



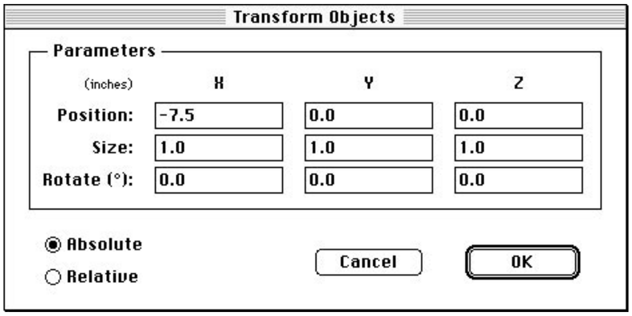
More accurately, you are inserting an instance of the "Earth-rotating" shape into a new shape. The instance carries all of the animation scripting entered in the original shape to this level. No event markers for that level's animation scripting appear here, but if you click on the Preview button, you will see the rotation action working on the instance.

3. **Insert the moon.** Do this by an alternate method. Switch the Textures palette to the Shapes palette, then select and insert the "Moon-rotating" shape using the controls and buttons on the Shapes palette.



This has the benefit of automatically inserting the shape at the view set center (which should be at 0,0,0 by default) and at the same size.

4. **Move the moon 7.5 inches away from the earth.** As discussed in the introduction to this project, the moon's orbiting diameter is 15 inches.



Transform Objects

Parameters

(inches)	X	Y	Z
Position:	-7.5	0.0	0.0
Size:	1.0	1.0	1.0
Rotate (°):	0.0	0.0	0.0

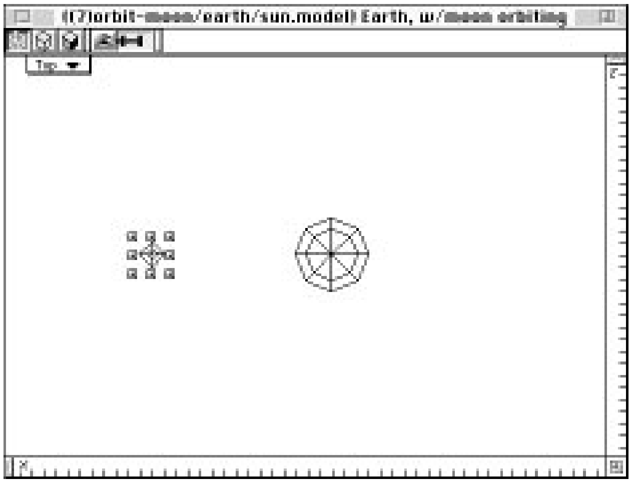
◉ Absolute
○ Relative

Cancel OK

The easiest way to do this is to use the Transform dialog. While the moon is still selected (as it sits inside the earth, since both were inserted at 0,0,0), choose the Transform command and enter -7.5 in the X axis field. Clicking on the OK button shifts the moon 7.5 inches to the left on the X or horizontal axis, as observed from the Front view.

5. **Switch to the Top view and zoom out until you can see both objects.** Obviously, the orbit will be on the X-axis' plane, so the Top view will give the best vantage point from which to script the orbiting action.

Using the Magnify tool, hold down the Option key while clicking in the active view. Each click zooms the view's scale out by 2X factor. You will need to click twice to see the moon in the view. This is done so you will have enough room in the view to rotate the moon around earth as you lay down the event markers.

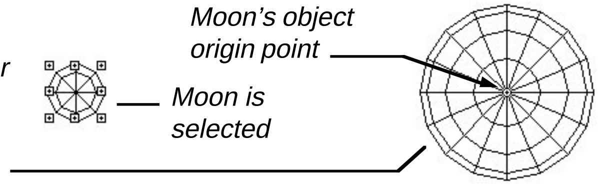


6. **Drag the moon's origin point to the center of the earth.** To do this, hold down the Control key, then click on the moon's origin point and drag it to the earth.

If you also hold down the Shift key, the move will be constrained to move only horizontally. Thus, it will be much easier to position the moons origin point EXACTLY in the center of the earth.

Asmall marquee appears representing the dot of the origin point in response to thedrag operation. When the screen redraws, notice that the moon's object origin point is positioned at the earth's center.

Dragging the moon's origin point to the center of the earth offsets it so you can easily script an orbit around the earth



The important thig is that the moon's object origin point is positioned at 0,0,0 while leaving the object stationary. While there are other methods to accomplish this task (see "origin points" in chapter 3), dragging with the Object Move tool while holdign down both the Control and Shift keys is the easiest, most direct way to accomplish this step.

7. **Move the time pointer to .5 seconds.** As discussed, the moon's orbit has the same period as its rotation. Thus, you will lay down three event markers that describe an orbit duration of 1.5 seconds.

8. **Rotate the moon around the earth in 120° increments.** Using the Object Rotate tool, with the moon object selected, click and drag a corner handle downward so that the orbit is counter-clockwise, just like the moon's rotation. As before, releasing the mouse button at 120° rotation causes the screen to redraw and a new event marker to appear on the Animation palette.

   Next, advance the time pointer to one second and rotate the moon another 120°. Then, option-drag the moon's first event marker to the 1.5 second mark. And finally, click on the last event marker (to select it), then enable cycling back to time zero for the object. Notice that the time pointer did not have to be moved to the 1.5 second position in order to use the Cycle command.

   Click on the Preview button to check the orbiting action. Notice how the moon also rotates as it orbits the earth, which is also rotating as prescribed. Notice also how the motion continues past the event markers you have laid down on the time line.

This concludes the third phase of the project. When you are satisfied that the animation scripting is correct in this shape, close the window. Remember to save again.

The earth's orbit

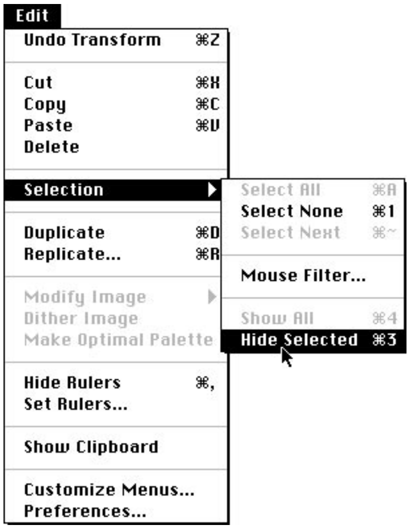
The last phase will script the animation of the earth's orbit around the sun. Finally you will begin putting objects in the main model space! You will draw a large sphere for the sun, then insert the "Earth, w/moon rotating" shape and give it an orbital action.

Yes, it is typical for a lot of work to be done in shape workspaces before beginning the assembly process in the main model space…at least for projects that are well-structured. Vision3d is very flexible; it allows you to build models any way you want. This project is an example of the recommended way to build models…the way that takes advantage of the program's modeling strengths.

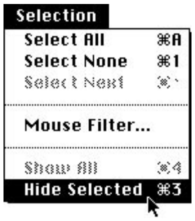
To script the earth's orbit around the sun:

1. **Begin this phase by switching the model window's view to the Top view.** You are preparing to draw the sphere for the sun in an orientation consistent with the other objects.
2. **Draw a sphere in the Top view.** This is the sun! Notice how the sphere's poles are oriented along the Y axis, top and bottom. If the sphere had been drawn in the Front view, its poles would have been oriented on the Z axis, front and back. In this project it really doesn't matter that much, but in your own projects this kind of attention to detail will be rewarded.

3. **Position the sun at 0,0,0 and make it ten inches in diameter.** As determined at the beginning of this project, this sphere representing the sun needs to be this size.
4. **Temporarily hide the sun.** Do this by selecting the sun sphere, then choose the Hide Selected command in the Selection menu (or submenu under the Edit menu on smaller monitors).



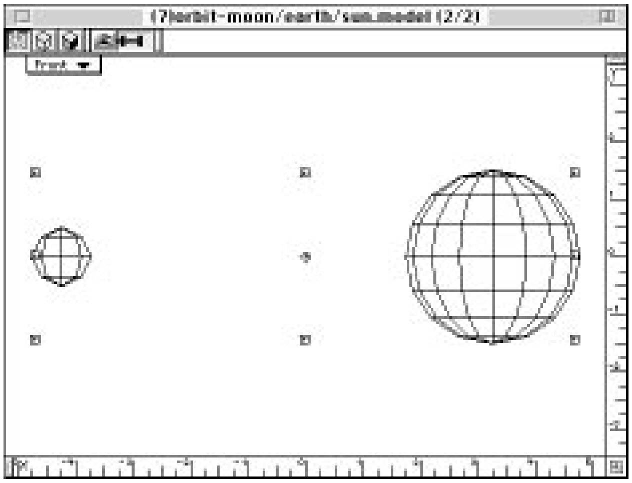
OR



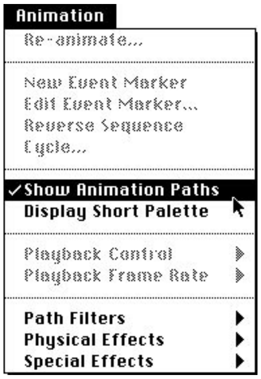
The reason for this step is to allow easier access to the inserted earth/moon shape instance. It will be inserted into the same location in the model occupied by the sun. If the sun is there also, it will be more difficult to selectthe instance and set up the earth's offset accurately.

5. **Insert the "Earth, w/moon rotating" shape.** Switch to the Front view first, if not already there. This is to maintain the appropriate orientation of the inserted shape. If it was inserted into the Top view, the entire instance would be rotated 90° from that of the shape workspace.

   Use the Shapes palette again to do this so this instance will insert in the model at actual size. Notice that the two objects — moon and earth — are inserted as a single, grouped instance. You cannot access the individual objects within the group at this level, unless you ungroup the instance and break its link with the shape. Notice also that there is one origin point for this instance, and that it is in the geometric center of the group. This is an important distinction.



6. **Center the instance's origin point on the earth.** Again, this is done with the Object Move tool as you did for scripting the moon's orbit. To move any object's or instance's origin point, you must observing the specific protocols that allow moving origin points without moving the object itself. This will be reviewed again here to help you remember it better.

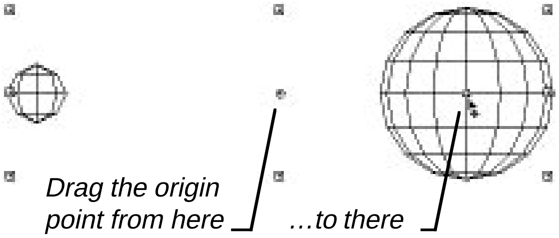


First, check to make sure the visible animation paths feature is turned off. This is done by looking at the Show Animation Paths command in the Animation menu to make sure no checkmark is present at the front of the command. Each time the command is chosen, it toggles on or off.

The object origin is the point on an object that follows its animation "path." If that point is moved, the position of the object shifts relative to its path. This can be useful, but the point cannot move without a conscious decision to do so. Thus, this command must be off to allow you to move the object origin point on objects.

Moving the object origin point

Hold down the Control key while dragging the origin point. Point moves while keeping the instance stationary

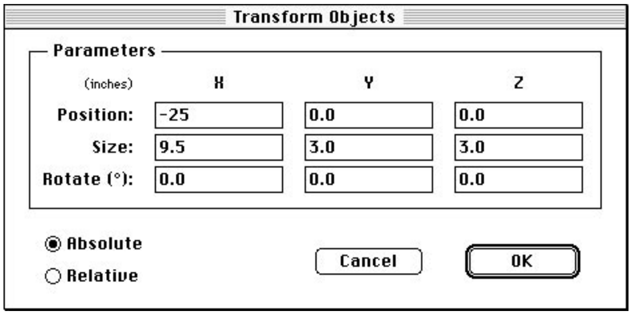


Drag the origin point from here

…to there

Second, hold down the Control key while you click and drag the round dot in the middle of the instance so it is in the middle of the large sphere, which is the earth in this project. Without the Control key, the entire instance would move as you drag its origin point. You can also hold down the Shift key to constrain the move horizontally.

7. **Move the "Earth, w/moon rotating" instance outward to its orbital distance.** The most practical way to do this is through the Transform command. With the mode set to absolute in the Transform dialog, set the instance's position to -25 on the X axis, as shown in the following illustration. This was the earth's orbital radius decided at the beginning of the project.



Transform Objects

Parameters

(inches)	X	Y	Z
Position:	-25	0.0	0.0
Size:	9.5	3.0	3.0
Rotate (°):	0.0	0.0	0.0

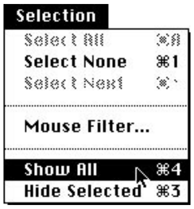
● Absolute

○ Relative

The position of a selection is determined by its object origin point. By moving the object origin point to the center of the earth before repositioning the earth/moon instance, the earth is positioned at 25 inches away from the center of the model. If the origin point was at the center of the instance's geometry instead, the earth would not have been positioned accurately.

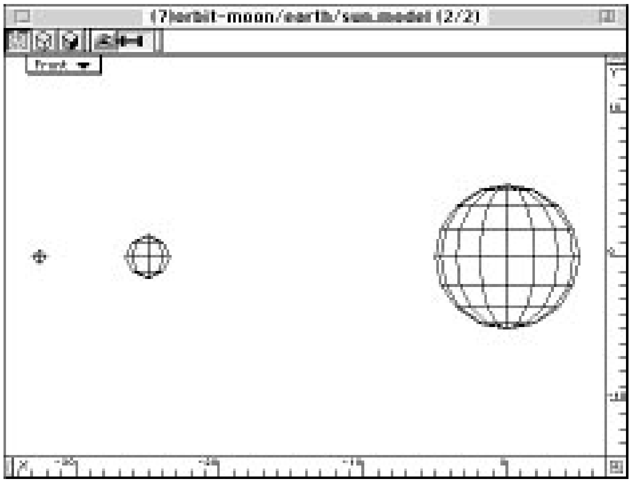
Clicking the OK button in the Transform dialog moves the earth/-moon instance out of the active view. The object is still in the model, just not visible at the view plane's current position in 3-D space and in the view's current scale. That will be corrected shortly.

8. **Show the sun again.** Do this by choosing the Show All command in the Selection menu. Alternately, you can use the keyboard equivalent, Command-4 (⌘4).



The view redraws to show the sun sphere…right where it was all the time, but hidden to keep it out of the way.

9. **Change the view position and scale to see the earth/moon again, along with the sun.** This can be done manually using the View Move and Magnify tools. However, the most practical way is to choose the "Fit Views to All" command in the Windows menu.



The view redraws to show all the objects at a reduced scale in the view. Of course, from your reading in chapter 3, you understand the mechanics of what has happened with the view, and that the objects have not changed scale. Notice the rulers also correspond to the reduced scale, but still accurately indicate position of the various objects as you have specified.

Notice also that the sun is shifted to the right edge of the view, which will make it a little difficult to rotate the earth completely around the sun during animation scripting. You could use the Magnify tool to zoom out, as before. But this alternate technique will easily deal with the situation when the time comes.

10. **Switch to the Top view again.** This is in preparation for scripting the earth's orbit around the sun. As you have seen before, in this project the Top view gives the best vantage point for working with the objects.

11. **Move the earth/moon origin point to the sun.** This time, rather than use the Link tool, you can simply Control-drag it in the Top view to the center of the sun. Remember, the Shift key constrains the move so you can be sure it is moved horizontally directly to the center of the sun sphere.

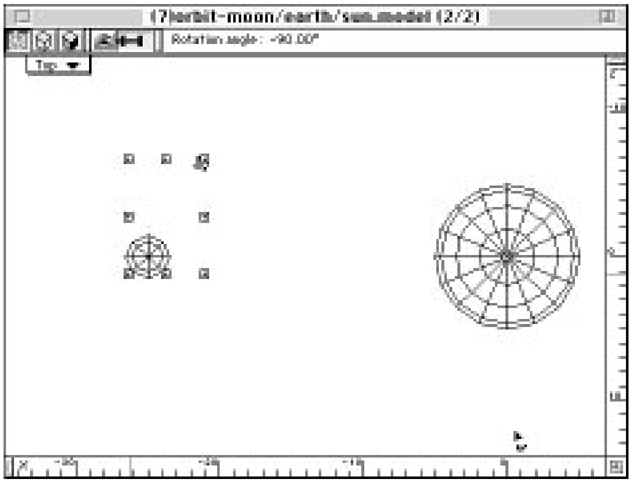
    Now you are ready to begin laying down event markers for the earth's orbit.

12. **Advance the time pointer to 2.25 second mark on the time line.** Of course, you could script event markers at three seconds, six seconds, and nine seconds, just as before. Whereas in previous scripting you have used 120° rotations, this will be slightly different by scripting four 90° rotations. There is no difference in the behavior, this is just a different way to accomplish the same task.

    Thus, the earth's orbit has a duration of nine seconds; one-quarter of that is 2.25 seconds. Notice that when the time pointer is moved to that point, the moon moves to the position for which it has been scripted in the shape workspace.

13. **Rotate the earth/moon instance 90° counter-clockwise.** With the earth/moon instance selected, and the Object Rotate tool selected, click and drag the bottom corner handle downward. The instance's marquee will move out of the view as you drag. Your cursor can even move outside of the window frame, but as long as the mouse button is held down you are still controlling the rotation of the instance in the model space.

    Watch the numerical feedback at the top of the window. When it reaches 90°, release the mouse button. The view redraws, but the earth/moon instance is NOT visible. Of course, it's still in the model, just outside the view. An event marker appears on the Animation palette at 2.5 seconds.



(7)orbit-moon/earth/sun.model (2/2)

Rotation angle : -90.00°

14. **Fit the view to the objects again.** Use the Fit Views to All command again to re-position the view set and show all the objects. This is in preparation for the next rotation.
15. **Advance the time pointer to 4.5 seconds and rotate the instance another 90°.** You now have a cumulative rotation of 180°. Use the Fit Views to All commands again. By now this should be old stuff.
16. **Advance the time pointer to 6.75 seconds and rotate another 90°.** You're almost there.
17. **Option-drag the first event marker at time zero to the nine second position.** The Animation palette will auto-scroll as you drag the event marker. Of course, this is the same technique used on all the 360° rotations you've created.
18. **Zoom out to see the entire "Solar System."** use the Magnify tool, with the Option key held down, to reduced the view's scale so you can see the whole orbit.
19. **Click on the Preview button.** Ensure that the earth orbits the sun in a smooth motion from beginning to end. Notice how the bounding box marquee for the earth/moon instance grows and shrinks, then shifts sideways. That's the moon orbiting the earth as it follows the earth around the sun. It may be hard to tell if the earth is moving smoothly around the sun with the bounding box continuously changing proportions, but if you've followed the steps correctly, everything should be right.

This ends the modeling phases of this project. Be sure to save the model again before continuing.

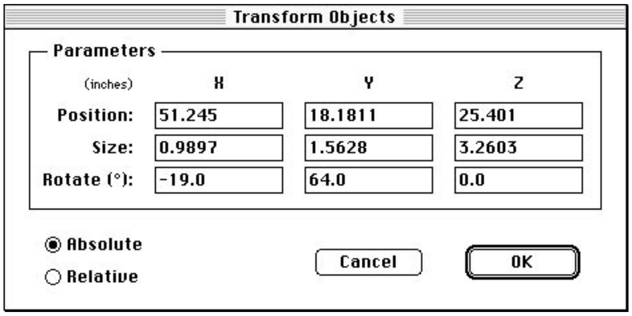
Rendering the animation

Obviously, the payoff for all your work in this project is to see it in a rendered animation…in motion. No textures have been applied to these objects, so it may be a little dull. But you can always fix that later. Note that if no textures are currently applied to an object anywhere on the time line, when a texture is applied it propagates through all the event markers uniformly. Once a texture is applied, changing the texture sets up a morphing action between event markers. THIS IS AN IMPORTANT DISTINCTION.

The trick for an effective animation is picking the best viewing position. In this project, the camera will not move. Unlike the real world (or should it be Universe), in effect you can go hang stationary in space and watch the world(s) go by. And Einstein said it couldn't be done! Afterward, you may want to script some motion to the camera — a flyby of the moon perhaps?

To set up the viewing position:

1. **Set the time pointer at time zero.** This is always good practice. It's important to always be aware of the time pointer's position on the time line before you start inserting objects into a model.
2. **Select the Camera Object tool on the Tool palette.** The most efficient method for setting up a viewing position is to use a camera object.
3. **In the Front view, click once.** This inserts a camera object just in front of the active view, wherever that plane happens to be in 3-D space.
4. **Choose the Transform command.** Let's make this part real easy. Just enter the numbers shown in the following illustration to set the position and orientation of the camera in 3-D space. Be assured this position was arrived at after much soul searching and experimentation, i.e. — "Yeah, that looks about right!"

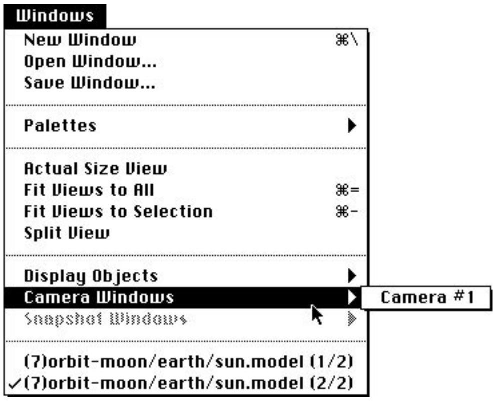


(inches)	X	Y	Z
Position:	51.245	18.1811	25.401
Size:	0.9897	1.5628	3.2603
Rotate (°):	-19.0	64.0	0.0

● Absolute
○ Relative

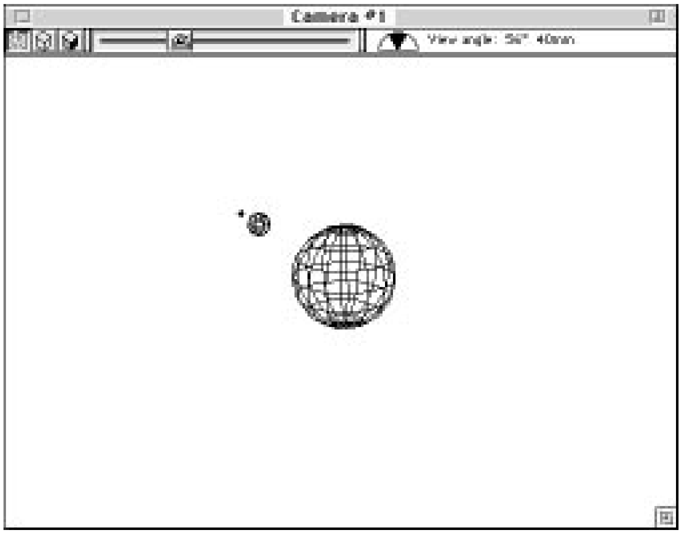
As seen from the Front view, these numbers position the camera object to the left and slightly in front of the sun, but way above the orbital plane. Seriously, it does give the best perspective view of the earth's full orbit.

5. **Open the camera object's window.** This can be done by double-clicking on the camera object or by choosing the camera by name from the Camera Windows sub-menu.



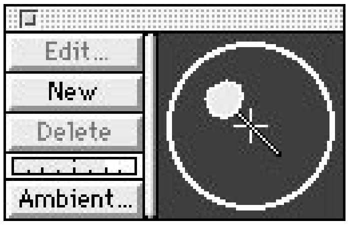
Either way opens a camera window showing the model as if looking through the viewfinder of the camera object. The name of the camera object defaults to Camera #1, but can be changed in the Edit Primitive command's dialog box.

6. **Re-size the camera window and adjust its focal length.** The camera window defaults to a fixed size, which is not wide enough for this project. Set the size to 500 X 350. Set the focal length to 56°/40mm.



With the time pointer at time zero, the above illustration shows what you should see in the camera window. The sun is closest to the camera, with the earth and moon behind it. As the earth orbits the sun, it will come toward the camera, passing slightly below this viewing position, then around the back side of the sun to its starting position. …All this in nine seconds.

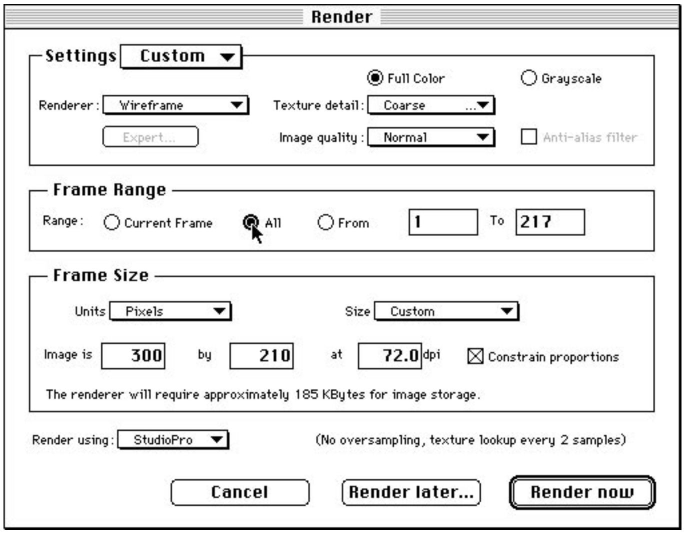
7. **Set the lighting position.** With the camera window the active view, note that the directional/global light source on the Lighting palette is positioned on the back hemisphere. Hold down the Control key as you drag the light into a new position on the front hemisphere. It should be positioned as shown in the following illustration. Of course, in wireframe renderings, the lighting doesn't matter, but it would be important at higher levels of quality.



The best arrangement would be to place a point light source in the center of the sphere, so that your sun would appear to be illuminating the earth and moon. You would then need to apply a texture to the sun that turned off the sphere's shadow casting (via a checkbox in the Expert level of the texture editing dialogs) so that light source's illumination could get through the surface of the sphere, as explained in chapter 6. You may want to do that as you enhance this project with textures. You would also want to delete the default directional light source or at least set it to a low intensity, so it didn't interfere with the sun as the primary illumination for this model.

8. **Render the animation.** It would be best to start with a wireframe version first. Wireframe renders quickly (remember, at 24 frames per second there are 216 frames to render) and it will give you a good sense of the motion (which is the purpose of this exercise). If you are happy with the wireframe animation, move up to the Gouraud rendering method, which gives the fastest smooth shading renderings. If you further apply surface textures with maps representing the earth's and moon's geographies, you will need to use the Phong renderer, which is the real workhorse in animation projects.

   To render the animation, select the Camera tool on the Tool palette. That's the big button at the bottom of the palette. (Do not confuse the Camera tool with the Camera Object tool, they perform radically different tasks.) Hold down the Shift key while clicking once in the camera window.



Render

Settings: Custom

● Full Color ○ Grayscale

Renderer: Wireframe

Texture detail: Coarse

Image quality: Normal

□ Anti-alias filter

Frame Range

Range: ○ Current Frame ● All ○ From 1 To 217

Frame Size

Units Pixels

Size Custom

Image is 300 by 210 at 72.0 dpi

☒ Constrain proportions

The renderer will require approximately 185 KBytes for image storage.

Render using: StudioPro

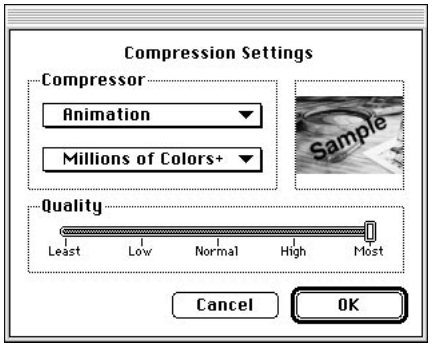
(No oversampling, texture lookup every 2 samples)

Clicking the camera tool takes a “snapshot.” More technically, it initiates a rendering. Holding down the Shift key while clicking opens the Render dialog, so you can set the rendering parameters before actually beginning the rendering. Without the Shift key, the rendering begins immediately. Note that shift-clicking with the camera tool is exactly the same as choosing the Render command in the Rendering menu. You may also click and drag a cropping marquee while holding down the Shift key.

In the Settings section of the Render dialog, set the Renderer field to “Wireframe” and the Image quality to “Normal.” In the Frame Range section, click on the “All” radio button. In the Frame Size section, set the first field of the Image size to “300.” With the “Constrain proportions” checkbox enabled, the second dimension field automatically adjusts to maintain the aspect ratio of the image frame. If the camera window was set to 500 by 350, the second value in the render dialog should be 210. Reducing the frame size increases the rendering speed, takes less disk space, and allows the animation to play faster when complete.

Click on the “Render Later” button at the bottom of the dialog. You may render immediately, if you wish. However, using the Render Later option lets you save a suspended rendering that you may open and begin rendering at your convenience. Suspended renderings are self-contained, so they do not require the original model to be open during rendering.

The Save As dialog appears for you to designate the name and location where you wish to save the suspended rendering file. This dialog also allows you to choose the file format for saving the animation. The two choices are PICS and QuickTime Movie. Movie format is available ONLY if the QuickTime 1.6 or later extension is installed in your machine's System Folder. (See the discussion on QuickTime support later in this chapter.)



Compression Settings

Compressor: Animation; Millions of Colors+

Quality: Most

Choose the Movie format and click on the OK button. A second dialog appears allowing you to specify the movie's compression parameters. For this project, just click on the OK button to accept the recommended default settings.

sun/earth/moon-orbits.suspended

The suspended rendering file is saved. To actually render the animation, just double-click on the file icon on the desktop, or use the Open command in the File menu. The various options and details of rendering are discussed in *Chapter 8: Rendering*.

When the rendering of your animation is complete, you can view it in Vision3d with an animation window. Animation windows have playback controls that work much like a home VCR. Please refer to *Chapter 1: General Operation* for a discussion of the different types of windows used in Vision3d — the Animation window is explained there in detail.

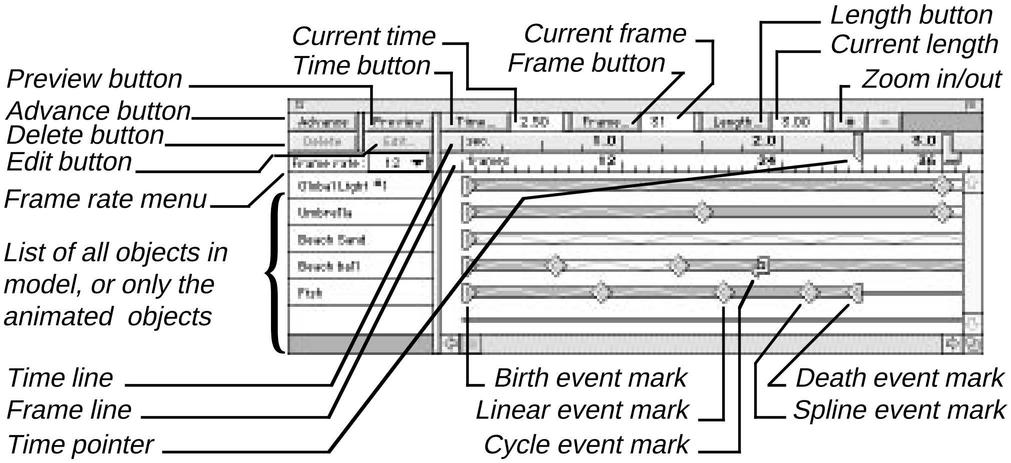
This example has taken you step-by-step through the process of efficiently creating complex motion — by scripting simple motions at each level, building up the actions to accomplish the desired motion. The advantage should be obvious: if you want to adjust part of an object's motion (speed up or slow down the orbit of the moon, make its orbit oval instead of round, etc.), you can just go to the shape level where that specific action is scripted to make the change.

The project itself was simple, involving only three spheres, but the concepts presented are what's important. Now you must extrapolate and adapt them to your own projects.

The Animation Palette

The Animation palette places all controls needed to create an animation in one convenient location. The palette may be switched between scene-based animation and object-based animation, i.e. expanded to show additional data for selected objects.

The Animation palette is designed to float in front of all document windows, but may be covered by other palettes. If another palette overlaps it, the Animation palette may be brought forward so it is fully visible by clicking on it.



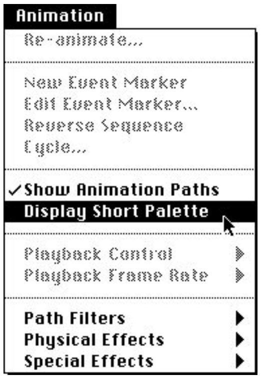
The Animation palette contains the standard window controls found on all desktop windows, including a move bar, close box, zoom box, resize box, and scroll bars. These controls should be familiar to you.

The animation palette may be sized and positioned on the desktop as desired. Its re-size box works both horizontally and vertically.

The specific Animation controls contained on this palette are as follows:

- **Advance button.** Use to increment the time pointer to the next whole second. If the length pointer is next to the time pointer, it is automatically advanced, too.
- **Preview button.** Use to preview the scripted animation in the model or camera window. This preview is not a rendering, but gives motion to the objects' bounding boxes. It is intended to be only the coarsest kind of preview to help detect gross errors in the motion.

  In complex models, the speed at which the preview progresses can be improved by selecting only those objects that you wish to see in motion. Only the selected objects bounding boxes become animated. If there is no selection, then all objects bounding boxes are put in motion, which may slow the running speed.
- **Delete button.** Use to remove an event marker from the Animation palette, and from the affected object's database. This operation is undo-able.
- **Edit button.** Use to open the Event Options dialog for a selected event marker. The dialog allows you to edit parameters for handling the object as its motion path passes through that "event."
- **Frame rate pull-down menu.** Use to set the frames per second for the document. A new document defaults to 12 frames per second. 30 frames per second is the rate at which video tape is recorded.
- **Object list.** Use to select objects in the model. Clicking on an object's name selects that object in the model, and vice versa of course. Multiple objects can be selected by holding down the Shift key as you click on the names.



You can choose whether to have all the objects in the model listed on the Animation palette, or only those for which an animation path has been scripted. This is set by choosing the Display Short List command in the Animation menu. This is helpful with models containing lots of objects, only some of which have motion.

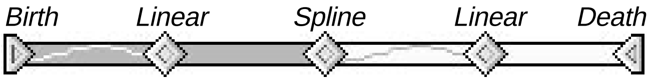
- **Time Pointer.** Use to set the current time. This is done by dragging the pointer with the cursor to the desired position on the time line. As the pointer is dragged, the objects in the model with animation paths will move to reflect their action at that moment in time. Releasing the mouse button causes the model to redraw as it is defined by the animation paths scripted for objects in the model.
- **Time Button,** Use to set the current time numerically. Clicking on this button displays a dialog into which you enter the time to which you wish to move the time pointer. Clicking on the OK button moves the time pointer. This can be useful when the time pointer cannot be set accurately enough by dragging it with the mouse on the palette. A field next to the time button indicates the current time at which the time pointer is set.
- **Frame Button.** Use to set the current frame numerically. There may be times when you want to set the time pointer at a specific frame rather than at a specific time. Clicking on this button displays a dialog into which you enter the frame to which you wish to move the time pointer. Clicking on the OK button moves the time pointer. This can be useful when the time pointer cannot be set accurately enough by simply dragging it on the palette. A field next to the frame button indicates the current frame at which the time pointer is set.
- **Length Pointer.** Use to set the length of the animation. This is done by dragging the length pointer with the cursor to the desired position on the time line. The time pointer cannot be moved past this position on the time line, although event markers can. Further, it is the length displayed in the Render dialog. No animation frames can be rendered after this point on the time line, even if event markers are present.
- **Length Button.** Use to set the animation length numerically. Clicking on this button displays a dialog into which you enter the time to which you wish to move the length pointer. Clicking on the OK button moves the length pointer. A field next to the length button indicates the current time at which the length pointer is set.
- **Zoom in/Zoom out Buttons.** Use to expand or contract the display of the time line horizontally on the palette. Each click on the plus (+) or minus (-) buttons zooms in or out 2X. The maximum expansion is six times normal, or three clicks on the plus button. The zoom out (minus button) will not contract further than the normal scale. This

feature is useful in situations where you need to position the time pointer and/or event markers more accurately than is possible in normal scale.

- **Event markers.** Use to script the animation paths of object or light sources in a model. Event markers are created as objects are positioned, rotated, scaled, or given textures. Event markers are created for light sources as they are positioned, rotated, given gels or textures, and adjusted for illumination intensity.

Four main types of event markers

…and variations on their path control



Dk. green: change	*Dk. green: change*	*Lt. green: no change*	*Lt. green: no change*
Wavy line: spline path	*No line: linear path*	*Wavy line: spline path*	*No line: linear path*

The type of event marker present at a specific point on the time line is indicated by a wavy line between it and the next marker for spline; the absence of a line for linear. Spline and linear markers share the same icon (thus, the wavy line). Birth and death markers each have unique icons that identify their purpose.

Finally, Cycle-type event markers appear when cycling is enabled for a specified segment of an animation path. This is done through the Cycle command in the Animation palette. The event marker itself may spline, linear, birth, or death, and behaves as it would without cycling, but the icon for that marker at which cycling occurs is changed to a folded arrow pointing back to the left.

Selecting and moving event markers on the palette

Event markers are selected by clicking on them. The inset of a selected event marker changes to a red color to indicate its selection status. Multiple event markers may be selected by holding down the Shift key while clicking of each marker icon, or by dragging a marquee around the event markers you wish to select. Multiple selections do not need to be contiguous and can include event markers from more than one object.

Selecting an event marker DOES NOT automatically move the time pointer to that position on the time line. If you hold down the Command key while clicking on an event marker, the time pointer will move to that position; causing Vision3d to redraw all modeling views to show the state of the model at that point in time.

Event markers may be moved along the time line to alter the object's pacing and sequence in the animation. Event markers cannot be dragged from one object line to another.

An event marker is deleted by selecting it, then clicking on the Delete button on the palette. Alternately, you can use the Delete command in the Edit menu or push the Delete key.

Modifier keys that apply to event markers:

A. **Shift key —** Extends the selection of event markers on the palette. Multiple event markers may be selected at one time for copying, deleting or moving in unison.

B. **Option key —** Leaves original event marker in the original location and creates a copy of that event marker as you drag it to a new location.

C. **Command key —** Moves the time pointer to the position of an event marker when you click to select it.

D. **Control key —** Spaces out or compresses the distance between all event markers as you drag the selected marker This is useful when you want to open some room for one or more additional event markers or close the gap after removing some.

Quick Tutorials

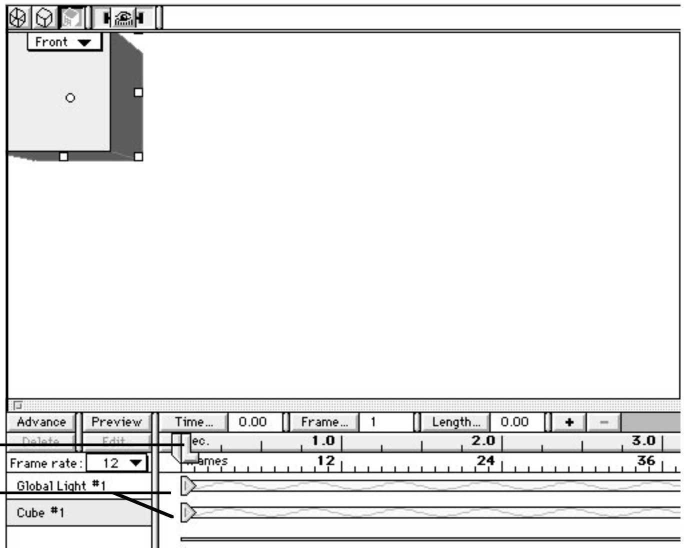
The purpose of the tutorial exercises in this chapter is to familiarize you with the concepts of the event-based animation mechanism in Vision3d.

Exercise 1: simple motion

In Exercise 1, you will learn how to lay down event markers on the Animation palette to animate a simple object. This is fundamental to all animation scripting operations in Vision3d.

1. **Open a new model.** Set the display method to Shaded Surfaces and the three-position lens control to Normal. Make sure you're in the Front view.
2. **Draw a cube in the upper left corner of the model window.** The size is not important, just make it a comfortable size for the view, as shown in the illustrations of this exercise.
3. **Display the Animation palette, if it's not already visible on the desktop.** Pull down the Windows menu and choose Show Animation Palette from the Palettes sub-menu. Vision3d shows the floating Animation palette where is was last positioned, usually at the bottom of the screen. Notice on the Animation palette there is a list

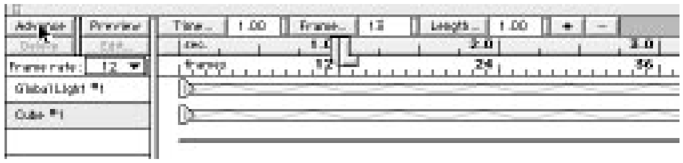
of the elements in the model. Currently, Global Light #1 and Cube #1 are listed. Notice each element has its own timeline extending to the right the element name. Notice also that the animation time indicator is currently at time zero. The single event markers on each timeline indicate that both elements in the model — the light source and the cube — exist in their current positions at time zero of the animation.



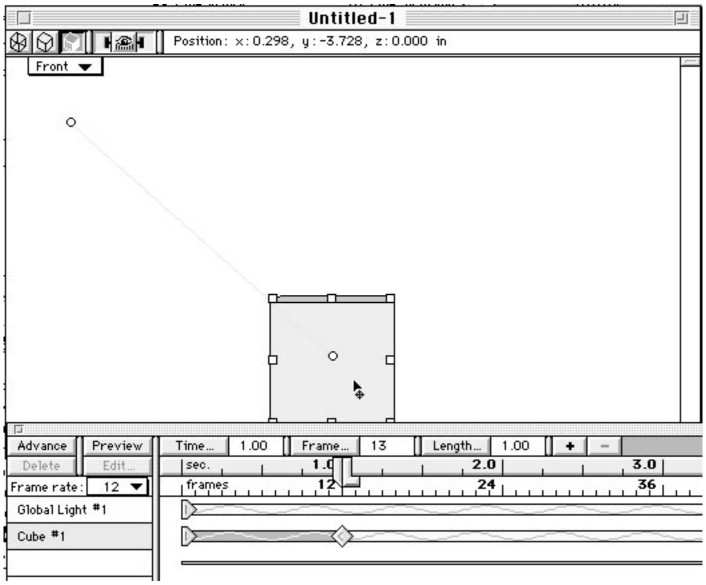
Time Indicator

Event Markers

4. **Click on the Advance button.** Vision3d responds by moving the time indicator to the one-second position on the time line.



5. **Drag the cube to the lower center of the Front view.** Be sure to choose the Pointer tool to do this, if it's not already chosen. Notice that when the cube is repositioned, Vision3d creates a new event marker at the one-second time on the Cube #1 timeline. The new marker indicates that at one second, the cube is in the position you have specified. Notice that the Global Light #1 timeline is unchanged. This is because the position of the light source does not change between time zero and one second in your model.

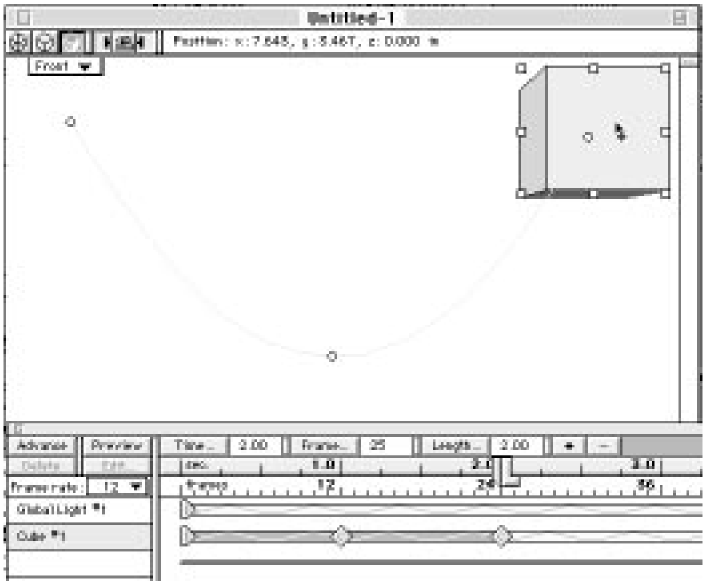


Position: x: 0.298, y: -3.728, z: 0.000 in

Time... 1.00 Frame... 13 Length... 1.00

Frame rate: 12

6. **Click the Advance button again and drag the cube to the upper right corner of the Front view.** This moves the time pointer to the two-second position. Again, Vision3d creates a new event marker at the two second position for Cube #1. This event marker will be the last one in your animation.

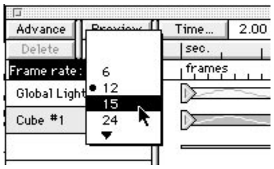


Position: x: 7.643, y: 3.467, z: 0.000 in

Time... 2.00 Frame... 25 Length... 2.00

Frame rate: 12

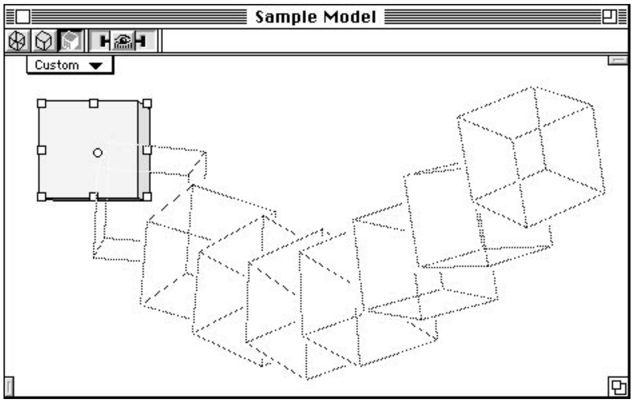
8. **Set the Frame Rate field to 15.** This indicates to Vision3d that you want to render 15 frames for each second of the animation. Using as time and position references the three event markers you have scripted, Vision3d automatically calculates the in-between positions for the cube, i.e. — those that come between the event markers.



The resulting animation will contain a total of 30 frames. Note that the starting frame at time zero is numbered frame #1 and the last frame is numbered frame #31.

9. **Click on the Preview button to see your animation in the model window.** Vision3d plays a simple “bounding box” run through of the animation.

As the time pointer is dragged or when the Preview button is clicked, animations play as bounding boxes in the modeling windows



From this very simple exercise, you’ve learned how to set event markers, edit the frame rate and preview an animation. The next step in the process is to actually render and save each frame of the animation sequence to your hard-disk drive so they can be replayed for viewing.

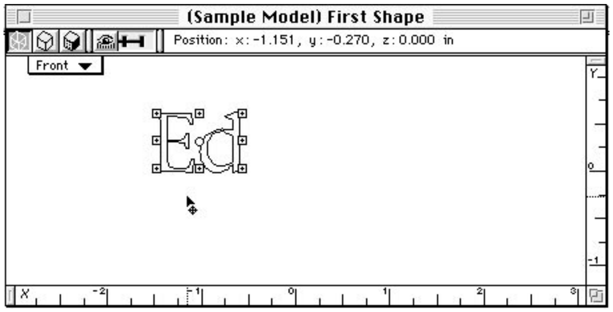
You’ll tackle that process of rendering an animation in the next exercise. But before you get to that point, the next exercise will have you try some more advanced animation techniques.

Exercise 2: a flying logo

In this exercise, you’ll learn do a more complex animation. The purpose is to acquaint you with Vision3d’s hierarchical animation structures, with camera objects, and with the concept of rendering a draft version of the animation to check for problems.

1. **Open a new model.** Set the three-position lens control on Orthographic and the view position at Front, if not already using those setting. Just in case, also choose the Actual Size command in the Windows menu. This resets the window scale to 1:1 and moves the view set center to 0,0,0 in the model space. It's a good place to begin.

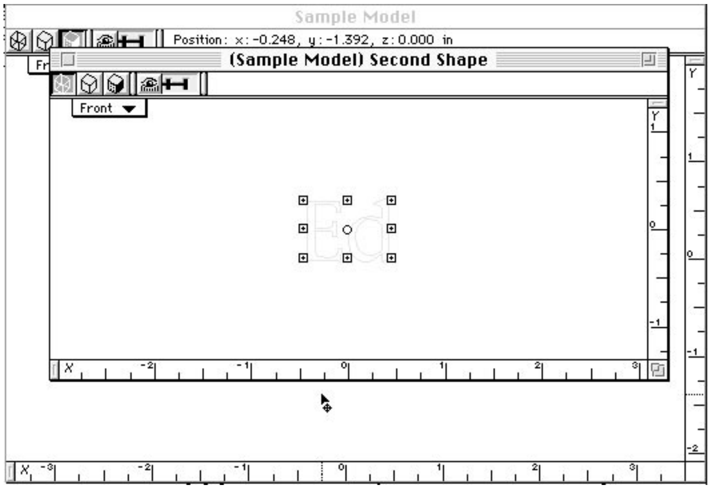
   Choose the display method you are most comfortable with. Wireframe is the fastest for redrawing the model views.
2. **Create a new shape.** Name it "First Shape."
3. **In the First Shape workspace window, create a 3-D object of your given name.** Use the text tool to do this. In the text dialog, choose a font and style, then type in your first name. Scale the 2-D letter-objects so they're about an inch high. Using the Extrude modeler, extrude to a depth of one-half inch.



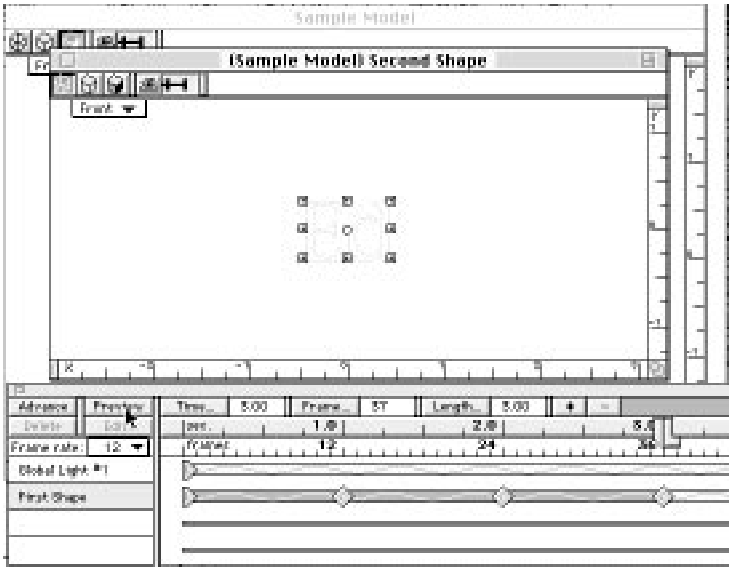
Position: x:-1.151, y:-0.270, z:0.000 in

4. **Apply the Blue Enamel texture to your 3-D name.** Still in the shape workspace, choose the texture and apply it to your 3-D name. If you want to see how the texture looks on the object, do a quick preview rendering now.
5. **Close the shape workspace window.** Click on the close box in the upper left corner of the workspace window. Vision3d returns you to the model window, which at this point in the exercise is still empty.
6. **Create another new shape.** Title this one "Second Shape."
7. **Insert the First Shape into the Second Shape.** While in the Second Shape workspace, pull down the Shapes menu and choose First Shape from the Insert Shape sub-menu. Notice when you move the cursor back into the shape workspace's window a "loaded" cursor is displayed, meaning the chosen shape is ready for insertion into the workspace. With the loaded cursor in the window, click once.

Vision3d inserts an instance of your 3-D name (with texture) into the Second Shape's workspace.

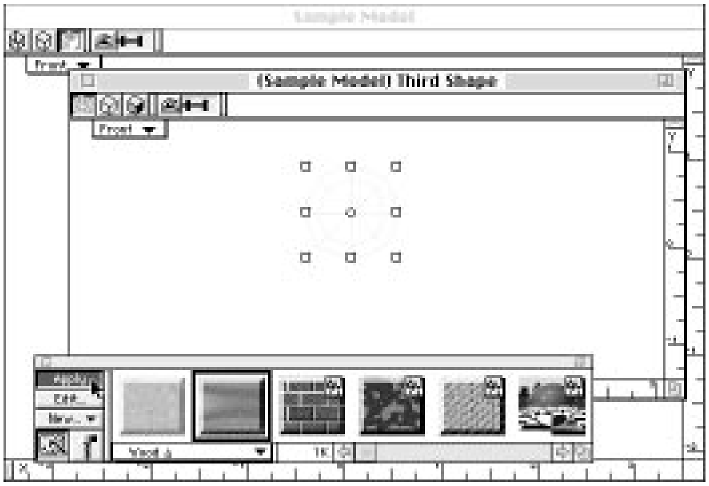

Sample Model
Position: x: -0.248, y: -1.392, z: 0.000 in
(Sample Model) Second Shape
Front
Y
X

8. **Still in the Second Shape workspace, script a three-second animation of your name rotating 360 degrees counter-clockwise.** Display the Animation palette, if it's not already showing on the desktop. Click on the Advance button to move the time indicator to the one-second point. Then select the Rotate tool from the tool palette. With your name selected, click on any corner selection handle and rotate your name counter-clockwise a minus 120 degrees.

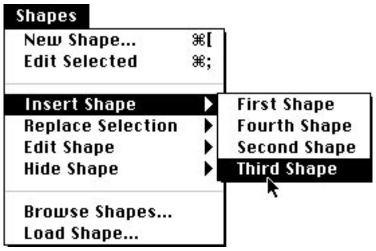


Then click on the Advance button again, and rotate your name a further -120 degrees counter-clockwise. Finally, click once more on the Advance button, and rotate your name another -120 degrees counter-clockwise. You should see three event markers on the First Shape timeline; one each at the one-, two- and three-second points. Check your animation scripting by clicking on the Preview button. The bounding box preview should make a 360-degree rotation.

9. **Close the Second Shape workspace.** This would be a good time to save the model. Use a name like, “Flying Logo - my name.”
10. **Create a new Shape.** Title this one “Third Shape.”
11. **Draw a Sphere in the Third Shape workspace.** Its size should be about 2 inches in diameter.
12. **Apply the procedural Wood texture to the sphere.** If you want to check this before moving on, do a quick preview rendering now.



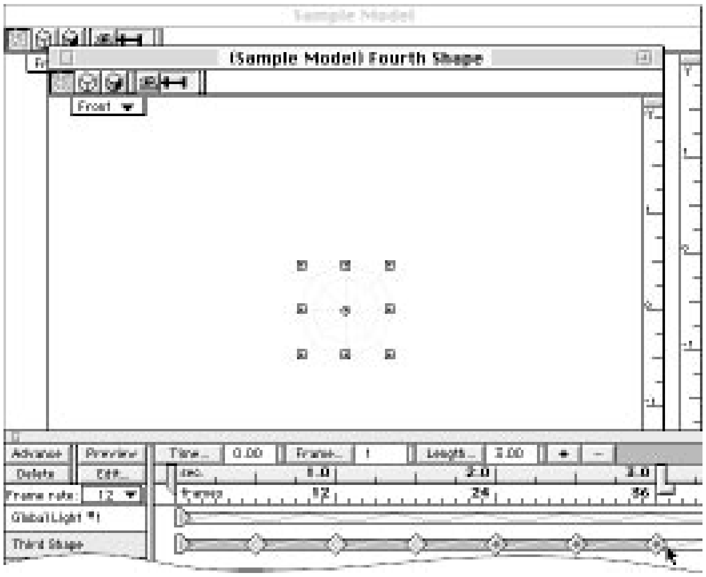
13. **Close the Third Shape workspace.** Save the model again.
14. **Create a new Shape.** Call this one…you guessed it, “Fourth Shape.”
15. **Insert the Third Shape into the Fourth Shape.** Do this just as you previously inserted the First Shape into the Second Shape.



16. **In the Fourth Shape workspace, script a 3-second animation of the sphere rotating a total 720 degrees clockwise.** This is very similar to what you did in step 8, except the object is a sphere and you will script it to make two full rotations in three seconds. Refer to step 8 if needed.

    Because the sphere will rotate twice as fast as the object in step 8, you must lay down an event marker every one-half second instead of every second. To do this, simply click on the current time indicator and drag it to one-half second. Then rotate the sphere 120 degrees clockwise. Then click-and-drag the current time indicator to one second. Repeat the rotation. Then do the same with the current time pointer at 1.5 seconds.

    Now, instead of creating three more event markers, copy the three you have already made, and use them again at different times. To do this, click-and-drag a marquee around the three event markers on the Third Shape timeline, so all three are selected. You know they are selected because each has a red center (for grayscale monitors, the center is dark gray; on monochrome, the center is black).

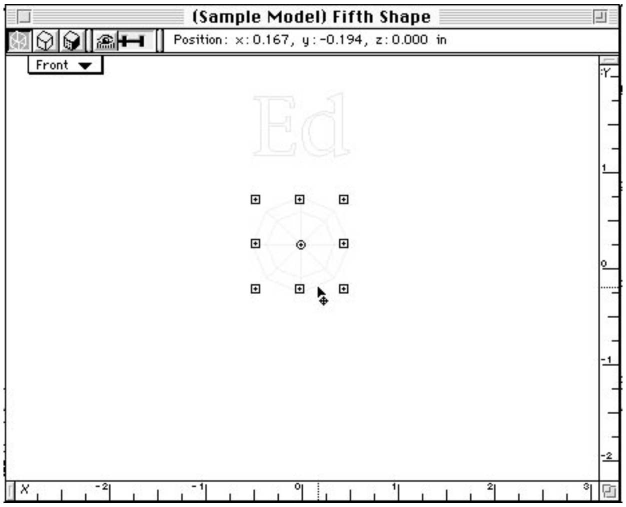


Time... 0.00
Frame... 1
Length... 3.00
Frame rate: 12

    Then, holding down the Option key, click-and-drag the three event markers to duplicate them on the timeline. Drag them to the right until the event marker on the far right side is aligned with the animation length indicator at three seconds. Then release the mouse button. The result gives you six event markers on the Third Shape's timeline, with marker #4 a duplicate of marker #1, #5 a duplicate of #2, and #3 a duplicate of #6.

To check your work, preview the animation in the shape window. The bounding box should rotate two full times over the course of the three-second animation.

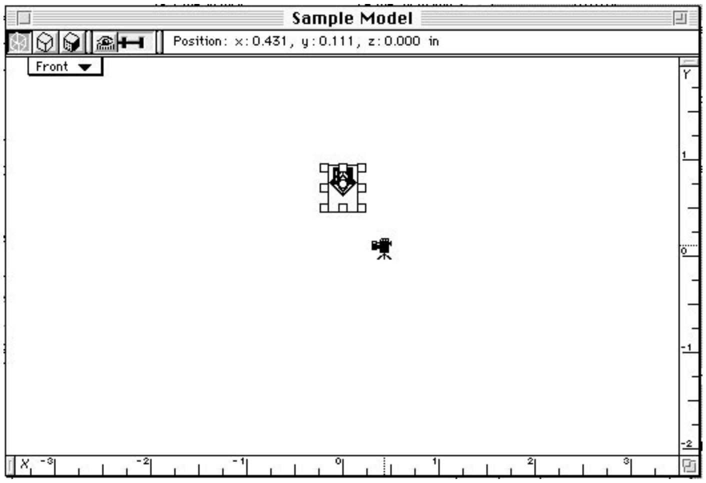
17. **Close the Fourth Shape workspace.** Time to save the model again.
18. **Create a fifth new Shape.** I think you know what to call it.
19. **Insert the Second Shape and the Fourth Shape into the Fifth Shape.** Refer to step 7 as needed.
20. **Assemble the instances together.** Position the second and fourth shapes as shown in the following illustration.



(Sample Model) Fifth Shape

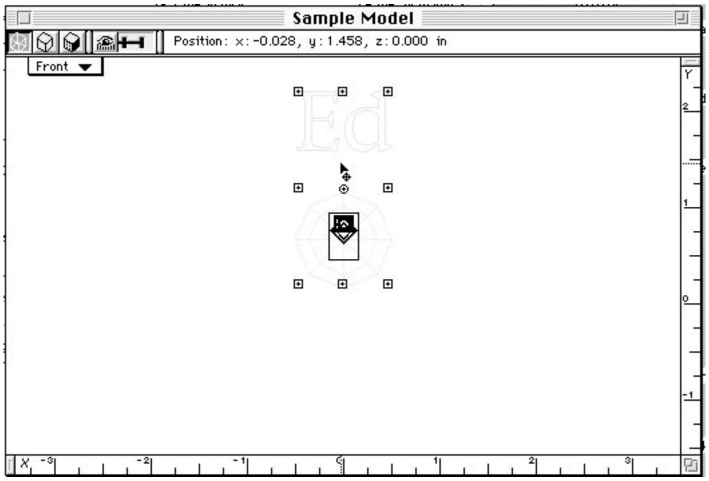
Position: x: 0.167, y: -0.194, z: 0.000 in

21. **Close the Fifth Shape workspace.** Save the model again.
22. **Insert a camera object into the model window.** Choose the Camera Object tool from the tool palette, and click once in the model window. When the view redraws, position the camera object as shown in the illustration on the following page.



Position: x: 0.431, y: 0.111, z: 0.000 in

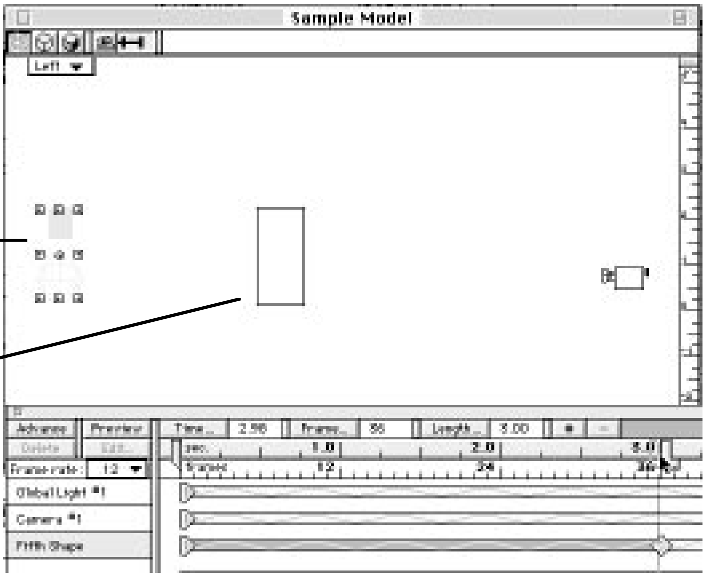
23. **Insert the fifth shape into the model window.** By now, this should be “old hat” for you. Do it just like you’ve done with other shapes.



Position: x: -0.028, y: 1.458, z: 0.000 in

24. **In the model window, switch to the left view.** Use the “Fit Views to All” command, if necessary, so you can see the camera object and the fifth shape instance in the Left view.

25. **Script an animation that moves the name and sphere objects toward the camera.** Be sure the instance is selected. (Since instances of both objects were inserted into the fifth shape, and the fifth shape in turn was inserted into the model window, clicking on either object will select them as a group.)



The Fifth Shape objects start here at timepoint zero.

As you preview the animation, the bounding box stops here at timepoint three seconds.

By now you know how to script an animation. You probably need only two event markers here — one at time zero for the beginning position, and another at three seconds to set the finishing position.

Since the first event marker at time zero was created automatically when you inserted the Fifth Shape into the model window, you only need lay down the finishing marker. Move the time pointer first (to the three-second time), then move the Fifth Shape's instance to its finishing position.

That's it! The event marker is created and the animation is ready to fly. Preview the animation in this view before going on. Also remember to save the model again.

26. **Double click on the camera object.** Vision3d opens the camera window, which displays a view of your model from the camera object's position. You should see something similar to the illustration on the following page.

27. **Render a draft of the animation.** It's generally a good idea to render a draft of any animation to check for problems before committing to a final, high-quality rendering.

    Be sure the camera window is the active window. Pull down the Rendering Menu and choose the Render command. In the Render dialog box, in the Settings section, choose "Movies•Draft" from the pull-down menu. This sets up a complete definition of parameters for your rendering. In the Frame Range box, click on the "All" radio button, if not already selected. This is to render all frames of the animation.

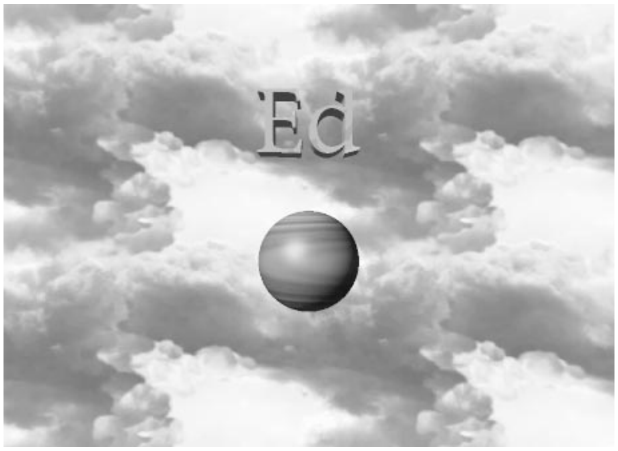
    Then click on the "Render Now" button. Vision3d opens a Save As dialog, allowing you to name the animation file and choose a file format and location in which to save it. If you have Apple's QuickTime system extension installed, save your animation in QuickTime Movie format.

    Clicking on the OK button presents a second dialog asking for movie compression settings. Go with Animation and 256 colors as the compression settings. If you don't have QuickTime installed, save the animation as a PICS file with 8 bits/pixel resolution, optimal palette and dithering enabled as your options.

    With either format, once you click on the OK button, Vision3d will begin to render each of the frames and automatically save each to the hard disk location you chose.

When the rendering finishes, play back the animation. Check it closely with a critical eye. Are the objects placed too close together, causing them to intersect as they rotate? Do they move too close to the camera as the animation finishes? If you see these or any other any problems, now is the time to correct them. But remember, this animation is hierarchical; specific animation events take place in specific levels. If you edit the animation, be sure to go to the proper workspaces in order to preserve the hierarchy.

If you want, you can render a final high-quality version of this animation. You may like to put a background behind the objects for the final rendering. If so, pick a background environment from the Environment Effects dialog window, which is reached through the Rendering menu.

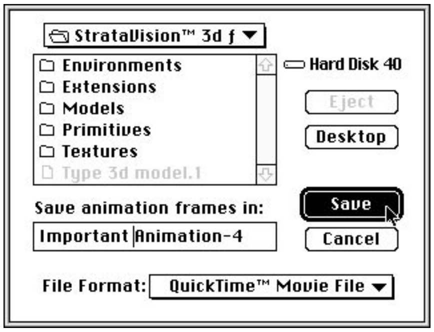


Render your final version using the Movies•Better rendering setting. Before initiating the higher quality rendering, go to the Rendering Effects dialog window and disable all the Object Effects EXCEPT Surface Mapping, Bump Mapping and Smooth Shading. Keep in mind that rendering a high-quality final version of a still shot or an animation always takes longer than rendering a preview or a draft version. But once you've seen a preview or draft and know that everything is ready, you're ready to commit to a final rendering. And the final results will be worth waiting a little longer for.

QuickTime support

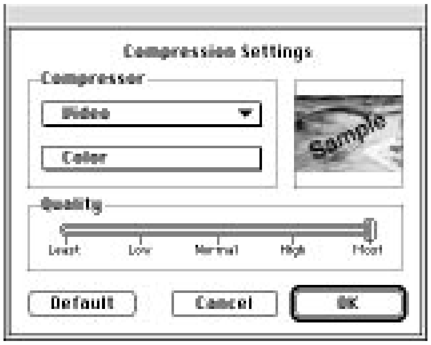
When saving an animation in QuickTime format, a second dialog appears in which you are given a choice of the compression algorithm to use and how much the file is compressed. A sample image is displayed in the QuickTime compression dialog that previews the effects the various settings will have on your animation frames. You have the option of using a QuickTime compression method appropriate to the format — either during the creation of the animation or when switching formats with the “Save A Copy As” command.

You should be aware that whenever QuickTime compression is used to save a file, the QuickTime extensions must also be present to open the file again or you will not have access to the compressed file.



Save animation frames in: Important Animation-4

File Format: QuickTime™ Movie File



Compressor: Video

Color

The compression settings are a key part of the rendering process when saving in QuickTime Movie format. They can greatly affect the results. And there is also an interplay between the various compressors and the amount of compression. Vision3d’s default settings use the Animation compressor, with Millions of Colors+, and Most quality. This is obviously biased toward quality over file size. However, it is important to have a good understanding of what the compression settings are, and what affect modifying each one of them has for the different compressors. Once this is achieved, you can make more intelligent choices of what compression settings to use, based on the intended use. Of course, you are encouraged to experiment with different settings, to get a feel for what the limitations and advantages are for each compressor.

An explanation of the different compressors and their recommended use is included later in this section of the chapter. To better illustrate the tradeoffs between compression settings and intended use, here are a few example scenarios:

1. Suppose you want to send a short (approx. ten seconds) preview animation to a client for approval over a 9600 baud modem. To do this, it may be acceptable to greatly reduce the image quality, so that a practical transmission time can be achieved. In this case, low quality settings would be used, even if this incurs noticeable blocking artifacts. This is an example of “time is of the essence.”

2. Suppose a colleague wants to add an animation at the last minute to an already-completed presentation for a small meeting. In this case usage of average compression settings would be a reasonable tradeoff, as this would yield acceptable image quality without a large file size. Since the animation will be seen by a relatively small number of people only once, it is not essential that it be perfect image quality.
3. The final example is for CD-ROM production. In this case, you would want to get the best image quality possible. You would use high quality settings without undo regard for file size, since the CD-ROM has such a large capacity and it will be distributed to many people.

In addition, after an animation is rendered in QuickTime Movie format, if it does not play fast enough , you can re-save the animation to a new file to compress it further.

When saving in Movie format, the individual frames of an animation being rendered are compressed frame by frame using a proprietary Apple algorithm. An additional compression step, called "frame differencing," can be performed on existing animations by using the "Save A Copy As" command.

Frame differencing stores a complete frame once each second (in this case, called a key frame), with in-between frames containing only the changes in the image from frame to frame, using the last whole image as a reference. Each second starts the process over again. Depending on the number of frames per second, the data savings can be substantial. By storing the difference between frames rather than a full image every frame, the animation file can be much smaller. And playback is faster. Actual compression ratios and achieved performance are highly dependent on the type of images in the animation.

There are a couple of considerations when using frame-differenced compression. Visual anomalies may occur in the animation playback when frame skipping is allowed, since the image on the screen must be sequentially constructed from one frame to the next. If this is a problem, disable the frame skipping option in the Animation menu to force QuickTime to play every frame of the animation. While it may play slightly slower in this mode, the animation will always appear intact.

Using QuickTime for still images

QuickTime compression for PICT and PICS files is enabled by a check box in the secondary Save dialog box, but is only available if the QuickTime extension is present. The compression method for these files use the Joint Photographic Experts Group (JPEG). This is a "lossy" method of compressing images. Compression is achieved by removing redundant or imperceptible data from the image, which must be reconstructed upon expansion by extrapolating the remaining pixels. Depending on the amount of compression, JPEG can affect an image's appearance. However, Vision3d gives priority to maintaining image quality and does not compress its files as tightly as may be possible.

No compression is available for still images saved in TIFF format.

Playing a compressed PICS file from hard disk is not recommended without specific hardware support — i.e., a NuBus card containing the JPEG compression/decompression code in ROM. Without hardware support, the time required to expand each frame virtually eliminates any kind of smooth, flowing motion.

On the other hand, compressed PICS animations are significantly smaller and easier to transport…and they work just fine when played from memory. In Vision3d, this is done by first opening the file, then using the "Playback from Memory" command in the Animation menu.

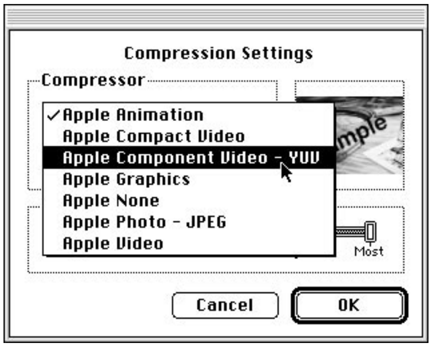
Explanation of QT compressors

Several software compressors are provided through the QuickTime extension. These are listed and chosen in the pull-down menu of the Compression Settings dialog. Each compressor was designed for a specific type of use. Using a compressor not intended for your needs is not wrong, per se, but using the right compressor can significantly improve playback performance.

The following explanations are provided for background:

- **Video compressor.** Of all the available compressors, the Video compressor is best suited for capture and compression of analog video, high-quality playback from hard disk and moderate to high quality playback direct from CD-ROM. This compressor can play back at frame rates of 10 fps or better, and can be transcoded to achieve further compression and to permit the ability to play back directly from CD-ROM. A file compressed with this method is re-compressible with minimal or no quality degradation, allowing digital effects processing and editing to be applied in repeated passes.
- **Compact Video compressor.** This compressor is an extension of the Video compressor. All characteristics and uses identified for the Video compressor apply here as well, with one exception. It is achieves a better image quality for a given compression level because it has been optimized for 16-bit color ONLY (thousands of colors). The dithering algorithm is significantly improved because of this optimization.

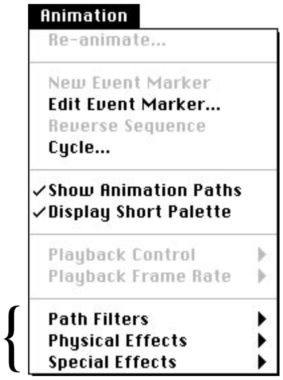
- **Animation compressor.** The Animation compressor is best suited for images that are originally created in digital form, such as rendering. It performs best on images that have no noise. The Animation compressor can be used to compress digitized analog video, but it may not produce high compression ratios from that type of material. If the Animation compressor is used for captured analog video, it will produce very good picture quality which can be played from a hard disk. It is unlikely that this compressor will be fast enough to compress and/or play analog video sequences from CD-ROM at sufficient speed.
- **Photo compressor.** Of all the currently available compressors, the Photo compressor produces the highest quality images at the best compression ratios. It is better know as JPEG compression. However, it is not currently fast enough to play back moving image sequences with sound at acceptable frame rates. Because it does achieve such a good result at high compression ratios, the Photo compressor can be used effectively as an archival method for video clips with high disk space requirements. JPEG is supported on several platforms, making this compressor a viable interchange format between computers of different manufacturers.
- **None compressor.** The None compressor is a very good format to use for acquisition of analog video. It is also good for digitally-generated animations. It has the advantages of having a very good image quality (since no compression is applied), it can capture in real-time with sound on higher-end machines, and it can be further transcoded off-line to produce movies with much better compression ratios. The disadvantage is the high disk space requirement.
- **Graphics compressor.** The Graphics compressor is a good format for acquisition of analog video when there is a need to have very good picture quality for playback on eight-bit displays. To achieve the best results, it is necessary to capture the video at 24 bits per pixel, apply error diffusion dithering as it is converted to eight bits per pixel, then compress the result. Although the Graphics compressor was intended primarily for use with 8-bit source material, it can be effective for video in this context. It does not achieve high compression ratios for video, so it is not currently suitable for playback directly from CD-ROM. It will work well for play back from hard disks.
- **Component Video —YUV compressor.** By default, the YUV compressor does not appear in the standard compression type list. If you hold down the Option key while clicking on the pull-down menu, it will appear.



The YUV compressor is somewhat specialized. It is useful with certain video-editing solutions. It makes a good intermediate storage format if you are applying multiple effects or transitions to an animation. The compression ratio is always 2:1 and the image quality is extremely high. However, it does not support frame differencing.

Animation Extensions

Installed animation extensions are accessed by commands that appear at the bottom of the Animation menu, as well as buttons on the Button bar. Vision3d includes several extensions in the package as part of the standard product. As other extensions become available and are installed in the Extensions folder, they also will appear as commands in one of the sub-menus on the Animation menu.



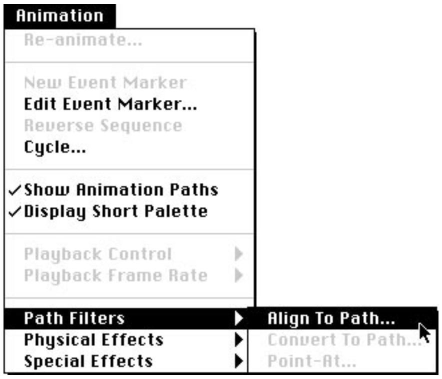
Animation extensions are grouped in three sub-menus at the bottom of the Animation menu

In general, extensions provide their own dialogs and menu bar allowing the construction of new objects, for converting existing objects to a type supported by the extension, or for further editing by the same extension. All necessary tools required to perform the work for which the extension is designed are also provided in the dialogs. When editing of the object is complete, the extension inserts the object back into the model space with the changes you specify.

The standard animation extensions are discussed in this section of the chapter, in alphabetical order. Other after-market or third-party extensions include their own documentation explaining how to use the animation techniques provided by that extension.

Align to Path

When the Align to Path extension is present in the Vision3d Extensions folder, the command to open the Align to Path extension appears in the Path Effects sub-menu in the Animation menu.



If Align to Path does not appear in the Animation menu, the extension is incorrectly installed.

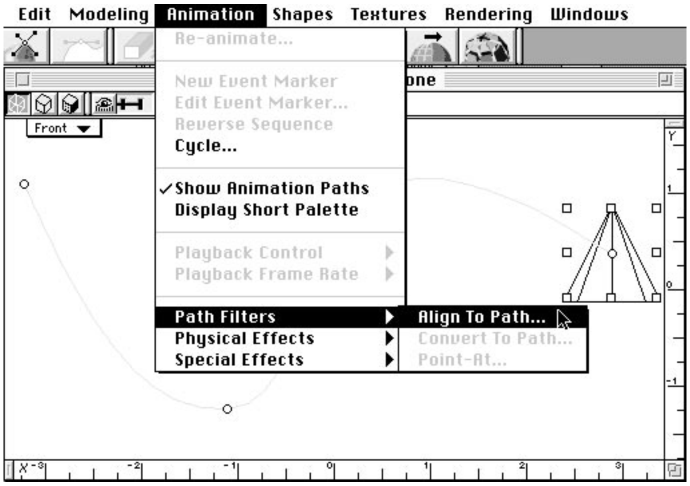
Description

As you might expect from the name, the Align to Path extension allows you to align an object in a Vision3d model to its defined animation path. Unlike the Convert to Path extension, which allows you to attach an object to an animation path, Align to Path provides control over how the object is oriented on an animation path previously defined through the Animation palette. In other words, you can control which direction the object is pointing as it travels along the animation path. As your object moves, do you want it to point forward, backward, up or some other direction? These object orientation options are provided by the Align to Path extension.

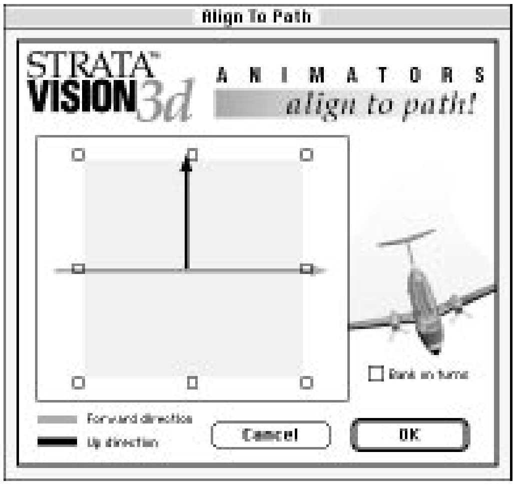
This extension also gives you an option to have your object “bank” on turns as the animation path changes direction.

Using Align to Path

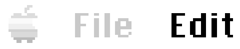
To use Align to Path, you must already have an object and its animation path present in a Vision3d model. When the object is selected, the Align to Path command becomes available.



Choosing the command opens the Align to Path dialog box. A representation of the selected object appears in the Align to Path dialog, with controls to initially orient the object relative to the animation path

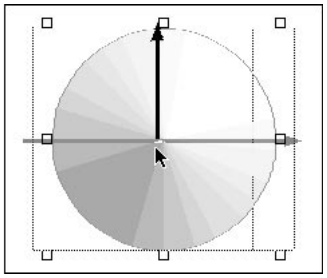


When the Align to Path dialog opens, the standard Vision3d menu bar is also replaced by one that supports the extension.

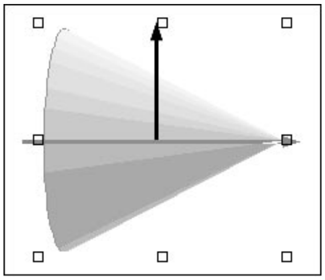


The extension's menu bar contains only three menus. The first two are dimmed because none of their commands are active while in the extension. The Undo command in the Edit menu is the only usable menu command in this extension.

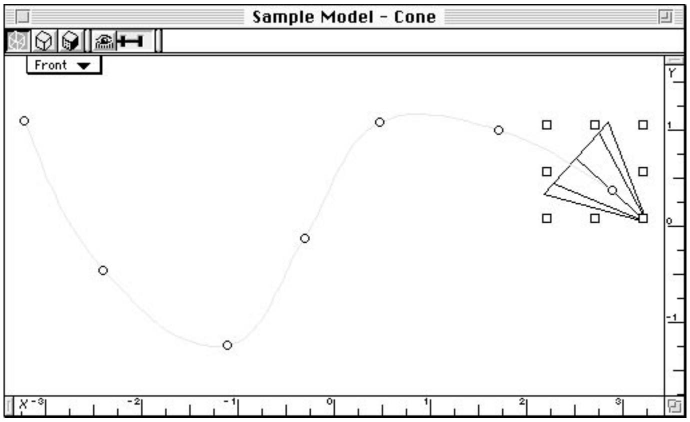
The selected object in the model (in this example it's a cone from the model illustrated above) is displayed in the window on the left side of the dialog. Selection handles are overlaid on the selected object. To change an object's orientation on its animation path, use the cursor to click on any of the selection handles. Then drag to rotate the object in the desired direction.



When you release the mouse button, the extension immediately redraws the object showing its new alignment.



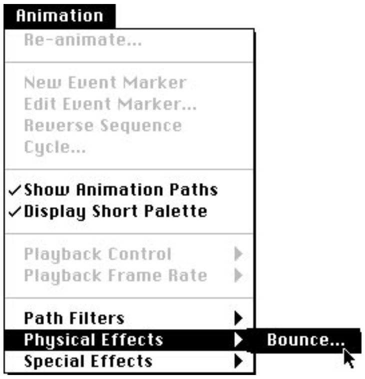
Two directional pointers — one indicating the forward direction and one indicating the up direction — are also overlaid on the selected object. These pointers remain constant to assist you in orienting the object to the path in the extension window. When you've aligned the selected object as desired, click on the OK button to apply the changes and return to the model window. The object will appear with its new orientation on the animation path.



Note that any new event markers whose creation was necessitated by the alignment operation will now be indicated by additional event markers in the Animation palette on the object's animation timeline, as well as anchor points on the animation path in the model window, if it is showing.

Bounce

When present in the Vision3d Extensions folder, the command to use the Bounce extension appears in the Physical Effects sub-menu of the Animation menu.



If the Bounce command does not appear in the Animation menu, the extension is not correctly installed.

Description

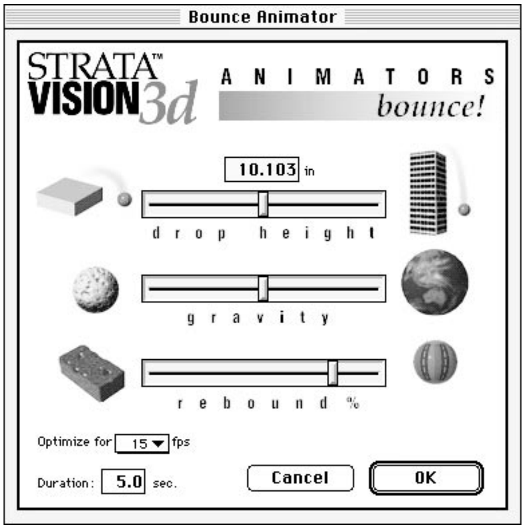
The Bounce extension creates the animation path to bounce an object. Event markers are generated and placed in the appropriate order and position on the selected object's time line causing the requested bouncing motion. Note that there is no collision detection; it is your responsibility to specify the parameters so that the object’s bounce appears correct for the scene you have assembled.

The advantage of the Bounce extension is that it performs all the calculations to automatically create a realistic bounce motion for a selected object. The event markers are generated and laid down on the time line for that object in the Animation palette. The bounce calculations even extend to setting the motion controls for each of the event markers found in the Event Options dialog, i.e. — spline or linear, spline tension, ease in/ease out, etc.

Only one object can be “bounced” at a time. If an animation path is already defined for the an object, it will be fully or partially replaced for the specified interval with event markers generated by this extension. If the original animation path extends longer than the that specified for the bounce effect, those event markers after the bounce interval are left intact, but the new animation path may become quite erratic during transition from the new path created by the Bounce extension back to its original path.

Using Bounce

To use the Bounce extension, in the model or a shape workspace, you must already have an object drawn or inserted in the workspace. First position the time pointer when you want the bounce to begin and select the object. Then choose the Bounce command from the Animation menu. A Bounce dialog box opens to set the motion parameters.



Bounce Animator

drop height: 10.103 in

Optimize for 15 fps

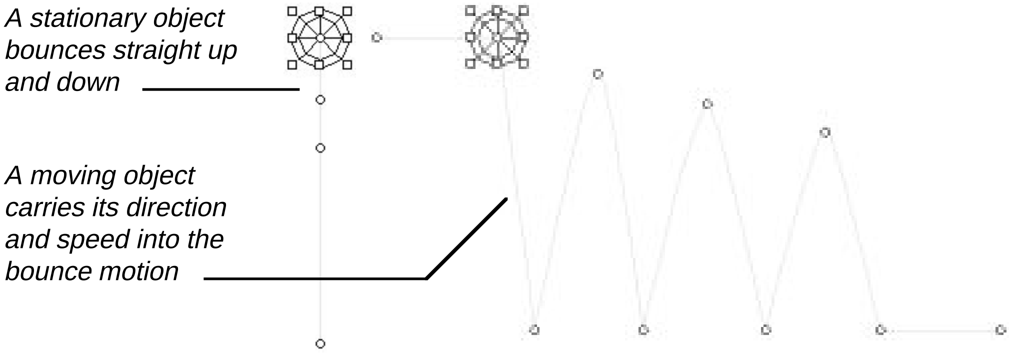
Duration: 5.0 sec.

Three slider controls allow you to set the drop height, the gravity strength, and the rebound percentage (a.k.a "bounce-osity"). Fields are provided to set the frames per second and the duration to be used in the bounce calculations. The preferred frames per second setting is important because of temporal aliasing concerns — a mismatch between the object's bouncing motion and the fineness at which the tweening occurs during rendering can cause jumps in the bounce. The duration determines the maximum time in which the bounce motion will decay to zero. It may stop bouncing sooner than the duration settings, depending on the drop height, gravity, and rebound, but it will never go longer than the duration specified.

A field is also provided to enter the drop height numerically for greater accuracy. Note that the drop height is measured from the object's current position at the time this extension is called.

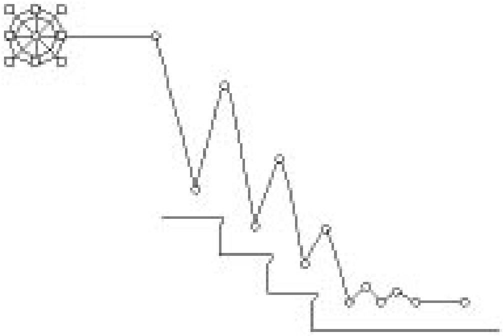
Carrying object motion into the bounce

If an object is stationary at the time the Bounce extension is called, it will bounce straight up and down. If the object is moving at the time it begins bouncing, that motion (both speed and direction) is added to the bounce calculations. The same motion is maintained after the bounce decays, unless other event markers are already defined to constrain its motion in a different way.


A stationary object bounces straight up and down

A moving object carries its direction and speed into the bounce motion

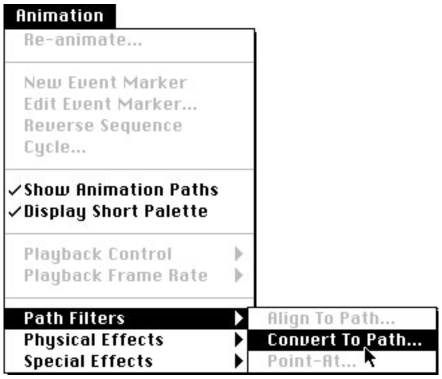
Of course, once the event markers are in place and the animation path is displayed in the active view, the path is just like any you would create manually. If desired, you can move the markers on the palette or drag the anchor points on the path on any axis to fine-tune the motion any way you want.



For example, suppose you want a ball to bounce down some stairs. There is no provision in the Bounce extension to set a bounce height that changes, like it would on stairs. But after the Bounce extension is done, you can click and drag the individual anchor points on the animation to match the step on a set of stairs, as shown in the previous illustration.

Convert to Path

When the Convert to Path extension is present in the Vision3d Extensions folder, the command to open the Convert to Path extension appears in the Animation menu.



If Convert to Path does not appear in the Path Filters sub-menu of the Animation menu, the extension is incorrectly installed.

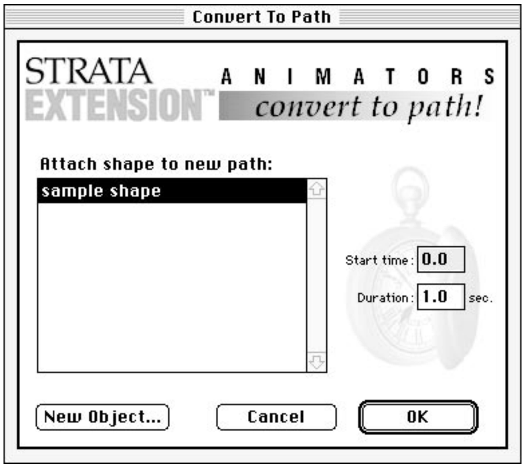
Description

The Convert to Path extension converts any 2-D line to an animation path. The line can be drawn in Vision3d or imported from another source. It can be an open line or a closed polygon, or even a closed bézier spline. Once the line is converted to a path it is no longer an object, and thus does not exist in the model as a physical entity.

The extension allows you to assign a shape, camera object or light source to the new animation path. The assigned object may already exist in the model, or can be initiated from within the extension. Convert to Path also allows you to set a start time when the object will begin its motion on the assigned animation path, and a duration for the motion. These time options are particularly useful when you're coordinating motion scripts within complex animations which have more than one object moving simultaneously. Only one object can be assigned to each path. But the line used to make any path can be duplicated and re-used multiple times as paths within a model.

Using Convert to Path

To use the Convert to Path extension, in the model or a shape workspace simply draw or import a 2-D line you wish to use for an animation path. With the line selected, pull down the Animation menu and choose the Convert to Path command from the Path Filters sub-menu. Vision3d opens the extension's dialog box.



Attach shape to new path:

sample shape

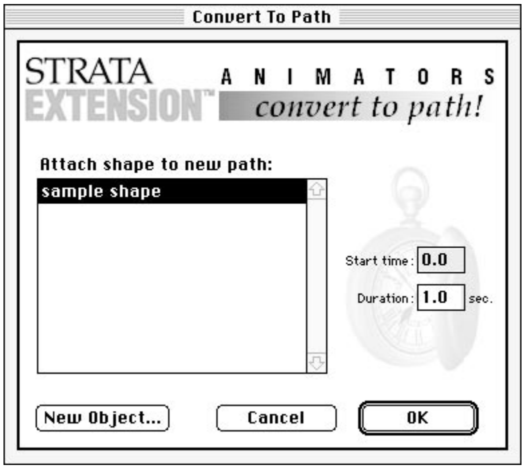
Start time: 0.0

Duration: 1.0 sec.

Additionally, when the Convert to Path dialog opens, the standard Vision3d menu bar is replaced by one that supports the extension.

The extension's menu bar contains only three menus. The first two are dimmed because none of their commands are active while in the extension. The Undo command in the Edit menu is the only usable menu command in this extension.

If no objects exist in the model that can be assigned to the path, the dialog opens with an empty directory. If one or more objects that can be assigned to the animation path already exist in the model, it opens with the objects listed in the dialog.


Convert To Path

Attach shape to new path:

sample shape

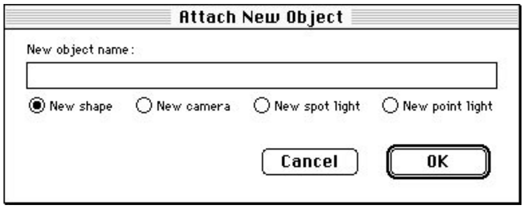
Start time: 0.0

Duration: 1.0 sec.

To attach one of the listed objects to the animation path, choose that object on the list and click on the OK button. The chosen object is attached to the path at the beginning point of the path.

The direction of movement along the converted animation path is determined by the direction that it was drawn, from its beginning point to its ending point. In the case of closed polygonal or bézier objects, the object travels from the beginning vertex point to the ending vertex point. Of course, you can change the location of those points on the object with the 2d Sculpter modeling extension before converting it to an animation path.

Regardless of whether objects are listed in the dialog window or not, you have the option to initiate new objects from within the extension. Clicking on the "New Object" button opens the Attach New Object dialog box.


Attach New Object

New object name:

● New shape ○ New camera ○ New spot light ○ New point light

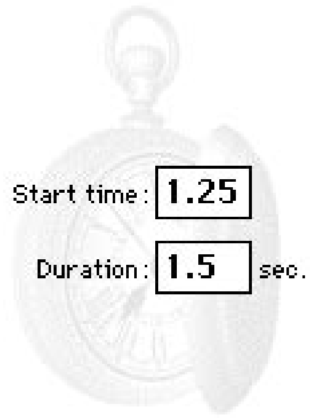
By clicking on the appropriate radio button, you can initiate a new shape, camera object or light source. If you are initiating a new shape or camera object, you must also give it a name. In the case of a new shape, a cube primitive is automatically drawn as a “place holder” in the shape’s workspace. You can replace the cube with your own object(s) later.

Clicking on the OK button inserts the new object into the model and attaches it to the animation path at the beginning point of the path.

Starting and Duration times

The value in the Start Time field is the number in seconds on the animation timeline at which the assigned object appears (or is born) and begins moving on its animation path. The value in the Duration field is the length of time in seconds over which the motion on the animation path will occur.

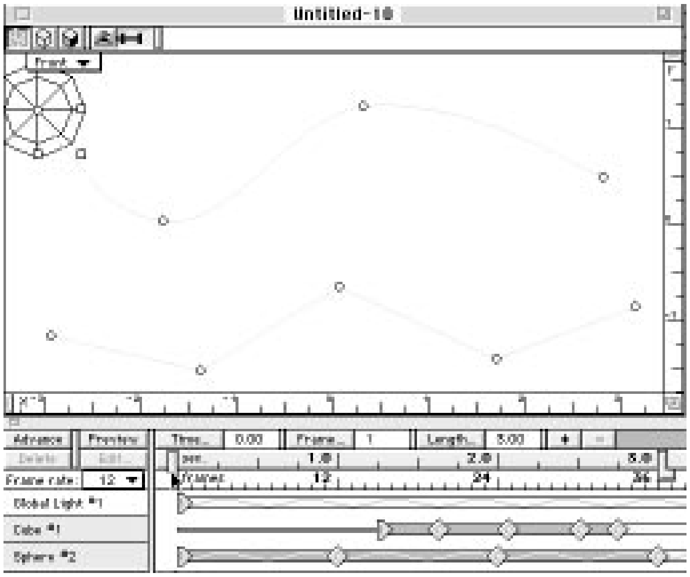
Here’s an example. In scripting a simple three-second animation of two objects — a sphere and a cube — the sphere’s motion is scripted over the entire length of the animation timeline. However, the values for the cube were set for a 1.25 seconds and 1.5 seconds (starting and duration times, respectively) in the Convert to Path extension dialog, as shown in the following illustration.



Start time: 1.25

Duration: 1.5 sec.

As a result, the cube becomes visible at 1.25 seconds, more than a second after the sphere has begun its motion (which is visible from time zero and throughout the entire animation). The following illustration shows the appearance of the window and timeline at time zero, with both animation paths visible. The sphere can be seen, because its life begins at time zero. While its animation path is visible, the cube is invisible because it isn’t scripted to appear until 1.25 seconds into the animation.

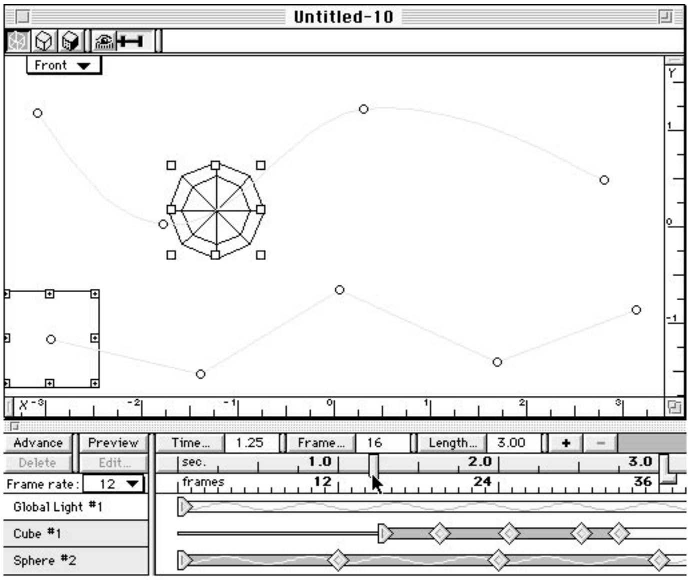


Untitled-10

Time... 0.00 Frame... 1 Length... 3.00

Frame rate: 12

Here’s the example again at 1.25 seconds. The cube appears right on schedule. The sphere is already well into its journey along its path.

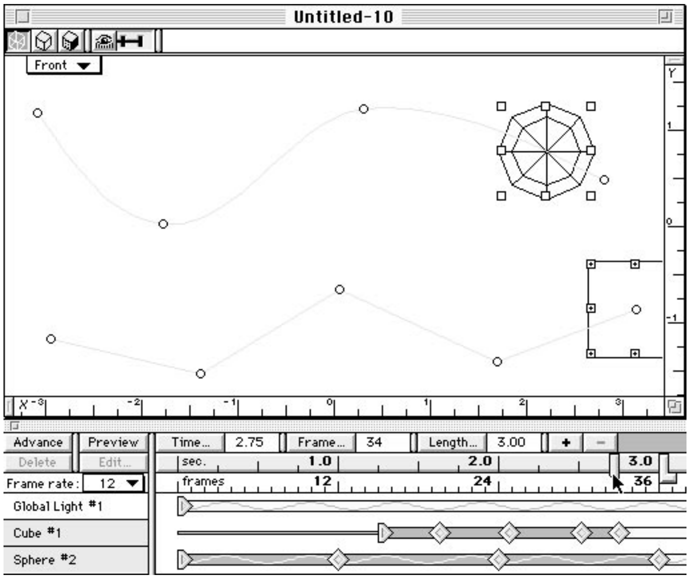


Untitled-10

Time... 1.25 Frame... 16 Length... 3.00

Frame rate: 12

Again at 2.75 seconds, as shown on the following page, the cube has reached the end of its path, while the sphere is still moving.



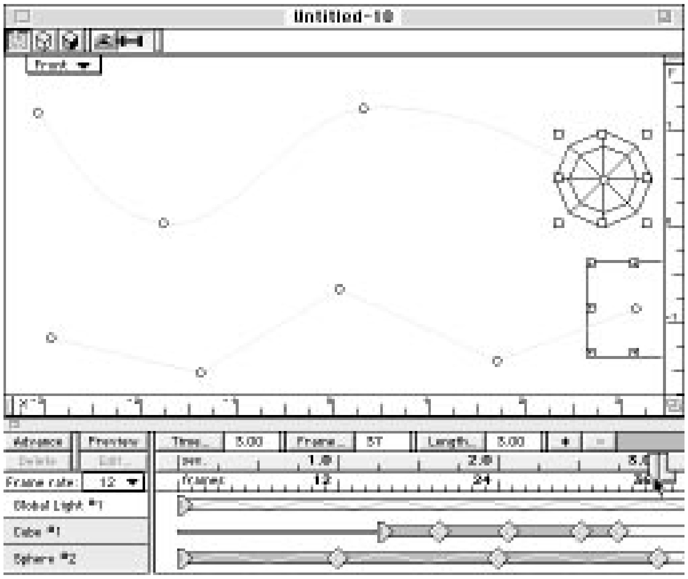
Time... 2.75

Frame... 34

Length... 3.00

Frame rate: 12

Finally, at three seconds, the sphere reaches the end of its animation path. Note that the distance traveled is independent of the time it takes to move that distance.



Time... 3.00

Frame... 37

Length... 3.00

Frame rate: 12

The spatial positioning of anchor points of a motion path is adjustable, as are the temporal positioning of event markers on the Animation palette.

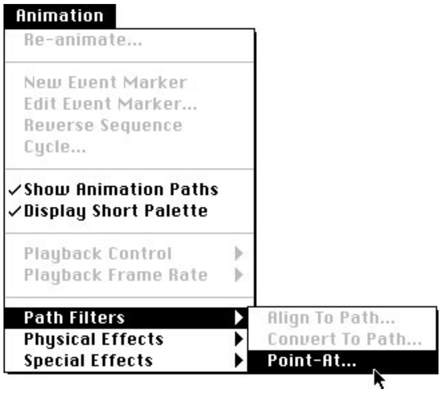
A caveat

When you attach an object to an animation path, the Convert to Path extension will create on the path as many event markers as needed to provide for a smooth animation. Because these event markers are created automatically, it's possible to draw a line that produces an excessive number of event markers. When you draw lines intended for use as animation paths, do so with the intent of minimizing the vertex points, which convert to event markers when made into an animation path. It's generally a good idea to check the number of event markers created during a conversion before proceeding. That way, if your path has too many event markers, you can use the Undo command and try redrawing the line so it will generate fewer event markers, or edit it in 2d Sculpter..

Alternately, you can selectively delete any redundant event markers from the Animation palette after the path has been converted. With experience, you will be able to decide which technique is most efficient.

Point At

When present in the Vision3d Extensions folder, the command to open the Point At extension appears in the Path Filters sub-menu of the Animation menu.



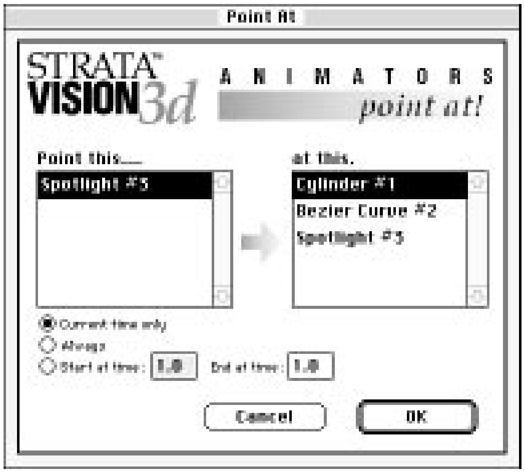
If the Point At command does not appear in the Animation menu, the extension is incorrectly installed.

Description

The Point At extension allows you to automatically point a camera object or local light source at an object within a Vision3d model. Directional/global light sources cannot be used with this animation extension. The advantage this extension offers is obvious. You don't have to manually script a light source's or camera object's animation to make it point at an object…checking several views of the model in order to line up a camera or light source just right. Point At will do all the calculations for you! This extension also provides control over when and for how long during an animation the camera or light source is pointed at an object.

Using Point At

To use the Point At extension, you must have a at least one object and one camera object or light source in a model. The camera object or light source must be selected. Then, when you choose the Point At command from the Path Filter sub-menu, the Point At dialog box opens.



● Current time only
○ Always
○ Start at time: 1.0 End at time: 1.0

As you can see, the names of camera objects and light sources appear on the left side of the dialog, while the names of objects at which they can be pointed appear on the right. Notice cameras and light sources appear on both lists simultaneously.

Once an object from the left-side list and one from the right-side list have been selected, you can click on the OK button to complete the operation. If you try to point a camera or light source at itself, Vision3d will prompt you to select another object at which to point the camera or light source. An object in your model can have more than one camera and/or light source pointing at it. But you can't point one camera or light source at more than one object.

If you have pointed a camera or light source at one object then decide to point it at a different object, simply select it again and open the Point At dialog to change the selection on the right-side list and click on the OK button. Alternately, you can choose the Re-Animate command in the Animation menu to change the Point At settings.

The time controls at the lower left of the dialog allow you to control when the camera or light source will point at the selected objects. By clicking on the Current Time Only radio button, you can point the camera or light source at an object at a specific moment in time (obviously, the *current* time as set by the time pointer). To point at an object over the entire course of an animation, click on the "Always" radio button. You can choose a specific portion of an animation during which to point at the selected object, simply by clicking the "Start of Time" radio button. For this option, enter a time value in that field and an ending time value in the "End of Time" field before clicking on the OK button.

This animation extension can also be used for non-animated objects. Simply point at the selected object at current time only. This is useful in projects that you only plan to render as still images. Since you are rendering only one frame, that current time will always be time zero.

A caveat

If you use this extension to point a camera object and/or light source at an object, they are not actually pointing at the object itself, but at the object's coordinates in the 3-D space of your model. This is an important distinction. If the object in that location is moved to a new location after the Point At extenstion is used, the animation path for the camera or light source will still be pointing at the now-vacated position — an empty spot in 3-D space.

If you move an object that has a camera or light source pointed at it, you must redo the animation path for the object pointing at it. This can be done with the Re-Animate command in the Animation menu.

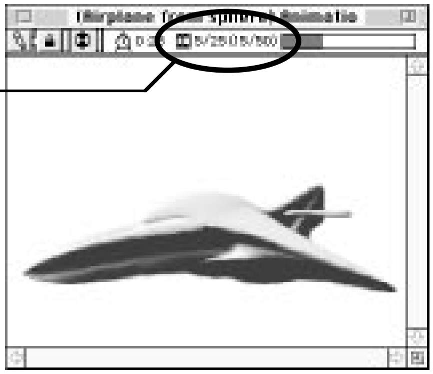
Commands and controls used with animations

The following commands and interface controls are considered relevant to the animation features of the software. Most of the commands are found in the Animation menu. The list of commands is alphabetical, and an explanation is included with each item in this section.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained more fully in chapter 1.

Animation windows (rendering and play back)

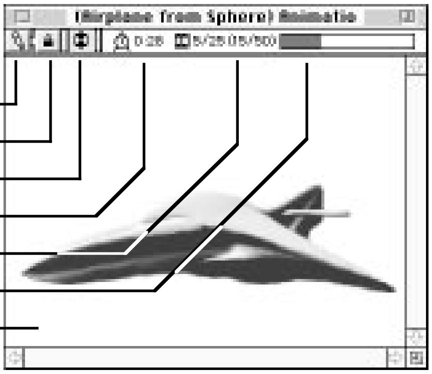
The specialized controls inside rendering windows are presented here. Mention is also made when the frame of the window types vary from the standard elements common to all windows on the Macintosh desktop.



Rendering windows for animations include frame information

…In this sequence:

1. Current frame
2. Frames to render
3. Current frame within larger animation
4. Total frames of larger animation



While a rendering is in progress…

1-time refresh button

Continuous feedback

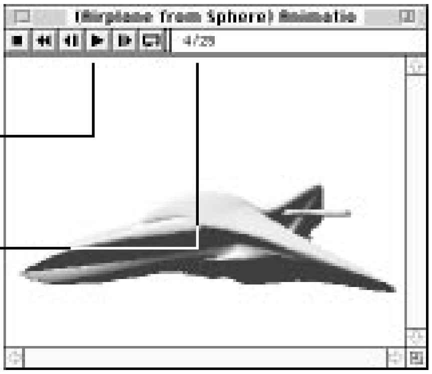
Activity indicator

Time estimate

Current proj. frames

Progress bar

Image display area



When an animation rendering is complete

VCR-style play back controls

Current frame & total frames in animation file

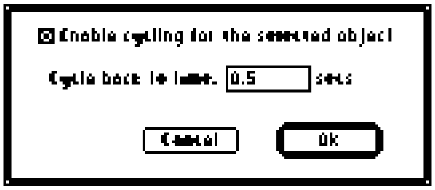
When a rendering is initiated, Vision3d opens a separate rendering window over the view selected. Both still images and animations use the same basic rendering window, with some additional animation information concerning the total number of frames to be rendered and the current frame being rendered.

A rendering window contains some specific controls that are unique to this type of window. These controls allow you to observe the rendering process as it occurs. Rendering windows serve as on-screen monitors during an image's creation. The main area of the window usually corresponds to the size of the rendered image. If the image size being rendered is larger than the screen, the rendering window controls will also include scroll bars and a re-size button to allow repositioning any part of the currently rendering image within the window.

As with any desktop window, a rendering window can be moved around the screen or to the background, even while the rendering continues. To terminate the rendering, simply click on the window's close box.

Cycle command (Animation menu)

The cycle command allows you to repeat a sequence of event markers for the length of the animation. Cycle is available when an object is selected for which two or more event markers have been defined.



☒ Enable cycling for the selected object

Cycle back to time: 0.5 secs

Choosing the command opens a dialog that contains a check box for enabling or disabling the cycle function and a field to designate the point on the time line at which the cycle is to repeat. When the last event marker is reached for the object, and the animation length is longer, the sequence is repeated from the time designated as often as needed to fill the balance of the animation length.

Cycle is useful for defining a repeating sequence that fills the entire animation without having to manually position event markers all along the time line. With this feature, you can set up a lead-in sequence, then cycle the main part of the animation over and over as needed.

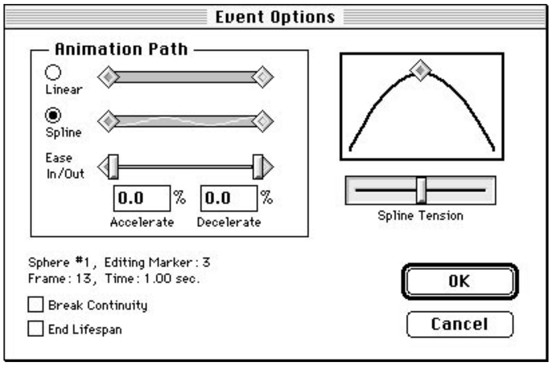
Display Short List command (Animation menu)

Use this command to display on the Animation palette only those objects in a model that have an animation path. This command toggles on and off each time it is selected in the menu. When enabled, a check mark appears next to it when the menu is pulled down.

This feature is useful when there are many objects in a model, but only a few of them have animation paths defined. Removing objects from the list on the Animation palette reduces the clutter.

Edit Event Marker command (Animation menu)

Use this command to open the Event Options dialog for the selected event marker. This command performs the same function as clicking on the Edit button found on the Animation palette.



Event Options

Animation Path

○ Linear

● Spline

Ease In/Out

0.0 % Accelerate

0.0 % Decelerate

Spline Tension

Sphere #1, Editing Marker: 3
Frame: 13, Time: 1.00 sec.

□ Break Continuity

□ End Lifespan

You may also double-click on an event marker on the Animation palette to open the Event Options dialog box.

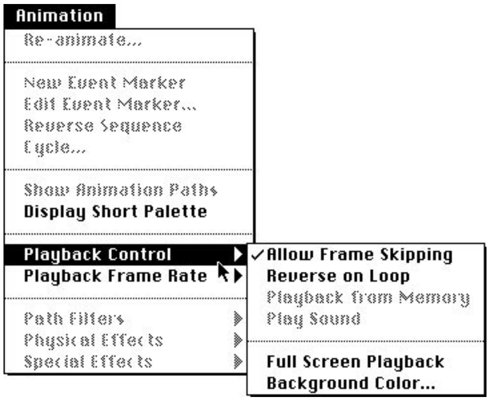
New Event Marker command (Animation menu)

This command inserts an event marker at the current position of the time pointer on the time line. The New Event marker command is available when the time pointer is not positioned on an existing event marker.

The current state of the object at that moment on the time line is used for the event marker. It is a useful way to generate an “anchor” event marker from which the object’s movements or appearance can make sudden changes later in the time line. Anchor-type event markers are used to keep later event markers from causing the object to change before you want.

Playback Controls sub-menu (Animation menu)

Use the commands in this sub-menu to control the play back of animation files. Some of the commands apply only to PICS-formatted files; some only to QuickTime Movies. As per interface guidelines, if a command is dimmed, it is not available in the current situation.



The Playback Controls menu item is active only when an animation file is open and its window is the active window. An explanation of the commands found in this sub-menu follows.

Allow Frame Skipping command

Use this command to enable or disable frame skipping in QuickTime Movies. The command toggles on and off each time it is selected in the sub-menu.

When frame skipping is allowed, the sound channel(s) that may be part of an movie file are given priority. The sound is then played at the correct speed and the frames are played as they can to keep up, sometimes requiring a frame to be skipped here and there to maintain the correct time sequence.

No sound is available when this command is disabled. If this command is turned off, so that all frames of a QuickTime animation are played in sequence — no matter how slowly — the Play Sound command is dimmed automatically.

Reverse on Loop command

The Reverse on Loop command allows you to alternate an animation sequence forward and backward when the loop button it pushed on the animation window. Reverse on Loop is available when an animation window is open.

The Reverse on Loop command is a check item that toggles on and off each time it is selected from the menu. When enabled, a check mark is visible in front of it as the menu is pulled down. If more than one animation window is open at one time, this option can be set independently for each file.

When enabled, an animation sequence will play forward first, then reverse the frame sequences. When the first frame is reached again, the sequence reverses again to play forward. The cycle is repeated until the stop button is pressed on the window, the loop button is clicked off, or this command is turned off.

If the loop button is on, but this option is disabled, the animation will play the frame sequence to the end, then begin again at the first frame, repeating the cycle until the stop or loop buttons interrupt it.

Playback from Memory command

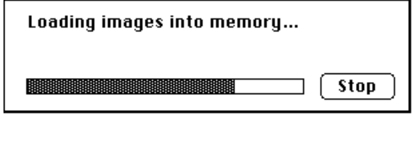
The Playback from Memory command allows you to load the frames of an animation into memory. Playback from Memory is available when a PICS animation file is open in a viewing window. Playing a PICS animation from memory is desirable, because the animation can then play at its true speed, producing a more pleasing appearance …and a more accurate representation of its creator's intentions.

Normally, only one frame of a PICS animation is in memory at a time. In this mode, each frame must be spooled through the window from the hard disk; causing the animation to play slower than it should and without any "flow" in the motion.

The Playback from Memory command is a check item that toggles on and off each time it is selected from the menu. When enabled, the command has a check mark in front of it as the menu is pulled down. If more than one animation window is open at one time, this option can be set independently for each file.

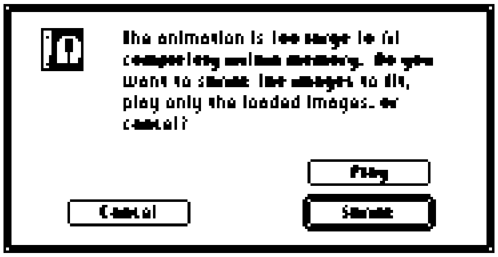
The color depth of the monitor on which the animation is to be played determines the color depth of the images as they are loaded into memory, i.e. — an animation playing on a monitor set to 256 colors (8-bits per pixel) will take less room in memory than one set to 16.7 million colors (32-bits per pixel). Setting a monitor to a different color depth after an animation has been loaded into memory will cause Vision3d to re-load the images at the new bit depth. The color depth of the frames in the file stored on hard disk is unaffected.

Turning this option off for a window in which the animation has been loaded into memory purges the buffer allocated for the frames; playback for that window then reverts to spooling the file from disk.



Loading images into memory...

Selecting this command causes Vision3d to begin loading all the frames into a buffer in memory. The size of that buffer depends on the amount of free memory available. Under System 6.0, the memory comes from within the partition allocated for Vision3d. You can improve the probability there will be enough memory by specifying a larger partition in the application's Get Info dialog while on the desktop. Under System 7.0, the buffer is allocated from available memory outside the partition.



The animation is too large to fit completely within memory. Do you want to shrink the images to fit, play only the loaded images, or cancel?

If during the loading operation, there is not enough room to fit all the frames of the animation into memory, Vision3d gives you the option of reducing the frame size so they can all fit or to just proceed with playing the partial animation that will fit at full size into memory.

Choosing the shrink-to-fit option restarts the loading process so the frames can be downsized as they come in. Choosing the play option allows you to view that portion of the animation that was able to fit, but no more. Choosing cancel terminates the loading process and turns off the Playback from Memory command. Further viewing of the PICS animation reverts to reading the frames from the hard disk one at a time.

Play Sound command

Use this command to turn sound on or off when playing a QuickTime Movie file. The Play sound command is a check item that toggles on and off each time it is selected from the menu. When enabled, the command has a check mark next to it in the sub-menu. If more than one Movie file with sound included is open, only the animation in the active window plays the sound. This command may be set independently for each animation as its window is made the active window.

If the animation open in the active window is a PICS file or a Movie file without sound, the command is dimmed.

If the Allow Frame Skipping command is disabled, so that Vision3d is constrained to play every frame of an animation, this command is dimmed and Vision3d will not play the sound even if present in the animation.

Full-screen Playback command

This command provides the option for blanking the desktop behind the open animation window. The "Full Screen Playback" command is a check item. It toggles on or off each time it is selected.

Note that the size of the image frame does not increase to fill the monitor's screen, the normal desktop is just temporarily hidden by a solid color outside the animation's frame.

When full-screen playback is on, the menu bar is invisible. Menus are fully active, however, and can still be pulled down to choose commands. You can return to normal desktop operation by turning off full-screen playback through the Animation menu or by simply pushing the Escape (ESC) key.

Keyboard Equivalents for Animation Window Controls.

Keyboard controls have been added to make the operation of the Animation window easier, especially in full-screen playback mode. The equivalents are:

Start playing animation	UP arrow, RETURN/ENTER key
Stop playing animation	UP arrow (again) RETURN/ENTER key (again)
Next frame	RIGHT arrow
Previous frame	LEFT arrow
First frame (rewind)	DOWN arrow
Turn on looping	BACKSLASH key (\)

You can also simply click on the image with the mouse to start and stop the animation. If the full-screen playback is off — so that the controls at the top of an Animation window are visible — you will also see those buttons echo the keys as they are pushed on the keyboard.

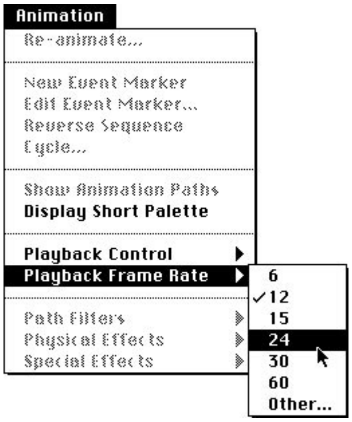
Background Color command

This command sets the color of a background that may be used behind an animation while it is playing in full screen mode. The default color is black.

The Background Color command opens the Apple Color Picker dialog box to select a color. The color choice is not retained. If you close an animation window, the background color reverts to black for the next animation window that is opened.

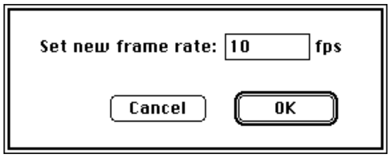
Playback Frame Rate command (Animation menu)

The Playback Frame Rate allows you to set the speed at which an animation will play. This command is available when an animation window is open and is the active window on the desktop.



When an animation file created by Vision3d is opened, the frames per second rate set in the Animation palette when it was rendered is automatically pre-selected in this sub-menu. You can use that setting, choose to use a different playback rate from the sub-menu, or set a custom frame rate through the Other option.

Choosing "Other" from the sub-menu opens a dialog in which you can enter the frame rate you wish for playback. A value between one and 60 can be entered.


Set new frame rate: 10 fps

You should know that the maximum speed at which the animation window can play an animation depends on the capacity of the hardware. For a PICS file to play at the speed specified by this command, it must be loaded into memory.

On the other hand, a QuickTime Movie file may be able to play in real time from hard disk, depending on the compression method used, the size of the frame on the screen, and if the hardware is capable of spooling the image data fast enough from the hard disk through the SCSI port and into video RAM. Some of the newer high-end Macintoshes are capable of this performance.

If a PICS animation is being played from hard disk, it will play as fast as the images can be read and displayed on the screen, without the benefit of file compression or spooling. It will be substantially slower than the target rate set here, no matter how capable the hardware.

Even when playing from memory, depending on the frame size, color depth, and speed of the CPU, few factory-standard Macintosh computers are able to display the frames fast enough to reach a true 60 fps play rate, which is the maximum allowed by the software. If the play rate chosen is faster than the hardware's ability to respond, the animation will be played as fast as possible, whatever that speed may be.

Re-animate command (Animation menu)

Use this command to re-calculate the most recent animation effect that has been applied to an object. This is useful if you wish to change the settings for a special effect provided by one of the animation extensions available for Vision3d.

If no animation effect has been applied to the selected object, the command is dimmed. If more than one object is selected, the command is also dimmed. It works on only one object at a time.

This command is specifically intended for use with the animation extensions installed with the software. The available animation extensions are listed in sub-menus at the bottom of the Animation menu.

Note that re-animating an object often requires more memory than the first time. That's because the first set of event markers and animation path generated by an animation extension is kept intact during the re-animate operation. If a re-animation effort is successful, the first animation set is deleted. If the re-animation fails for some reason, the first animation set is put back into the model

Reverse Sequence command (Animation menu)

The Reverse Sequence command allows you to reverse the sequence of a selection of event markers along the time line. Reverse Sequence is available when two or more event markers are selected. They do not have to be next to each other.

Selected event markers are physically moved on the time line to their new locations. The reversal occurs symmetrically around the mathematical center of the selected event markers. The selected event markers do not have to be contiguously grouped. Un-selected event markers within the time span being reversed are left stationary.

Show/Hide Animation Palette command (Windows menu)

The Show Animation Palette command displays the Animation palette. This command is available when the Animation palette is hidden. Hide Animation Palette is available when the Animation palette is showing. Choosing the Hide Animation Palette command is the same as clicking on the close box of the Animation palette.

Alternately, you can use the show/hide button for the Animation palette found on the Button bar.

Vision3d displays only one Animation palette, even if multiple model documents are open. However, when the active model changes, the palette is redrawn to reflect the status of the current model. This and all the other palettes in Vision3d always manage the active document.

The Animation palette's show/hide status when a new session is started is determined by the previous session. If the palette was showing when the application was quit the last session, it will be showing when a model is opened in the current session. If it was hidden at quitting last time, it will be hidden on startup this time…the status is retained from session to session.

Show Animation Paths command (Animation menu)

Use to show or hide the animation paths scripted for objects. When enabled, the animation paths of objects may become visible when the object is selected. This command toggles on or off each time it is highlighted in the Animation menu. When on, a checkmark is visible next to the command in the menu.

Note that you must turn this feature off to allow the moving of origin points on objects. When this command is toggled on, the positions of ALL origin points are locked relative to their respective objects throughout the entire model.

CHAPTER 8: RENDERING IMAGES

The effort invested in creating a model — modeling the objects, applying textures, assembling a scene, and positioning lighting — inevitably brings you to that anticipated moment when you are ready to render the finished image(s). When done correctly, the result can be an incredibly realistic still image or an animation that breaks new cinematic ground.

When you click or drag the camera tool in a view, the “snapshot” event initiates a rendering and brings all the rendering prowess of Vision3d to bear on the model you have created. Think of the rendering process as developing the snapshot.

Obviously, the approach you take to set up the finished rendering varies with the ultimate use of the product. If the rendered image or animation will be viewed solely on the computer screen, the resolution and size may not be as critical as when the image is to be printed in a high quality, full-color sales piece or the animation is to be part of an industry video distributed to thousands of customers.

Note that no matter what its ultimate use, the basic steps for setting up and rendering your masterpiece are the same…only the size and resolution settings change.

Vision3d has nine different algorithms or methods available to render your images — from wireframe to Raydiosity™. All the renderers are built in (no extra cost), thus they are readily available for previews during modeling, scene construction and fine adjustments, as well as for the finished renderings. Notice that Vision3d’s Raydiosity renderer is trademarked. It is a proprietary implementation for calculating inter-object diffuse illuminations. The spelling is different from the radiosity algorithm developed at Cornell University.

The type of rendering method used determines the effects that will be seen in the rendered image. As more effects are requested (through the Rendering Options command in the Rendering menu), the quality of the image is enhanced. There is a trade off, however. More rendering effects usually mean longer rendering times. But there are ways to minimize the impact as you work to maximize an image’s photo-realism. This chapter discusses the available options and suggest ways to improve performance.

The actual time required to render an image or animation depends on many variables. Some of the variables directly affecting rendering efficiency are:

- **Platform and hardware configuration.** The primary factor in rendering speed is raw CPU "horsepower." Because Vision3d is so CPU intensive, it is obvious that faster Macintosh platforms will render in a shorter period of time than lower performance versions of the series. Hardware acceleration is probably the single most significant improvement that can be made to your computer when used for 3-D rendering purposes.
- **System software version and configuration.** Vision3d requires version 6.0.5 or later of System software. Later versions of the Mac OS have improved efficiencies and use enhanced editions of Color QuickDraw, QuickTime, and memory management. Parameters like RAM cache size, MultiFinder, and the number of active Inits or extensions also affect overall operating speed.
- **Amount of total memory and the partition size given Vision3d in MultiFinder.** The amount of RAM available to Vision3d determines how large of an image segment it can render at a time. Less memory means smaller buffers and smaller segments for rendering, requiring more disk access time to swap segments in and out of the virtual image cache on the hard disk.
- **Options chosen in Rendering Effects dialog.** Enabling a rendering effect means that the renderer must perform the calculations to include that effect in the image, even if no surface exists in the model using the effect. Disabling an effect allows the chosen renderer to ignore that effect, even if surfaces are defined that would use the effect. Reflectivity is a good example: even if there are reflective surfaces in the model, when reflectivity is not enabled as an option for an image, the renderer can pass over that part of its code; saving that much time in the process.
- **Size and resolution of the image to be rendered.** The rendering time is directly related to the number of pixels in the image. The number of pixels is mathematically determined by the image's dimensions and the dots per inch within those dimensions.
- **Type of objects in the model.** As discussed in other chapters, primitives use less memory and render faster than polygonal or bézier objects. Of the various primitive shapes, spheres render the fastest of all.
- **Number of objects/polygons in the model.** Models with many objects require more overhead to keep track of them. Vision3d must

make several decisions while rendering an image. One crucial decision is which surfaces are visible from the chosen viewing position and which are not. Another factor is which objects are in front and which are behind. If the objects are described using polygons, polygonal mesh, and bézier patches rather than primitives, the time required to sort through a model's internal database takes even longer.

- **Number and types of light sources in the scene.** The calculations performed on illumination increases exponentially with each additional light source. Because you can use an unlimited number of light sources, the potential for abuse is very high. Each light source must be accounted for. Spot lights have additional calculations related to their cone angle, penumbra effect and soft edges at the cone boundary.
- **Types of textures used for object surfaces.** Solid textures render faster than surface. Reflective and transparent textures add rendering time, especially when using raytracing for refraction. Textures using surface maps require additional buffers for the maps so the renderers can refer to them during the rendering process. Each surface map increases the model overhead.

Some of these factors can be altered or optimized to yield better performance for Vision3d without adversely affecting the image quality. They are presented here so you will be aware that you can improve performance by turning off options that are not actually required or removing objects that make no contribution to the image.

By making you aware of the things that most directly affect rendering speed, you can make more educated choices in the modeling and rendering processes. On the other hand, you can also be prepared to invest the time required to obtain the image worthy of the project.

Of course, you are encouraged to use as many rendering options as needed to obtain the desired results. The incredible range of options available in the software are what make Vision3d the superior product and photo-realistic standard that it is.

Much of this manual has focused on how to create objects, define textures, insert lights, etc., then assemble all the elements into an intelligible scene. This chapter now addresses how to render that scene into an intelligent image. There are many options controlling the rendering process. Rendering is a simple process when going with the standard settings provided in the software. But, because of its many photo-realistic options, Vision3d also offers much power to the advanced user as well.

There are three areas requiring attention when rendering an image from a three-dimensional scene:

- **What is the content of the image as determined by the camera position and focal length?** The camera is designed to imitate an actual camera and is the mechanism used to gather the 3-D scene into the 2-D image.
- **How will the pixels that represent the image be decided?** In other words, what rendering method will be used? The 2-D image to be rendered is defined by the rendering window. It is the frame of the image. The image created in the rendering window is analogous to photographic film in the process of developing.
- **The image produced must be accurate.** How good is the renderer? Does it faithfully capture the image you intended, or does it look obviously computer-generated? What can you do to enhance the look you're after?

A digital "pinhole" camera

The purpose of a camera is to obtain an image (2-D, of course) from a three-dimensional scene. Whether this is done in the physical world by clicking the shutter on a real camera or in the computer by clicking the mouse button, the steps are quite similar.

Instead of actual objects, lights and cameras, the rendering algorithms in Vision3d work on internal databases of non-existent objects drawn by you in 3-D space to resemble (or represent) something recognizable. The presence of light in the model, its interaction with surfaces, and its reflection toward the viewing position are simulated by algorithms that operate on this internal data.

The principal job of the Vision3d interface is to let you communicate enough information to form this replica, i.e. — object geometry (including scale and proportion), object position, surface descriptions, placement of light(s), as well as the camera's position, so it can be captured on the view plane used by the camera and rendered with total integrity to the model.

There are four factors that determine what a rendered image will look like:

- **Position.** In the real world, no matter how automatic a camera is, someone has to decide where to put it…
- **Orientation.** …and point it at something…
- **Field of View/Focal length.** …then decide how much of the world or scene is visible in the camera lens…

- **Resolution.** …and, of course, the type of film used determines the quality of the image.

The metaphor comparing the process of rendering computer-generated images and taking snapshots in the physical world is very accurate. However, because of hardware limitations, modeling in the computer dimension still requires some accommodation to the digital world represented in the model.

Better than a real pinhole camera

The camera design used by Vision3d is a simple pinhole camera. It was chosen because it is the most efficient, computationally speaking. With a pinhole camera, the pixel color at any given point on the viewing plane (or vice versa, in the rendered image) can be determined by projecting a straight line back from that point through the pinhole aperture and out into the model.

There are several advantages to simulating a pinhole camera on the computer. The most obvious enhancement to a real pinhole camera is an infinitely small aperture, which real cameras cannot match. Physical cameras have a measurably larger aperture that can focus objects within only a narrow range of distances. The extent of that focus range — known as the "depth of field" — is related to the diameter of the lens aperture, sometimes referred to as the "f-stop." The larger the opening, the narrower the depth of field. So, while physical cameras have limited ability to resolve detail with distance, by contrast the computer camera has perfect focus at all distances.

Greater focal range

Vision3d's version of the pinhole camera also provides for both orthographic and perspective projection. Hence, the adjustable focal length controls on both the modeling views windows and camera object window have a much greater range than can be achieved in the real world.

Orthographic projection uses an infinitely long telephoto lens. The stronger the telephoto effect, the more foreshortening is apparent in the image; causing all objects of similar scale to appear the same size regardless of their distance from the camera. On the other hand, perspective lens settings produce images in which objects near the viewing position appear larger than objects positioned farther away. Vision3d's extreme wide angle is likewise beyond the physical ability of today' lenses. The amount of perspective in a view is inversely proportional to the focal length used, and directly proportional to the field of view.

Chasing after light rays

Another significant difference that exists in Vision3d , when compared to the physical world, is the approach taken for dealing with light in a scene. Consider for a moment how you process images of the world around you. As you look around a room to form a visual image, you are sub-consciously asking yourself, “Where is the light going?” In the physical world, light is everywhere, with only some of it actually arriving at where you are standing.

The computer must take a more conservative approach when it analyzes how images should be formed. A much more efficient approach is to interrogate only the light rays striking the viewing plane and trace them back through the model to their sources. All other rays of light produced by those sources must be completely ignored. The computer must ask, “Where did the light come from?” Notice the difference? The computer cannot afford (and does not have the power) to spend time looking for and digesting light rays that never reach the camera’s lens. You can easily do this several times a second with your brain’s superior abilities, but the computer is too limited.

Shading objects

The rendering process calculates the appearance of objects in a scene in a specific lighting situation. It does this by calculating the nature of the light reflecting from a surface as a function of the light striking it. There are three main factors that must be factored into the equations used by the renderers in deciding how object surfaces should appear:

- **Color.** The interaction of the surface color with the light color is critical to the appearance of objects. The color of most real objects usually changes from point to point on their surfaces. To appear equally “real” in Vision3d, the renderers must be able to support complex texture mapping to reproduce that level of object detail.
- **Orientation.** The position and orientation of an object with respect to the light sources must also be considered. The greater the distance, or the more a surface faces away from the light, the darker the surface appears. On a curved object, the surface’s orientation changes from point to point. For some materials, even the color changes with orientation. This is true especially in areas of high specular reflectivity, called “hot spots.”
- **Smoothness.** The relative roughness or smoothness of a surface determines how light is reflected from it. A surface where the light’s

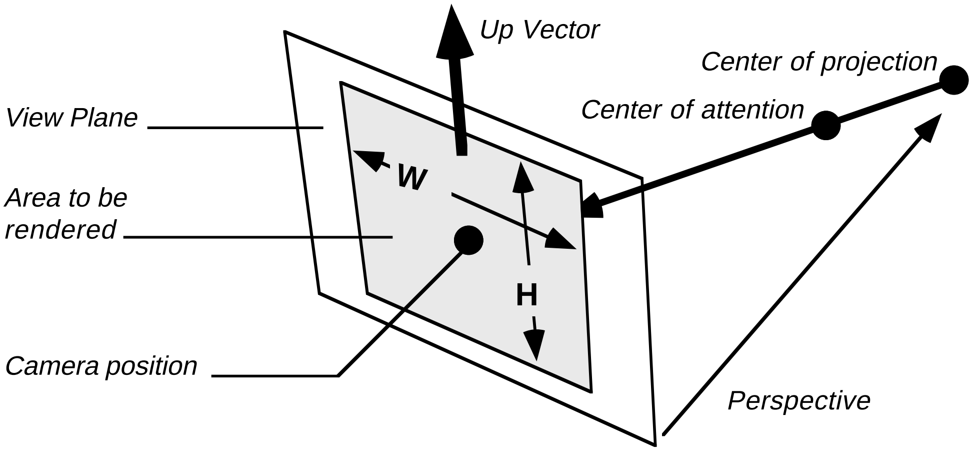
angle of incidence approximately equals the angle of reflection will appear shiny. In exact situations, the object acts like a mirror. A surface where the light is scattered at random upon reflection will appear dull.

The permutations of combining these factors are as endless as the surfaces they can represent. The appearance of the light reflecting from a surface carries important visual information about its composition. Often texture maps are the only way to represent surface variations too small or complex to be geometrically modeled. Determining the reflected appearance — that is, the direction and color of the light coming from a surface — can be complex and computationally expensive; hence the occasional extended rendering times. When done right, however, the results are rewarding…and very realistic.

The rendering methods used, the nuances of color and shading, the setting of each pixel in the context of the overall image — all these things exert powerful, yet subtle influences over the final appearance of an image. Time invested in fine-tuning texture settings, whether that takes the form of editing the fields and surface maps in the actual texture-editing dialogs or in their application to the objects through mapping style, orientation, and scale application, probably offers more opportunity for creativity and innovation than any other aspect of the modeling/rendering process.

Perspective

The viewing position apparent in the render image(s) depends on the active view used by the camera tool when taking the snapshot.



While you may use any open view for the viewing position of a rendering, camera windows usually lend themselves to this task better than modeling windows.

Vision3d's method of specifying the viewing position is similar to the way an actual photographer might think about positioning his camera. Seven parameters are obtained at the moment the camera tool is clicked in a view:

1. **The center of projection of the view used by the camera** (this corresponds to the location of its view set center in model space).
2. **The direction the camera is aimed** (what is the view plane's position in model space).
3. **The orientation of the camera** (which way is "up?").
4. **The distance from the objects** (determined by the view plane's distance from its view set center and that view set's position in space).
5. **The width of the area to be rendered** (determined by the view's width in the case of a single click in the view or the portion of the view defined by the drag marquee if the camera tool is dragged in the view).
6. **The height of the area to be rendered** (using the same data as in #5).
7. **The focal length of the view** (its view angle and resulting apparent perspective).

These parameters are made available to the renderer chosen to perform the rendering task. The data is then plugged into the rendering calculations; allowing the renderer to faithfully reproduce the viewing position chosen for the rendered image.

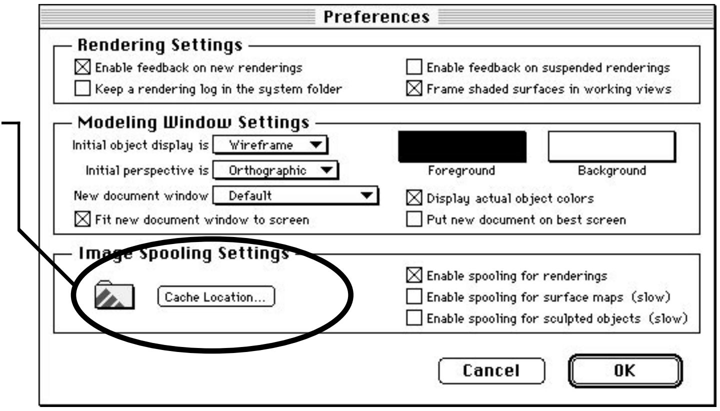
Internal image handling

When a rendering is initiated, the representation you see on the screen is just for your benefit. The image is rendered and stored internally in full 32-bit color or 8-bit grayscale, independently of the image displayed in the rendering window.

If spooling for rendering is enabled, Vision3d sets up a virtual image cache on your hard disk at the location designated in the Preferences dialog. You may locate the cache on any mounted volume that has sufficient free space available. This is where the rendered image is assembled as it is being rendered.

Set the virtual image cache location with this button in the Preferences dialog

…for use during renderings when the “Enable spooling for renderings” checkbox is on.



Preferences

Rendering Settings

☒ Enable feedback on new renderings

□ Keep a rendering log in the system folder

□ Enable feedback on suspended renderings

☒ Frame shaded surfaces in working views

Modeling Window Settings

Initial object display is Wireframe

Initial perspective is Orthographic

New document window Default

☒ Fit new document window to screen

☒ Display actual object colors

□ Put new document on best screen

Image Spooling Settings

☒ Enable spooling for renderings

□ Enable spooling for surface maps (slow)

□ Enable spooling for sculpted objects (slow)

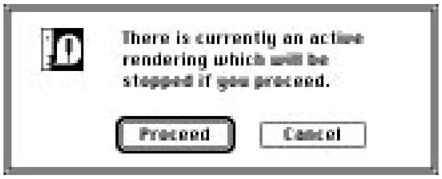
In addition, one or more buffers are opened in RAM to handle the various components of a rendering, i.e. — surface maps, a depth sort of the visible surfaces, light sources and their illuminations, etc. The size of these buffers depends on the following factors:

- **The actual algorithm performing the rendering.** Raytracing and Raydiosity typically require more memory than other rendering methods.
- **The size of the rendering.** Larger images contain more detail…and more pixels, thus requiring more memory to store.
- **The model’s complexity.** Each surface map requires one or more buffers for the renderer to reference.
- **The amount of memory available.** The more memory available (in and outside of the application), the faster it will render.

If the entire rendering can be handled in memory (buffers and all), no image cache is created. Tight memory constrains the size of the rendering buffers, relying more on the virtual image cache. If sufficient RAM is not available, Vision3d employs the virtual image cache to store and retrieve data on a hard drive during the rendering process. This method allows the image-creation process to continue. If there is still not enough RAM, even with the image cache, an alert appears and the rendering is terminated automatically.

Rendering in the background

One of the most powerful features of Vision3d is its ability to render images in the background. Any number of renderings can be initiated at the same time within the limits of available memory, each within a separate rendering window. The only limitation is that no alteration can be made to the model data during rendering.



There is currently an active rendering which will be stopped if you proceed.

If any model modifications are attempted while a rendering is in progress, an alert dialog is displayed warning of the consequences if the change is allowed — which is to terminate the current rendering, since the data upon which it is based would no longer be valid.

Background rendering frees the desktop's foreground environment to work on other tasks. These tasks may include using desk accessories, setting up views for other renderings or, if in System 6 MultiFinder or System 7, running another application.

Rendering in the background splits available CPU time among the renderings in process and the activity in the foreground. The foreground activity ALWAYS takes priority, but there are many CPU cycles that go unused. So, while background rendering speed is somewhat slower, it is better than none at all. If a particular rendering window is made the active window by bringing it to the foreground, it takes priority and all processing time will be given to that rendering until it is completed or until another window is brought forward instead.

Of course, it doesn't take a rocket scientist to figure out that the more images the computer is rendering at one time, the longer it takes to complete each one. In multiple rendering situations, it is often more productive to suspend the renderings, then queue them up to rendering one at a time overnight or at other non-use periods.

Suspending and restarting renderings

Vision3d also contains the option of suspending and restarting renderings as often as needed to fit your schedule. If you need to work on some other project that requires the total computer, you can suspend and save a Vision3d rendering at its current state of completion, then restart it later at the point where it left off.

In addition to the project flexibility this feature can give you, there are two more benefits of using suspended renderings:

- The original model does not have to be open to restart a suspended rendering. Because the file is self-contained, suspended renderings tend to be a little more memory efficient, and you can restart a suspended rendering on another machine.

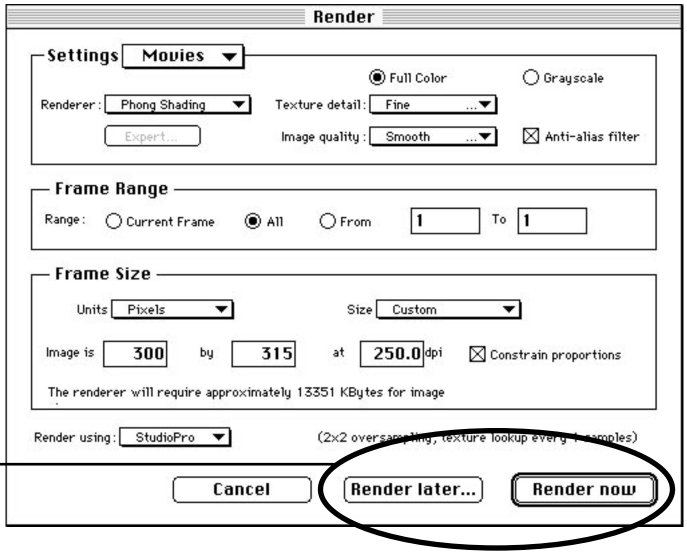
- Restarted renderings automatically save upon completion; replacing the suspended rendering file with a finished image. (That means, if a suspended rendering completes at 2:00 am in the morning and automatically saves, you won't lose all your work when an unplanned power outage occurs at 2:15 am!)

To suspend a rendering at the time it is initiated:

1. **Hold down the Shift key while clicking in the active view with the Camera tool.** This takes the entire view as the snapshot to be rendered. If you do not want to render the entire active view, you can use the Camera tool to drag a snapshot marquee in the active view. The marquee defines the extent by cropping in on the scene in the active view.

   Alternately, you can directly choose the Render command in the Rendering window. By design, this also defines the snapshot to be rendered as the entire active view.

2. **The Render dialog appears.** This dialog allows you to set the rendering parameters prior to beginning the actual rendering process. Note that the top section of the Render dialog is similar to the Rendering Options dialog. You can use a pre-set from the pull-down menu at the top of the dialog, or manual choose the settings.



Render

Settings: Movies

● Full Color ○ Grayscale

Renderer: Phong Shading

Texture detail: Fine

Image quality: Smooth

☒ Anti-alias filter

Frame Range

Range: ○ Current Frame ● All ○ From 1 To 1

Frame Size

Units: Pixels

Size: Custom

Image is 300 by 315 at 250.0 dpi

☒ Constrain proportions

The renderer will require approximately 13351 KBytes for image

Render using: StudioPro

(2x2 oversampling, texture lookup every [illegible] samples)

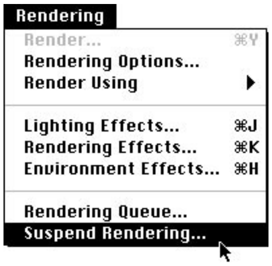
Render dialog has the option to render now or later

The Render dialog gives you the option of beginning the rendering immediately or deferring the rendering until a later time. Of course, you can also cancel this process, in which case no rendering is initiated and you are returned to the model to do further work.

If you choose to render later, a directory dialog appears asking for the name of the suspended rendering file that will be saved and where you want to store it. You are free to choose a location on any mounted volume. Assuming you have write privileges to the chosen volume, a suspended rendering file is saved containing the model data and the rendering parameters requested. No rendering work has yet been performed on it, but it may be opened at your convenience to begin the rendering.

To suspend a rendering already in progress:

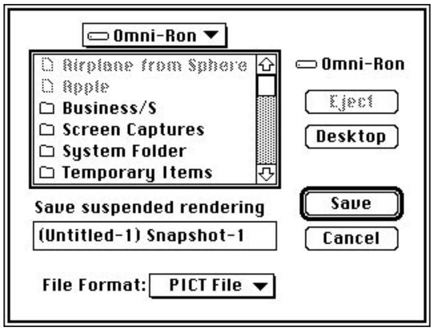
1. **Make the rendering window the active window.** If the window is not the active window, but is visible behind another window, just click the cursor anywhere in the window to bring it forward. If the window is completely hidden, use the Snapshot Windows sub-menu to select and make it the active window.
2. **Choose the Suspend Rendering command from the Rendering menu.** You are asked for a name and location to save the file.



Choosing the Suspend Rendering command normally will save the rendering project in its current state of completion and close the rendering window; thus terminating the project (for now).

Alternately, you can hold down the Option key while choosing the Suspend Rendering command. This saves the rendering in suspended rendering form (in its current state of completion), but will not terminate the project. The rendering will continue on after the suspended rendering file is saved. This feature may be used to create a backup of your rendering project, so that if anything untoward should happen before completion, such as a power failure or system crash, only the work performed since the last save will be lost.

3. **Click “OK” in the Save As dialog box.** The rendering is saved with all the data required to restart it at some later time.

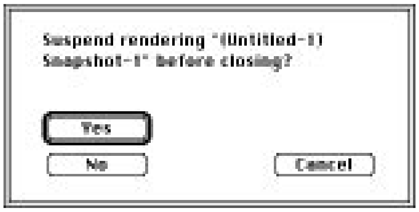


Save suspended rendering

(Untitled-1) Snapshot-1

File Format: PICT File

Another method for suspending a rendering in progress is to just close it. If the rendering has been going for more than four minutes, an alert appears asking if you want to suspend the rendering or just close it out.


Suspend rendering "(Untitled-1) Snapshot-1" before closing?

Clicking on the Yes button allows you to save a suspended rendering in the same manner as using the command does. Clicking on the No button closes the rendering window and throws away the working completed. If that rendering was from a previously-suspended rendering, the file is left untouched. Clicking on the Cancel button ignores the close attempt and continues working on the rendering.

Playing back partially-completed animations

The single frame of a still image is rendered to completion then saved, and the project is done. But an animation rendering is more complex. Vision3d must save each frame of the animation as it is rendered before proceeding to the next. The frames of an animation are appended to the end of the file as they render. If you stop an animation before completion, AND DO NOT SUSPEND IT AGAIN, you will be able to play the animation as far as it is done. The problem is you can't restart it, because it is no longer a suspended rendering.

If you want to be able to continue rendering an animation, you must suspend it. THIS IS IMPORTANT: When a suspended animation rendering is restarted and it renders and saves one more frame, the suspended rendering shell is stripped off the file. If you quit an animation in the process of rendering without suspending it, that is the end of the project. You must suspend it to be able to continue rendering when you start a new session. But here's the catch: Once suspended, the frames of an animation are not available for playback.

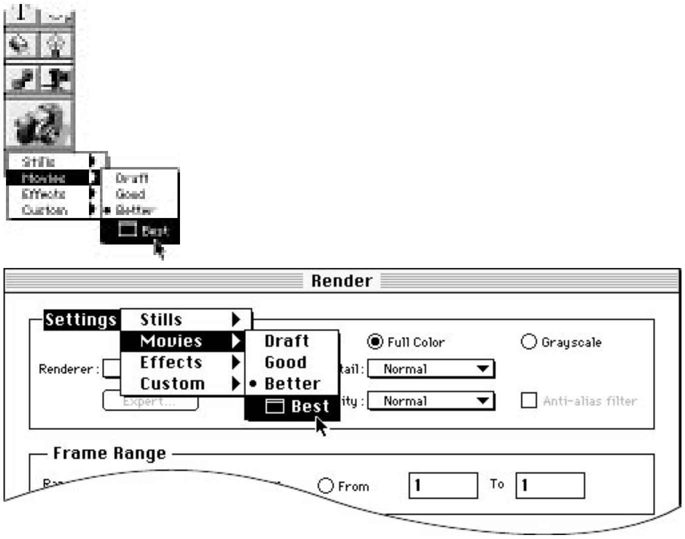
So, how do you play back a partially completed animation without ruining it for further rendering? It is a simple matter of returning to the Finder/Desktop and making a duplicate of the in-progress animation file while it is not a suspended rendering. Then bring Vision3d forward again and suspend the rendering. Now you are free to open the copy of the animation file and preview the frames that are completed to this point. If satisfied, re-open the suspended rendering file and continue on with the project.

Setting and using preset rendering options

The Preferences file residing in the System Folder contains some preset groupings of rendering options entered at "the factory." These settings are given names indicating their most appropriate usage, and are available for all models.

You may access the pre-defined rendering settings through a pull-down menu at the bottom of the Tool palette or through the Render command in the Rendering menu. The Render dialog contains a similar pull-down menu, as shown in the following illustration.

Choosing pre-set rendering options

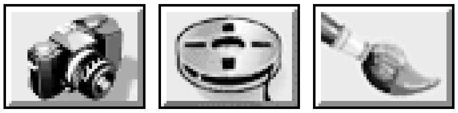


Use the pull-down menu on the Tool palette

…Or the pull-down menu in the Render dialog box

Choosing a rendering options pre-set from the pull-down menu on the Tool palette automatically selects the Camera tool. If you select a pre-set from the Movies sub-menu on that menu, the Camera tool changes to a movie reel to indicate a set of rendering options have been selected for rendering animations. If you select a preset from the Effects sub-menu, the Camera tool changes to a paintbrush to indicate one of the RayPainting™ effects has been chosen.

Camera tool's three icons



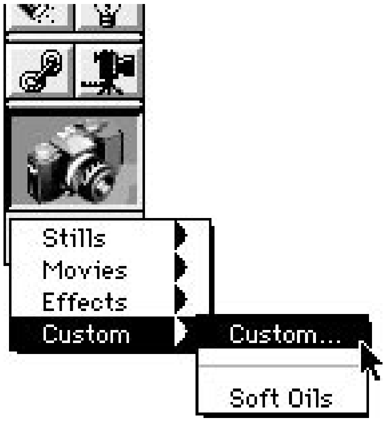
Stills *Movies* *Effects*

Choosing a pre-set rendering option causes a "bullet" dot to appear next to that preset. The dot remains until another pre-set is chosen, or unless the settings are manually altered in the Render dialog. If more than one of the pre-sets happens to match the currently-selected rendering options, dots appear next to all that match. Thus, it is possible that the "Movies•Better" can be the same collection of rendering option settings also used by "Stills•Draft."

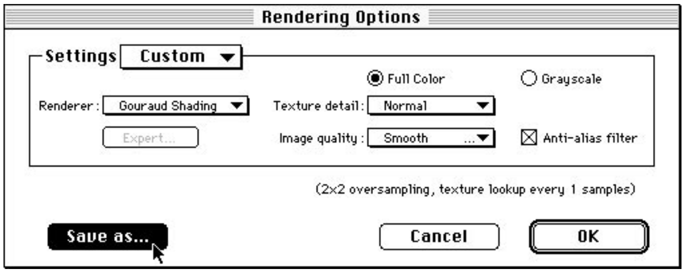
To switch to another preset of rendering options, simply click on this pull-down menu and select the desired entry. The factory pre-sets cannot be altered or deleted from the lists in the pull-down menu. However, the rendering options you define and save off can be deleted (see next topic).

Saving rendering options by name

If you assemble a particular set of rendering options that you would like to recall later — either in the current project or in another model — you may save them by name and attach that name to the list of pre-sets available in the rendering options pull-down menu.



The Rendering Options dialog box is opened by choosing the Custom entry in the pull-down menu on the Tool palette, or by choosing the Rendering Options command from the Render menu.


Rendering Options

Settings: Custom

● Full Color ○ Grayscale

Renderer: Gouraud Shading

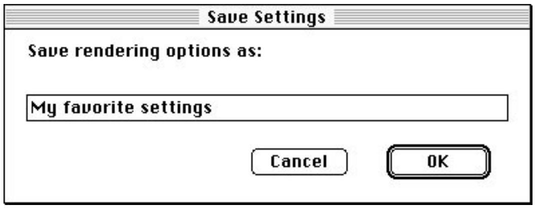
Texture detail: Normal

Image quality: Smooth ...

☒ Anti-alias filter

(2x2 oversampling, texture lookup every 1 samples)

Saving custom rendering options is done through the Save button in the Rendering Options dialog box. You will be prompted for a unique name under which to store the setting. Clicking "OK" adds the name to the list in the Set Rendering sub-menu.


Save Settings

Save rendering options as:

My favorite settings

NOTE: *Do not confuse the Rendering Options command with the Render command, both in the Rendering menu. It is true that they have some common elements in their respective dialogs, so at first glance you might think they do the same thing. Upon further investigation, you would find that the purpose of the Rendering Options dialog is to set (and optionally save) rendering parameters, while the Render command and its dialog is begin a rendering; no save function is provided in the Render dialog.*

Deleting preset rendering options

To delete an entry from the custom rendering options list, hold down the Option key while selecting it by name. Upon releasing the mouse button, the system "beep" will sound to confirm the deletion of that entry in the sub-menu list. This event is not undo-able, so you should be sure of your actions.

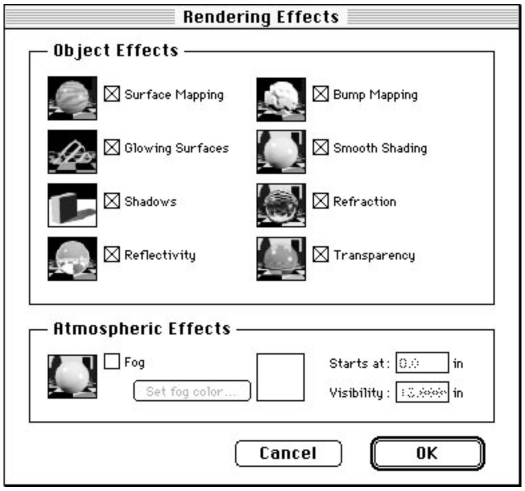
If you do change your mind, you can always add the preset again by manually enabling the same rendering options and saving them to the list.

Rendering options explained

The effects included in a rendered image are determined by the options enabled in the Rendering Options and Rendering Effects dialog. The rendering algorithm used must support the effects chosen. Conversely, if an effect is not enabled in this dialog it will not appear in the rendered image, even if the renderer supports it.

Turning rendering effects on and off

The Rendering Effects dialog contains check boxes to enable or disable the primary effects available in Vision3d. This dialog is opened by choosing the Rendering Effects command, or by double-clicking on the Camera tool.



Rendering Effects

Object Effects

☒ Surface Mapping
☒ Bump Mapping
☒ Glowing Surfaces
☒ Smooth Shading
☒ Shadows
☒ Refraction
☒ Reflectivity
☒ Transparency

Atmospheric Effects

□ Fog
Starts at: 0.0 in
Visibility: 10.000 in

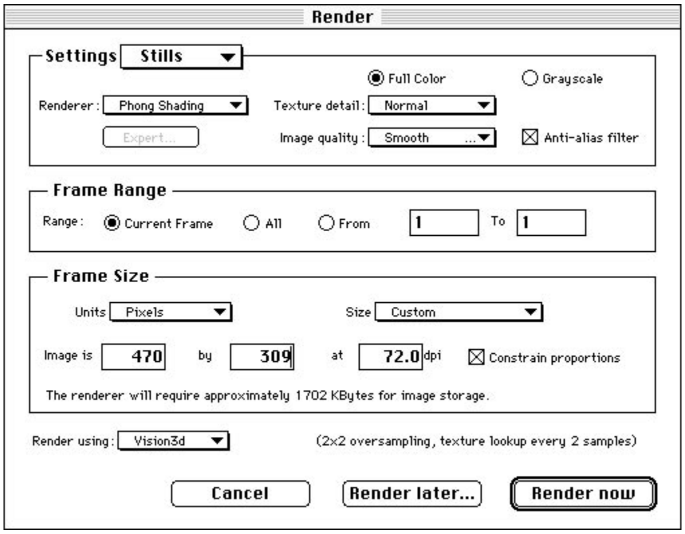
Whether a specific effect appears in a rendered image also depends on the rendering method chosen. Obviously, you cannot expect to see refraction with anything less than Raytracing. Or smooth shading when using the flat shading renderer. On the other hand, if refraction is turned off in this dialog, even if you do use the raytracer, it won’t appear in a rendering. The two must be coordinated.

To avoid confusion, each time Vision3d is launched for a new session, all of the rendering effects, except Fog, are automatically enabled. The on/off status of each effect can be set for the current session only.

Setting rendering parameters

Choosing the Render command initiates a rendering. It has the same result as clicking the Camera tool in the active view while holding down the Shift key. Either method defines a rendering window the same size as the active window in which you clicked.

You may not want to render an image the full size of the active view, or you may wish to crop in tighter, but still have access to the Render dialog to set the rendering parameters before beginning the actual rendering. You can crop the active view by dragging a snapshot marquee with the Camera tool. Holding down the Shift key as you click-drag the Camera tool allows you to crop and open the Render dialog.



Render

Settings: Stills

● Full Color ○ Grayscale

Renderer: Phong Shading

Texture detail: Normal

Image quality: Smooth

☒ Anti-alias filter

Frame Range

Range: ● Current Frame ○ All ○ From 1 To 1

Frame Size

Units: Pixels

Size: Custom

Image is 470 by 309 at 72.0 dpi ☒ Constrain proportions

The renderer will require approximately 1702 KBytes for image storage.

Render using: Vision3d (2x2 oversampling, texture lookup every 2 samples)

Within the Render dialog are fields to set all the relevant parameters for a rendering. From this dialog, you can adjust the following:

1. **Frame size.** You may designate pixels, inches or centimeters as the units for the image dimensions. Several pre-defined sizes are provided in the Size field pull-down menu. Selecting a size from the menu automatically resets the image dimensions in the height and width fields. You must return to the Custom setting to be able to edit the image's dimension fields.
2. **Base resolution.** This field allows you to adjust the number of dots per inch or dots per centimeter in the final image. This setting is the resolution of the final image and the level at which the anti-aliasing option interpolates its super-sampled pixels. Anti-aliasing is enabled with a check box at the top of this dialog.
3. **Render a single frame or multiple frames for an animation.** Each time this dialog is opened the Range radio buttons are active. The current frame is identified by the position of the time pointer on

the time line in the Animation palette. If the Length pointer on the Animation palette has never been advanced, there is only one frame available to be rendered, at time zero. If you have laid down event markers for an animation, you can choose to render some or all the frames in the animation, based on the animation parameters set on the Animation palette. Two parameters on the Animation palette affect the number of frames that can be chosen for rendering in the Render dialog: the frames per second and the animation length.

4. **Rendering method and quality level.** A pull-down menu provided in this dialog box allows you to choose the rendering method. Other fields in the Settings section of the dialog set the texture(s) sampling rate and use of anti-aliasing.
5. **Render now or later.** You can choose whether to proceed with the rendering now or save a suspended file for later rendering. If you choose to render now, the Render dialog box closes and a rendering window opens to begin the rendering process, as specified by the parameters in the Render dialog box. If you choose to render later, the Save As dialog box appears to name and save a suspended rendering.

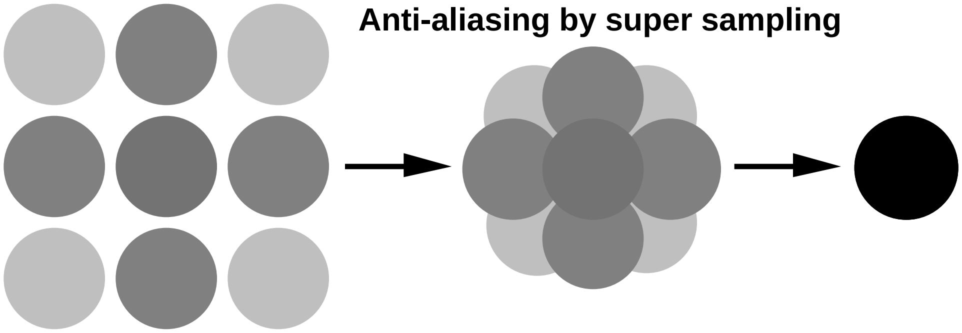
About image resolution and anti-aliasing

A computer-generated image is only as good as the “apparent” smoothness of its edges. Jagged edges usually result from not having enough information (resolution) to accurately describe the image. The process of smoothing edges in an image is called anti-aliasing. In addition to simply rendering at a higher resolution (a.k.a, the “brute force method”), there are two other ways to achieve a smooth-edged image:

- **Super Sampling.** With this method, the image is rendered with excess resolution. The extra pixels are then interpolated down to the designated base resolution. Super sampling requires that the image resolution be greater than the resolution of the medium on which it is to be eventually displayed or printed.

  For example, a 144 dot per inch (d.p.i.) image displayed by QuickDraw on a 72 d.p.i. screen will appear smooth because of the excess data available to the screen. You only see the 72 d.p.i. resolution, but the image appears smooth because of the super sampling effect produced by QuickDraw.

Anti-aliasing by super sampling



An Anti-aliasing Factor of three generates a three- by-three pixel grid centered around each pixel in the final image.

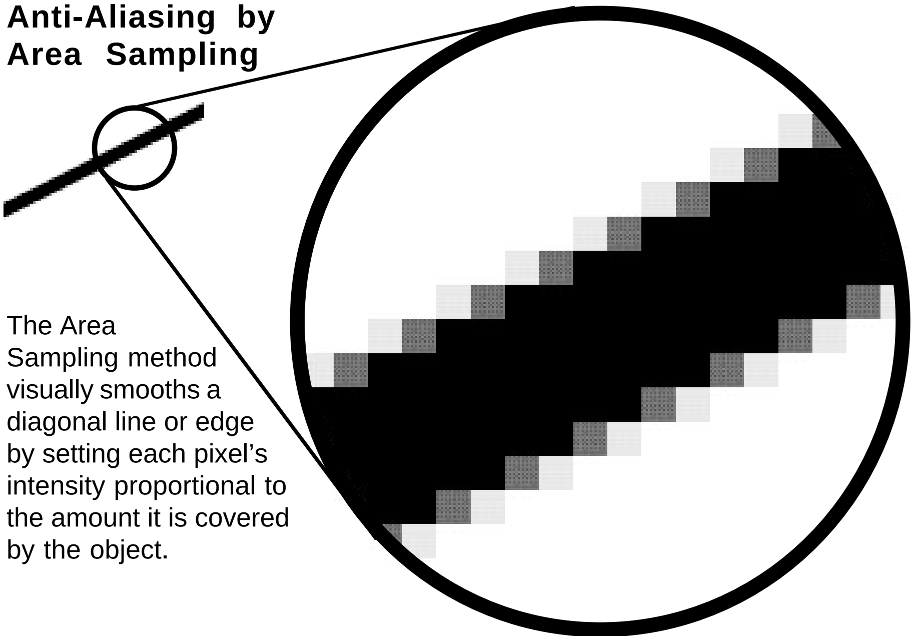
When the rendering is completed, the pixels are interpolated down to the one pixel used in the image.

The final image appears smooth because its pixels are the result of many extra samples taken during rendering.

In Vision3d, this is the same effect of rendering extra pixels and interpolating them down to the finished image.

- **Pixel Averaging.** This method alters the color of the pixels along diagonal lines or edges. The adjustment is proportionally weighted, based how much of each pixel the line covers.

Anti-Aliasing by Area Sampling



The Area Sampling method visually smooths a diagonal line or edge by setting each pixel's intensity proportional to the amount it is covered by the object.

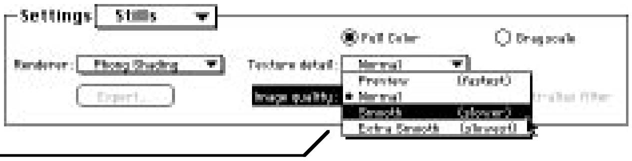
Pixel averaging uses only the existing pixels at the base resolution to perform its smoothing. Because of that, this method of smoothing images can produce subtle blurring that may cause loss of fine details

Each method has its advantages and disadvantages. In Vision3d, both methods are available, although Vision3d uses super sampling as its primary method of anti-aliasing.

Using the super sampling method

Because super sampling works its smoothing magic through extra resolution, it produces an image with superior sharpness along with the smoothness.

Super sampling anti-aliasing is used when the Image Quality field in the Render dialog is set to “Smooth” or “Extra Smooth.” The smooth setting generates two pixels vertically and two pixels horizontally for each pixel in the final image. The extra smooth setting renders three pixels both directions.



Settings: Stills

● Full Color ○ Grayscale

Renderer: Phong Shading

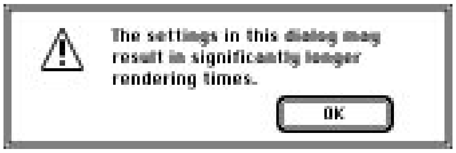
Texture detail: Normal

Image quality:
- Preview (fastest)
- • Normal
- Smooth (slower)
- Extra Smooth (slowest)

Choosing “Smooth” or “Extra Smooth in the Image Quality menu turns on anti-aliasing

Thus, the intermediate image is four times larger for the smooth setting, and nine times larger for extra smooth. The excess data is then interpolated down to the final image specified by the Frame size and d.p.i. settings in the same Render dialog.

If you render a 72 d.p.i. image with the Smooth image quality setting, the intermediate image is effectively 144 d.p.i., but it is re-sampled down to 72 d.p.i. for the final image. If you asked for a 150 d.p.i. image, with the smooth level of anti-aliasing the intermediate image would be 300 d.p.i. With the extra smooth level of anti-aliasing, the intermediate resolution would 450 d.p.i.



The settings in this dialog may result in significantly longer rendering times.

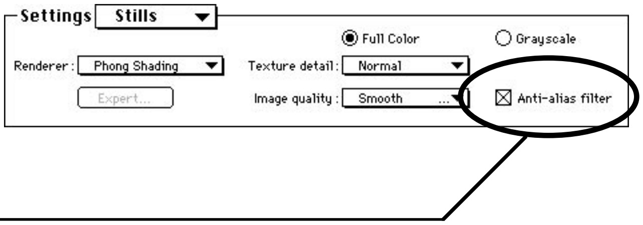
If the anti-aliasing factor is set to extra smooth, an alert is displayed warning of the longer rendering times required for the extra resolution.

This same alert can appear in other places throughout the program, whenever a setting is made that is considered atypical. It is only intended to make you aware of the significance of the choices you make.

Using the pixel averaging method

This method of anti-aliasing can produce visually pleasing images without the need for higher resolution and larger file sizes. Pixel averaging is especially useful in situations where extra resolution via super sampled pixels are not available, i.e. — existing images (rendered or otherwise) at low resolution.

A pixel averaging anti-alias filter becomes available when the Image Quality field is set to Smooth or Extra Smooth



Settings: Stills

● Full Color ○ Grayscale

Renderer: Phong Shading

Expert...

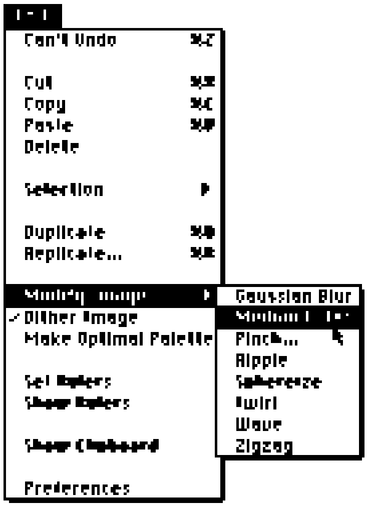
Texture detail: Normal

Image quality: Smooth ...

☒ Anti-alias filter

During rendering in Vision3d, pixel averaging anti-aliasing can be added to the super sampling to obtain a vastly superior image. This is enabled with the checkbox to the right of the Image Quality menu.

Note that this feature is only available when the Image Quality pull-down menu is set to Smooth or Extra Smooth. This is an image quality consideration. If Vision3d's anti-aliasing filter is used alone — on a simple, non-anti-aliased image — the image is too blurry. But when used in conjunction with the supper sampling, the resulting image is incredibly good.



Pixel averaging anti-aliasing can also be supported in Vision3d as a post-rendering filter or extension. In that manner, images from any source can be opened in the program and anti-aliased using this method. Vision3d includes two built-in filters for this purpose: 1) Gaussian Blur and 2) Median Filter. Other anti-aliasing filters may become commercially available from Strata Incorporated or third-party sources.

Additionally, you may use one or more “specialty” filters to further modify images for various purposes. These are also accessed through the Modify Image command in the Edit menu. All filters work on the entire image. There is no provision to select limited areas of an image in Vision3d. That capability is available in other applications, if needed.

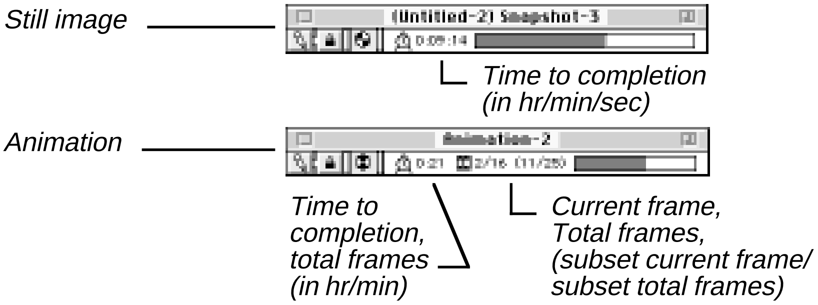
Specialty filters may also be obtained as extensions. For example, Vision3d can accept third-party plug-in filters marketed for use with PhotoShop. To install, copy them into the Extensions folder residing in the Vision3d application folder. Under System 7, you can put aliases of the plug-ins in the Extensions folder and leave the originals in the PhotoShop folder; thereby avoiding using excessive hard disk space to store duplicate files in two places.

Extension image filters appear in the Modify Image sub-menu, when present.

Rendering windows

Rendering windows are discussed in chapters 1 and 7. They are mentioned here again in the interest of the information provided at the top of a rendering window while it is in the process of rendering an image.

Feedback found in rendering windows:

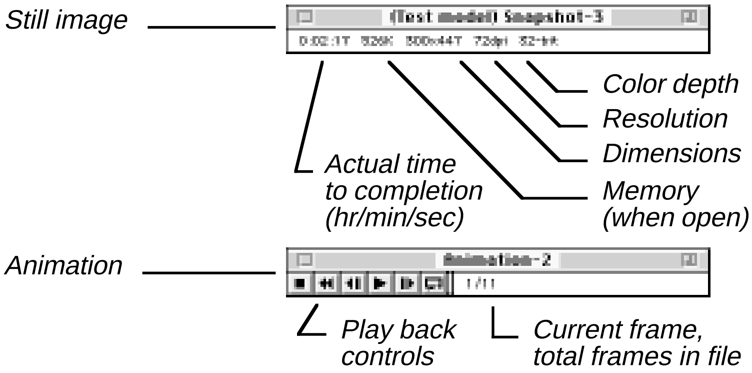

Still image
(Untitled-2) Snapshot-3
0:09:14
Time to completion (in hr/min/sec)
Animation
Animation-2
0:21
2/16 (11/25)
Time to completion, total frames (in hr/min)
Current frame, Total frames, (subset current frame/ subset total frames)

When a rendering is begun, a sequence of events occurs internally to allocate memory for the rendering. One of the events is to open a rendering window from which you can monitor the progress of the rendering. Several pieces of information are provided in the feedback area of a rendering window. That information varies slightly, depending on whether the rendering is a still image or part of an animation sequence. The differences are itemized in the illustration on the previous page.

Image Display Windows

Image windows are also discussed in chapter 1 and 7. They are mentioned here again to draw your attention to the information and/or controls provided at the top of the window. The content differs with the type of file, as shown in the following illustration.

Feedback found in image windows:

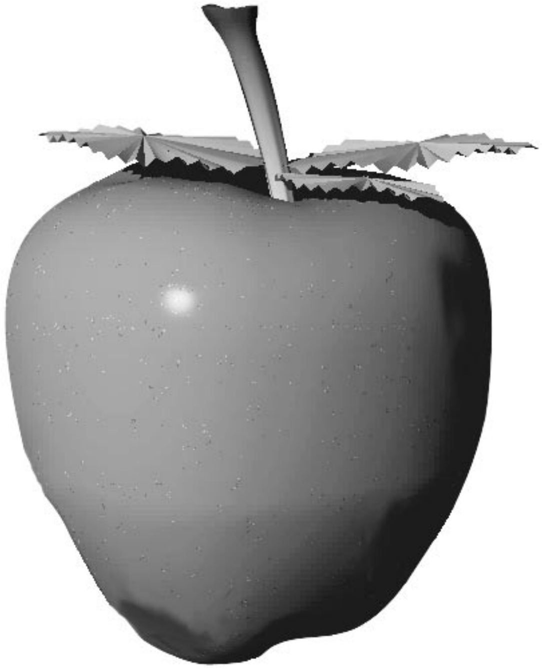


When a rendering is completed, the rendering window automatically converts to one of the image windows shown here. If the rendering was begun in the current session from a suspended rendering, the image is saved automatically. Otherwise, Vision3d waits for you to manually save it. In either case, the image window remains open until you close it.

Of course, one of the image windows is also used when you open an image or animation that has been saved previously. Images and animations from any source can be opened, not just files saved from Vision3d, as long as they are in a format recognized by the software.

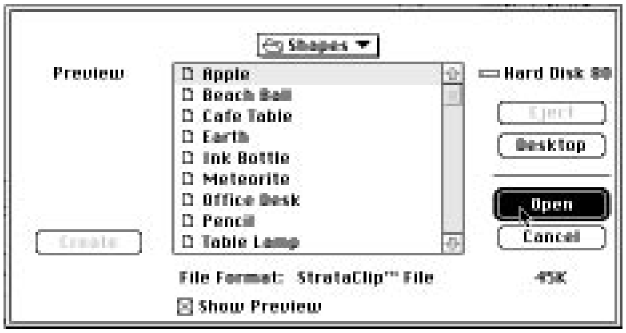
Quick Tutorial

The purpose of this quick tutorial is to acquaint you with some of the rendering methods in Vision3d. You will use a model, titled “Apple,” to render the same object using different rendering methods in order to compare and contrast the results. You will examine the different renderings in terms of appearance, image detail, and length of time required to complete the renderings.



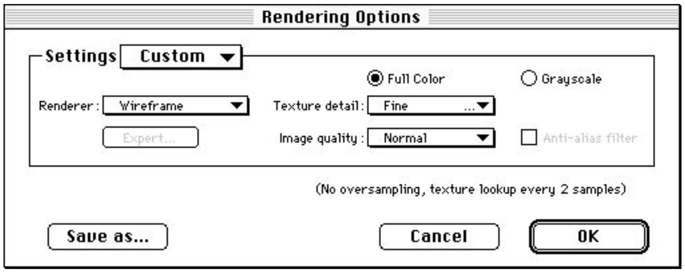
The model is that of a shiny red apple. Notice that this apple is not just a simple symmetrically-lathed object, but a realistic bézier construction of an apple. The apple actually started out as a sphere primitive that was “molded” in 3d Sculpter for a more apple-like appearance. The leaves were also made with the 3d Sculpter modeler, and the apple surface is made from a solid texture. Despite all of the explicit geometry, the modeling is actually quite efficient.

1. **In the Vision3d Shapes folder, find the shape titled “Apple” and open it.** Pull down the File menu and choose the “Open” command. Then navigate to the Tutorial folder. Find and open the Apple file.



When the file opens, you should be in the a custom view with three-position lens control set on Normal and the display method set to wireframe. Use the "Fit Views to All" command if necessary. The custom view is just 18° up from the Front view to enhance the perspective of the apple when it is rendered.

2. **Pull down the Rendering menu and choose the Rendering Options command.** Vision3d opens the Rendering Options dialog. You can also double-click on the Camera tool to open the Rendering Options dialog.
3. **In the Rendering Options dialog, set the Renderer pull-down menu to Wireframe.** Make sure the Image Quality pull-down menu is set to Normal. Then click on the OK button. For now, you will ignore the Texture Detail field because Wireframe rendering does not support textures.



Rendering Options

Settings: Custom

● Full Color ○ Grayscale

Renderer: Wireframe

Texture detail: Fine …

Image quality: Normal

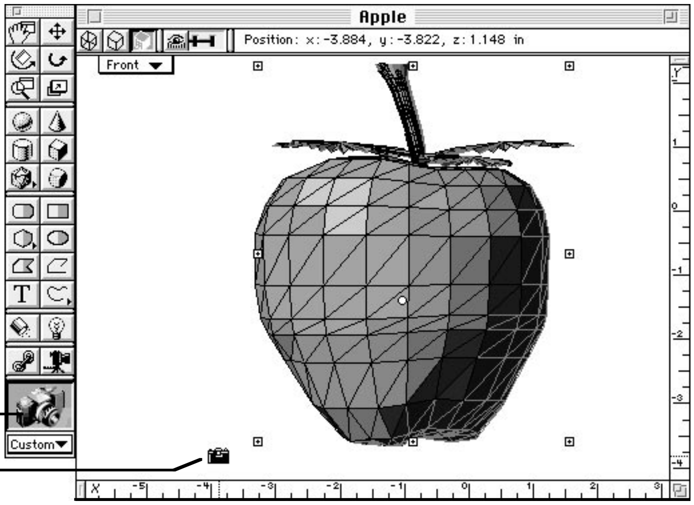
□ Anti-alias filter

Expert…

(No oversampling, texture lookup every 2 samples)

Save as… Cancel OK

4. **Select the large camera snapshot tool at the bottom of the tool palette.** Notice how the icon changes to a little camera when you move the cursor into the model window when this tool selected.



Apple

Position: x: -3.884, y: -3.822, z: 1.148 in

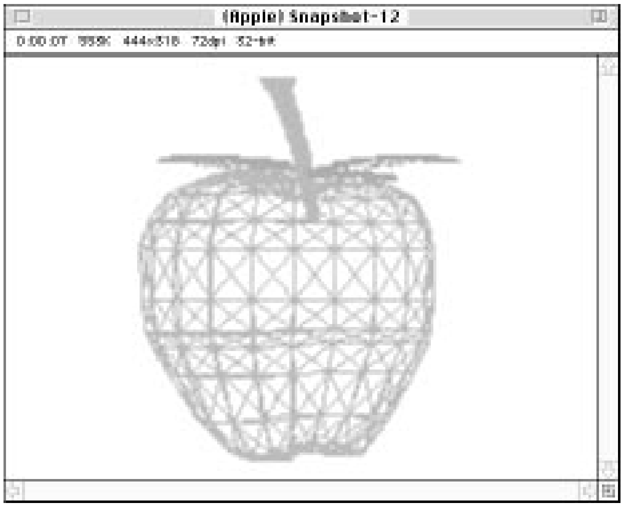
Front

Custom

When the snapshot tool is chosen …

…the cursor icon changes to a camera when it's inside the model window

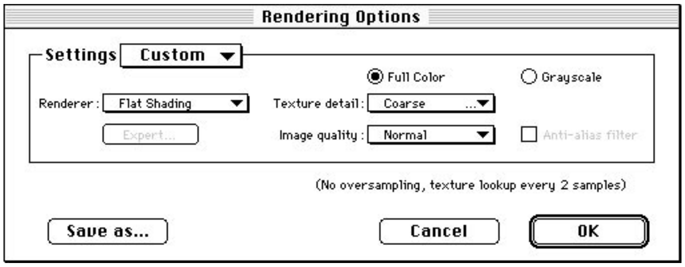
5. **Click once in the model window.** Vision3d begins to render the apple. Since this is a very simple rendering, it will finish quickly. The finished rendering should look similar to the following illustration.



6. **Save the rendering.** Name it “Apple-Wireframe.” Choose the PICT file format. You’re saving this rendering so that when you finish this exercise, you can open all your apple renderings and compare them side-by-side to see the differences between the various rendering methods.

   Once the image is saved, close the image window.

7. **With the model window active again, open the Rendering Options dialog again.** This time choose Flat Shading from the Renderer pull-down menu. Image Quality should be on Normal. Once again, you can ignore Texture Detail settings. Click on the OK button.



Rendering Options

Settings: Custom

● Full Color ○ Grayscale

Renderer: Flat Shading

Texture detail: Coarse

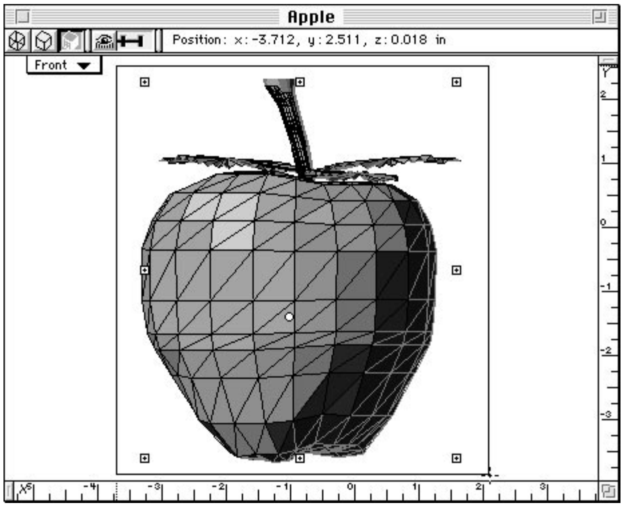
Image quality: Normal

□ Anti-alias filter

(No oversampling, texture lookup every 2 samples)

8. **Choose the snapshot tool again, then in the model window click-and-drag a marquee around the apple as illustrated.** When you

release the mouse button, Vision3d begins to render the apple using the Flat Shading rendering method.



Position: x: -3.712, y: 2.511, z: 0.018 in

The click-drag method of initiating a rendering allows you to render only that portion of the model window you wish to have rendered. By contrast, the single click method renders the entire model window.

Notice that the Flat Shading rendering method provides more detail than the Wireframe method. The apple has a surface in this rendering, not just a frame outline as with the wireframe rendering method. You will also notice it takes slightly longer for a Flat Shading rendering to finish than for a wireframe rendering, although the time is still very short. That's the rendering trade-off: the more detail in a rendering, the more calculations required to create the image, and the longer it takes for Vision3d to complete the rendering task.

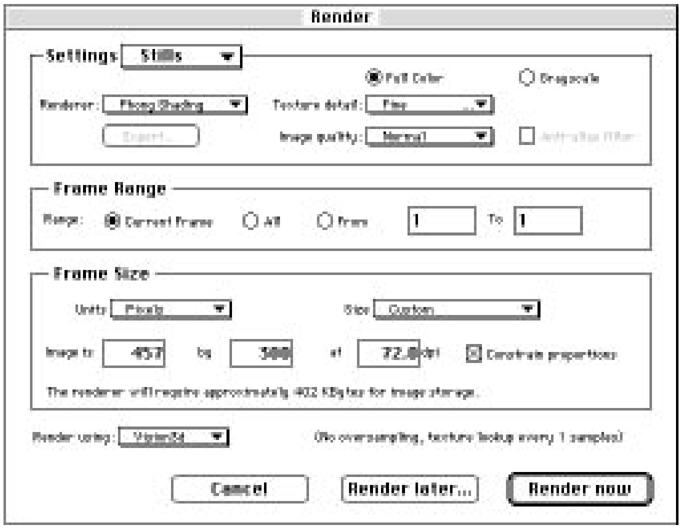
9. **When the rendering finishes, save it to your hard drive.** Call it "Apple-Flat Shaded."

10. **Open the Rendering Options dialog again.** Set the Renderer to Gouraud Shading. (Gouraud is named after the man who invented this method of rendering 3-D objects.) Make sure the Image Quality is on Normal. Also, this time set the Texture detail field to Normal. Then click on the OK button.

11. **Render the apple using the Gouraud method.** Use either the single click or the click-drag method to start the rendering. When the rendering finishes, again notice that the Gouraud rendering method provides even more detail than the previous rendering methods.

The apple has a smooth surface in this rendering, instead of the faceted surface you see in the Flat Shading rendering. This rendering also took slightly longer to complete than the Flat Shaded rendering. Save the rendering to your hard drive. Name it "Apple-Gouraud."

12. **Time for a change: Hold down the Shift key while you click once in the window.** The Render dialog appears. Notice how the top part of this dialog is very similar to the Render Options dialog.



Render

Settings: Stills

● Full Color ○ Grayscale

Renderer: Phong Shading

Texture detail: Fine

Image quality: Normal

□ Antialias filter

Frame Range

Range: ● Current Frame ○ All ○ From 1 To 1

Frame Size

Units: Pixels

Size: Custom

Image is: 457 by 300 at 72.0 dpi ☒ Constrain proportions

The renderer will require approximately 402 KBytes for image storage.

Render using: Vision3d

(No oversampling, texture lookup every 1 samples)

This time set the Renderer to Phong Shading. (Like Gouraud, Phong is named after the man who invented *this* method of rendering.) Image Quality should still be Normal, but now set the Texture Detail field on Fine.

Because the Phong renderer is the lowest rendering level that supports textures, you now can exercise control over the rendered quality of textures independent of the renderer itself. As the different texture detail settings explain, the less texture detail you render, the faster the rendering will finish. The same is true of the Image Quality settings: the higher the image quality setting, the longer the rendering will take to complete.

13. **Click on the Render Now button.** When the rendering finishes, again the rendering trade-off is quite noticeable. The completed rendering looks more realistic than ever, but took longer to finish. Save the finished rendering to your hard drive. Call it "Apple-Phong."

14. **Time for another change: Click on the pull-down menu at the bottom of the Tool palette.** Choose Effects, then select Watercolor from the sub-menu.

15. **Render the apple using the RayPainting rendering method.** This rendering method is a very interesting variation from the other rendering methods. This rendering algorithm is designed to simulated artistic effects rather than photo-realism. Save the finished rendering to your hard drive; call it "Apple-RayPaint 1."

16. **Repeat steps 14 and 15.** This time, choose Pencil on Rag from the Effects sub-menu. This produces a completely different look than the watercolor pre-set. Save this image as "Apple-RayPaint 2."

17. **Click on the Pull-down menu on the Tool palette.** Choose Stills, then select Better from the sub-menu.

18. **Render the apple using the Raytracing rendering method.** This rendering method takes longest to complete of all you will use in this exercise, and as you now know, it's also the most realistic. Save the finished rendering to your hard drive; call it "Apple-Raytracing."

Now open all the saved renderings using Vision3d or another application and compare them. Each rendering method you tried in this exercise has provided progressively more detail, and thus a better, more real-looking, image than the previous method. RayPainting is an interesting and useful variation from the other rendering methods.

The time trade-off was also clear. The wireframe rendering finished fastest. The raytracing took the longest. However, raytracing is also the most real-looking of the rendering methods. From this you've learned about the rendering trade-off: The more realistic the image, the longer it takes to finish rendering.

You can see it's wise to render an image only with as much detail and realism as needed to accomplish the project. It's not necessary to raytrace every image — using high quality and fine detail settings for everything. In many cases, the other rendering methods will produce a sufficiently realistic image, while finishing the rendering task more quickly.

Choosing the appropriate rendering method

The rendering process involves giving objects shading. The finished image is composed of pixels, each with a calculated color derived from the interaction of object surfaces with light sources inserted in the model. Each pixel in the rendered image requires four bytes (32 bits) to define if the image is in color, or one byte (8 bits) if only gray scale is used. Note that there is no speed improvement by rendering in grayscale, but it will require less memory during rendering, as well as after it is saved.

Once rendered, an image may be saved in various color depths and formats for further use in other applications, or it may be output directly from Vision3d to obtain a hard copy.

Vision3d contains nine different rendering algorithms in its repertoire to provide images of the model with different degrees of realism, and for various purposes. The extended choice allows you to balance the image quality requirements against the time required to generate it. If a lesser renderer can accomplish the job in acceptable quality, why waste time on an image that takes longer to render?

The renderers provided in Vision3d are: 1) wireframe, 2) solid (wireframe with hidden lines removed), 3) quick shaded, 4) flat shaded, 5) Gouraud, 6) Phong, 7) RayPainting, 8) Raytracing and 9) Raydiosity. While the lower-level renderers — wireframe, solid, and quick shaded — are used primarily for displaying objects in the modeling views, they are also considered renderers in their own right. Images in the modeling windows created by the wireframe, solid or quick shaded renderers may be saved as QuickDraw PICT files containing draw-type commands for the image. The higher-level algorithms use the polygons' normal vectors to properly calculate the shading of surfaces and produce a pixel-based image in the resolution specified.

As described elsewhere in this manual, a "normal" is a vector perpendicular to each polygon which indicates the orientation of the surface. Knowing which way a polygon is facing is very important for the renderers. The higher-level renderers use the normal vectors from either the polygon or the vertices of the polygon to produce smooth-shaded surfaces so essential for imaging realism.

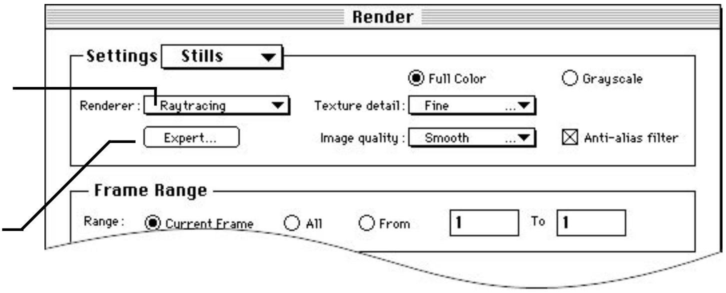
How raytracing works

More accurately termed "backward ray tracing," the raytracing algorithm operates by following each ray of light striking a rendering's view plane (defined by the active window) backward through the model to their source(s). If a ray comes from light reflecting off an object's surface, the texture applied to that object must be consulted to determine what that reflected light should look like. That may include surface maps, reflectivity, shininess, or many other lighting factors. If the ray of light is coming from another object that is reflecting in the first object's surface, then the ray must be pursued further backward from this surface to determine whether it came from another surface or from a light source.

This backward tracing continues until each light ray is accounted for; either ending at a light source or passing out of the scene. The minimum number of rays that must be traced is equal to the number of pixels in the image being rendered, plus any rays spawned to handle reflective surfaces, transparent surfaces and shadows. Thus, the larger the image and the higher its resolution, the longer the rendering time. A 288 d.p.i. image, for example, has four times as many rays to trace as a 72 d.p.i. image of the same dimensions.

Changing raytracing parameters

When you choose to use the Raytracer (by choosing it from the "Renderer" pull-down menu), the "Expert" button becomes active just below the Renderer field. This button allows you to open the Raytracing Esoterica dialog box. Operating settings for the Raytracer are displayed in this dialog box.



Render

Settings: Stills

● Full Color ○ Grayscale

Renderer: Raytracing

Texture detail: Fine

Expert...

Image quality: Smooth

☒ Anti-alias filter

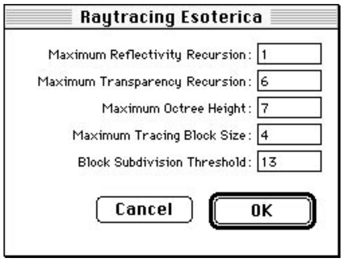
Frame Range

Range: ● Current Frame ○ All ○ From 1 To 1

Choosing the Raytracing renderer

…allows access to the "Raytracing Esoterica" dialog through this button

The default values (that are present in this dialog box when Vision3d is first installed) are settings designed by engineers at Strata Inc. to deliver the most efficient, best all-around performance from the raytracing rendering algorithm. The fields have been made accessible to allow for customizing the raytracer as needed for special imaging requirements. Before changing these settings, however, you should thoroughly understand how these fields control the algorithm. Otherwise, you may adversely affect the speed and efficiency of the raytracer.



Raytracing Esoterica

Maximum Reflectivity Recursion: 1

Maximum Transparency Recursion: 6

Maximum Octree Height: 7

Maximum Tracing Block Size: 4

Block Subdivision Threshold: 13

Cancel OK

If you do change the settings in this dialog, then want to return to the factory settings, just choose one of the pre-set rendering options from the Settings pull-down menu in the Render dialog. The values in this dialog will revert to those shipped with the software.

Intelligent rendering

To make the raytracing process more efficient, Vision3d assigns priorities in the method it goes about searching for the sources of light rays. This prevents or limits chasing after rays of light that do not materially contribute to the image.

One method used is to divide the model space into smaller volume elements, or "voxels." The model space is first sub-divided along the three axes, defining eight smaller cubes of the space. Each of those cubes may then be subdivided into another eight smaller cubes, depending on whether there are objects found in them. No objects, no further sub-division and no more time spent there.

Not all cubes are subdivided the same amount. The actual number of times one of the original eight cubes is subdivided ultimately depends on the number of objects in it, but is also controlled by a value in the Raytracing Esoterica dialog box (explained later). The various levels of subdivision is called an octree structure. "Octal" means eight, which is the number of smaller cubes created each time a larger cube is subdivided on each axis, and a schematic representation of the subdivision process takes on a tree-like appearance.

The subdivision may be repeated as often as needed to define small enough volume elements (with few enough objects in each voxel) that the Raytracer can interrogate them quickly and efficiently. By comparing the location of objects in the model to the voxels in each octree level, the algorithm can know where to spend time tracing rays of light and which areas it can ignore.

Field descriptions

The following descriptions of the fields in the Raytracing Esoterica dialog is not intended as a complete explanation of how Vision3d's raytracing algorithm works. Entire textbooks have been written on the concept of raytracing. Some of these are listed at the back of this manual in the Bibliography of the Appendices section. Of course, Vision3d's raytracer is proprietary, so it has been enhanced and may operate just a little differently than the public domain versions.

This discussion is meant only as a guide to help your own judgment in adjusting raytracing parameters for special rendering situations. You should realize that the optimal values for the raytracer can vary from project to project, depending on model size, available memory, the type of surface maps in the textures used, etc. There are no wrong settings, just degrees of rightness. Obviously, anything that slows down rendering speed without significantly improving image quality is less "right" than a configuration that gains performance while maintaining the excellent image quality for which Vision3d is famous. Arriving at the optimal raytracing values is often a matter of compromise.

Of course, you are encouraged to experiment with the numbers in this dialog. Do so with the intent of obtaining faster rendering or better quality (or both), once you understand what to look for. Understand that these two goals are usually mutually exclusive. You should develop a methodical approach to any changes you make and do some benchmark tests to compare relative performance.

The fields in the Raytracing Esoterica dialog and their effect on rendering are:

- **Maximum Reflectivity Recursion.** This field determines how far a ray of light is traced as it reflects from surface to surface. In visual terms, it determines how many times an object will be reflected in the surfaces of other objects. The default setting is one. For most models this is adequate. Increasing this setting usually adds to the rendering time. But if you are modeling a "House of Mirrors," however, you will want to set this value higher.
- **Maximum Transparency Recursion.** This field's setting only affects models with multiple transparent objects positioned in front of each other. It determines how many layers of transparent surfaces the Raytracer will look through to see if there is something behind before giving up. If more transparency layers exist than this setting allows the raytracer to look through, that area of the image will appear black. The default setting is six.
- **Maximum Octree Height:.** As discussed earlier, Vision3d's Raytracer sets up several octree levels during the rendering operation. An Octree is an internal data structure used to subdivide the model space for more efficient handling and to simplify the ray tracing process. Initially, all objects are contained in one large cube or model space. The model space is adaptively subdivided by a process where each cube on one level yields eight smaller cubes on the next level. The actual number of octree levels created for a particular rendering depends on the complexity of the model.

  Theoretically, the higher the number, the faster the raytracer will run. There is a point, however, when the octree structure become too big and it begins to take more and more time to maintain; negating any gains in its efficiency.

This field determines the maximum number of levels in the octree to be defined for a rendering operation. The higher this value, the larger the memory partition for Vision3d needs to be. The octree structure must be set up within the application's memory partition. Higher numbers divide the model space more finely, but take longer to set up and sort through. A lower value uses less memory, but may decrease the Raytracer's efficiency (meaning less speed).

- **Maximum Tracing Block Size.** In the creation of an image, the renderer starts with large blocks of pixels which are refined smaller and smaller during the rendering process. The refinement continues on every pixel block until the maximum block size is reached. At that point, the Block Subdivision Threshold setting decides which blocks to refine smaller. A setting of four in this field means a four-pixel by four-pixel block.

  Larger block sizes render much faster, but may sometimes cause the Raytracer to miss fine lines. If your model has lots of detail work — either as surface maps or as explicit object geometry — it is often desirable to set this field smaller. In models that do not have fine details, a larger block size is usually preferred.

  Setting the maximum block size smaller will not only ensure picking up the fine details in an image, but may actually render faster in those cases where it is needed. That's because starting with smaller block sizes at the start of a rendering doesn't waste time later having to go back to find the detail that was missed, if it is found at all.

  A tracing block size that is too large for the situation usually shows up as disappearing holes in 3-D text or the one-pixel lines in surface maps not showing up consistently. This becomes especially obvious in animations, where the detail is present in one frame, but not the next.

- **Block Subdivision Threshold (percent):** When the maximum tracing block size is reached, the Raytracer scans the image block-by-block to see if it needs to be refined further. It usually does. If two adjacent blocks of pixels have a difference in color values, this field decides the variation threshold — when it is significant enough for the Raytracer to pause and further refine the two blocks to resolve the difference. It will continue this process until all adjacent blocks (pixels) are within the allowed percent of variation.

  Setting the value higher increases rendering speed, but may cause noticeable color banding of certain colors on curved, shaded surfaces. Setting the value lower forces all block sizes to be resolved finer, even if banding is not visible.

Raytracing is the glamour algorithm in the Vision3d line up. There is a tendency to want to use raytracing for all rendering because it looks so good! But many of those extra effects are also available with the Phong algorithm. In many scenes set up for Phong rendering with an environment map, the result cannot be distinguished from Raytracing.

Effects like object reflectivity and refraction, come at a computational cost in Raytracing. Not every image needs to be raytraced; not every image is worth the investment in time. If tempted to jump right to the raytracing-required rendering options, you should stop and analyze the requirements of the image. Refraction and shadows are the only effects not supported in some manner in one of the other (faster) algorithms. If refraction and/or shadows will not make a major contribution to the finished image, perhaps Phong or one of the other renderers would do as the job as well, and a lot quicker.

How RayPainting™ works

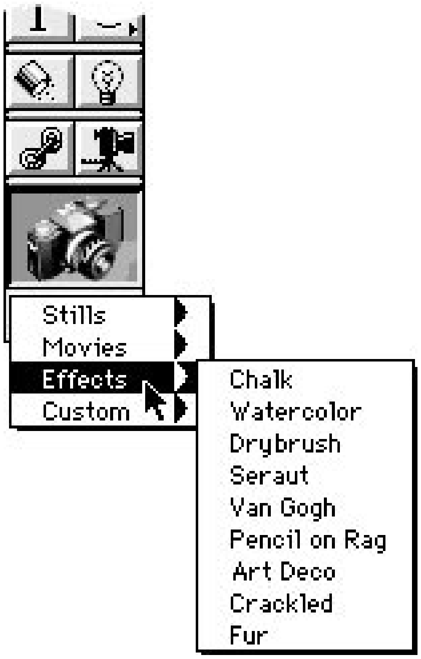
RayPainting is a diversion from the stereotypical 3-D rendering algorithm. While it uses the same basic rendering method as the raytracer, it also adds some very special and dynamic artistic effects that put creativity back into the rendering process. One thing you will notice right away is how much faster it is than raytracing. It is often faster than the Phong renderer. This is especially true with higher resolution renderings.

More than any of the other renderers, the RayPainter begs to be explored. You'll be amazed at how many different effects you can come up with as you try variations of the different settings that control the RayPainter. As you do, you will notice that this renderer is more than just a fancy image filter. It is a true 3-D brush whose strokes follow the geometry and contours of objects in ways that no image filter can begin to emulate.

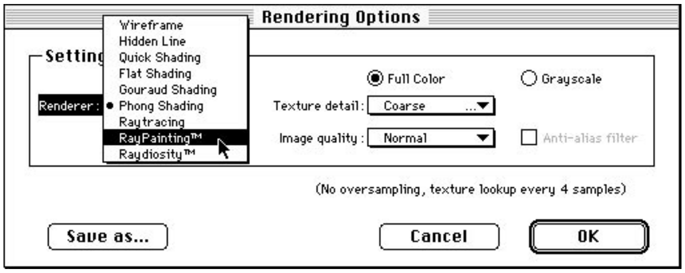
You do not set the RayPainting rendering parameters by the numbers, as you would with the raytracer. RayPainting is concerned more with artistic effects than realistic detail. RayPainting rendering settings are made by choosing items like canvas type, brush type, stroke gap, "jitter," etc.

To use the RayPainter, you can begin by trying the several pre-sets available through the pull-down menu on the Tool Palette.

The entire Effects sub-menu uses the RayPainting renderer to bring artistic creativity to 3-D rendering



To go beyond those effects built-in to Vision3d, just click on the Tool palette's pull-down menu, the choose the "Custom" entry in the Custom sub-menu (Custom•Custom) …or choose the Rendering Options command in the Rendering menu. Either method opens the Rendering Options dialog box.


Rendering Options

Renderer: RayPainting™

● Full Color ○ Grayscale

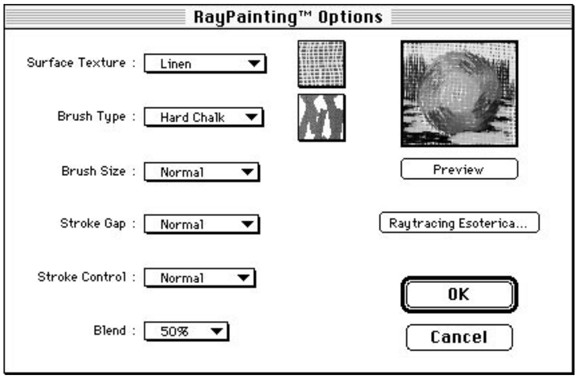
Texture detail: Coarse

Image quality: Normal

□ Anti-alias filter

(No oversampling, texture lookup every 4 samples)

Choosing RayPainting from the Renderer pull-down menu, makes the Options button just below this field active. Clicking on the Options button brings up the RayPainting Options dialog.



RayPainting™ Options

Surface Texture : Linen

Brush Type : Hard Chalk

Brush Size : Normal

Stroke Gap : Normal

Stroke Control : Normal

Blend : 50%

This is where you can experiment with the RayPainting parameters and try them on the preview model right there in the options dialog. The preview is a special model built-in the RayPainting code to show the various effects you choose to best advantage. While you are learning the impact each field has on the RayPainting effect, it is best to develop a methodical approach; changing one thing at a time and see what effect it has on the preview image.

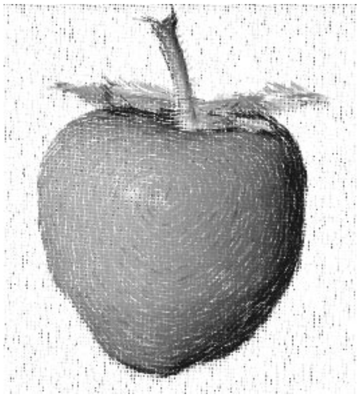


Image rendered with RayPainting

The canvas was set to linen; the brush type to hard chalk. All other field were left at normal. The blend was set to 50%.

When you find a combination of RayPainting settings you want to keep, you can click on the OK button to exit this dialog and save it by name in the Rendering Options dialog, using the Save As button. A collection of rendering option settings saved in this manner appear in the rendering options pull-down menu under the “Custom” sub-menu.

RayPainting Esoterica

The RayPainting Options dialog also contains a button titled Raytracing Esoterica. Clicking on this button displays the same Esoterica dialog you have seen in the discussion on Raytracing. But notice that the numbers in the fields are different. They are definitely “looser.” This is consistent with the type of images the RayPainter produces. As you know, the RayPainting renderer uses the raytracing algorithms to do many of its calculations. Therefore, the raytracing settings in this dialog also apply to the RayPainter.

NOTE: *The numbers in the Raytracing Esoterica dialog are changed to RayPainting specifications when you select one of the Effects pre-sets from the pull-down menu on the Tool palette. If you then plan to use the Raytracer again, IT IS IMPORTANT that you set the Esoterica fields back to those used for raytracing. The easiest way to do this is to choose “Stills•Better” or “Stills•Best” from the same pull-down menu. If you fail to do this, the raytracings you may do later will be noticeably less detailed than you might expect.*

As mentioned at the first of this discussion about RayPainting, the numbers in the Esoterica dialog are not as important as the settings in the RayPainting Options dialog. These are the controls that give RayPainting its unique look.

How Raydiosity™ works

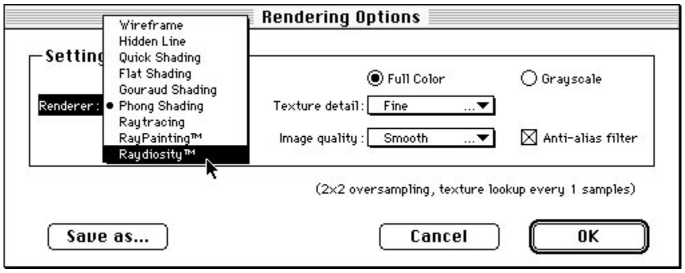
Beyond Raytracing and RayPainting is…Raydiosity! Raydiosity can calculate rendering effects even the venerable raytracer cannot do. Effects like the illumination of translucent objects and bleeding colors from one brightly lit object onto the surface of another. Effects like bouncing illumination from a light source off a white wall to provide softened fill lighting on the back side of an object.

Raydiosity calculates the illumination contributions by reflected light from nearby objects. This is called “inter-object diffuse illumination.” This is an advanced aspect of photo-realism that raytracing does not (and cannot) consider in creating its images.

Raydiosity performs its work by vastly increasing the number of ray tracing samples taken of a model over that of the raytracer. By virtue of the increased samples, the renderer can more accurately determine how the objects in a scene interact with each other, i.e. — how does the light reflecting from the many surfaces in a scene affect the overall illumination of all the objects?

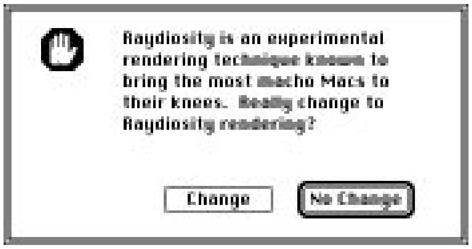
Because of the time required to perform these calculations on the Macintosh platform, this algorithm is rarely usable in day-to-day production work. Strata Inc. offers Raydiosity in Vision3d as a semi-experimental rendering method. Have fun with it; see how it improves the realism of a scene; try different lighting effects. Just be sure you have enough quarters (time) to “feed the meter.”

NOTE: *The spelling variation from the accepted term of "radiosity" is intentional. This is because Raydiosity is the name of a proprietary algorithm Vision3d uses that is substantially different than the radiosity developed at Cornell University. Vision3d's Raydiosity rendering algorithm is view dependent, but renders much faster for single images. The finished Raydiosity image, however, looks remarkably similar to that achieved on high-end platforms using radiosity.*



(2×2 oversampling, texture lookup every 1 samples)

The Raydiosity rendering method is chosen through the Renderer pull-down menu found either in the Rendering Options or Render dialogs. None of the Settings pre-sets choose Raydiosity as it rendering method. That's because it is so intensive and takes a long time. In fact, each time Raydiosity is chosen as the rendering method, an alert appears asking if you really intended to do this.

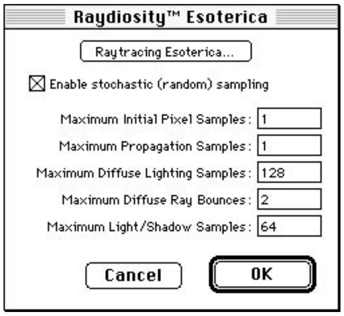


Raydiosity is an experimental rendering technique known to bring the most macho Macs to their knees. Really change to Raydiosity rendering?

Raydiosity is a very expensive rendering method in terms of the number of calculations required. The mechanism for interrogating the scene to be rendered still involves tracing rays of light backward through the model. If fact, the Raydiosity renderer uses the raytracer to do much of its work. However, where raytracing uses a finite number of rays limited to the viewing position of the rendering, Raydiosity is also capable of tracing rays that do not directly impact the view plane.

Raydiosity Esoterica

When Raydiosity is actually chosen as the rendering method (and you proceed with changing the rendering method when the alert appears), an Expert button becomes active just below the Renderer field in the dialog. Clicking on this button opens the Raydiosity Esoterica dialog box. This is where the parameters for running the Raydiosity renderer are set. Note that this second dialog also has a button to open the Raytracing Esoterica dialog. Since the Raydiosity renderer use the raytracing renderer to perform much of its works, those settings will also impact this renderer.



☒ Enable stochastic (random) sampling

Maximum Initial Pixel Samples : 1

Maximum Propagation Samples : 1

Maximum Diffuse Lighting Samples : 128

Maximum Diffuse Ray Bounces : 2

Maximum Light/Shadow Samples : 64

In this discussion of the Raydiosity Esoterica fields, reference is made to primary and secondary surfaces. Every surface is a primary surface at some time during the rendering process. Which surface is primary and which is secondary (and third, fourth, fifth, etc.) depends on what ray of light the Raydiosity renderer is tracing at any given moment.

Essentially, each time a ray of light reflects from a surface, the Raydiosity algorithm generates a cone of rays from the point of reflection to gather as much information as possible about the light directly and indirectly influencing the surface there.

The number and type of rays in that secondary cone are set in the Raydiosity Esoterica dialog. Each of the rays in the secondary cone can reflect off another surface and generate a cone of third-level rays…and a fourth, fifth, etc. With the settings provided from the factory for the Raydiosity renderer, the actual number of rays traced in a complex scene can easily exceed the one million mark!

An explanation follows on what the Raydiosity Esoterica fields mean:

- **Stochastic Sampling:** This field turns random sampling on or off. With this field on, the pattern of light rays traced within each cone of rays is randomized. This typically produces an image with a more realistic appearance. Without random sampling, the pattern of rays in each cone of rays generated by a ray reflection are symmetrical and uniform; resulting in an image that is more obviously computer generated. However, if the number of rays traced at each cone level is not sufficient, using stochastic sampling can produce a mottled or splotchy appearance to surfaces.

- **Max. Initial Pixel Samples:** This field determines the number of light rays initially traced for each pixel in the rendered image. Each increment in the value of this field is exponential in its impact throughout the rest of the model. This field should be set with a value of one.
- **Max. Propagation Samples:** This field determines the maximum number of secondary rays that are generated when a primary ray strikes a reflective or transparent surface. The actual number of rays (not to exceed the number set in this field) is based on the roughness of the surface, as determined by the First and Second Smoothness fields in the texture applied to an object.

  A setting of one means just one secondary light ray is traced for every primary ray; a setting of two means two rays are traced for each reflected ray, etc.
- **Max. Diffuse Lighting Samples:** This field determines the maximum number of secondary rays generated when a primary ray strikes a diffuse surface. Again, the actual number depends on the roughness of the surface. This field determines how much other surfaces contribute to the appearance of a primary surface. This field is the setting that most significantly influences the amount of inter-object illumination or color bleed between adjacent objects.

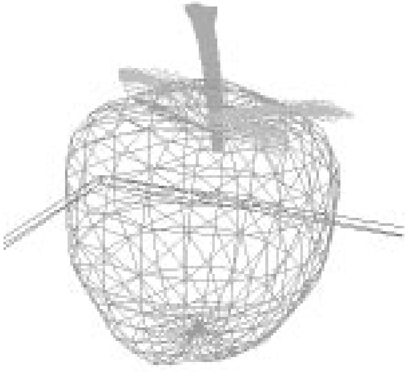
  A setting of one means just one secondary light ray is traced for every primary ray. The higher the value, the more adjacent (secondary) objects influence the illumination of the (primary) object, and the longer the rendering time.
- **Max. Diffuse Ray Bounces:** The field determines how far a secondary ray is traced during reflections between adjacent objects. If a secondary ray reflects or bounces off another object, it may also reflect off a third object, which, if close enough, can possibly also contribute to the primary object's surface illumination.

  A setting of one means the algorithm ignores the secondary light ray after the first bounce. The more bounces through which a ray is traced, the more the adjacent surfaces contribute to the primary surface's appearance, and the longer the rendering time.
- **Max. Light/Shadow Samples:** This field determines the number of rays traced to calculate the edges of shadows. It applies only to spot and point light sources. The size and shape of their icons are used as a pattern for the penumbra effect.

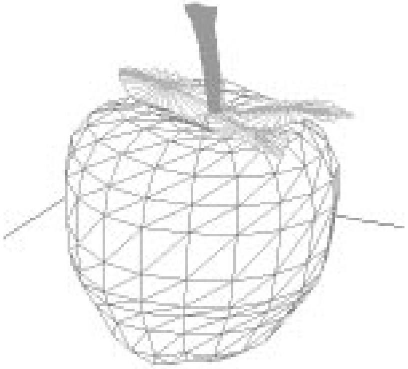
A setting of one means only one ray is traced for each pixel in a shadow area, which yields no softening of the shadows. Increasing the number generates correspondingly higher samples between the shadows and the light sources. Higher numbers produce a softer edge to the shadows, but again take longer to render.

Comparison of the rendering algorithms

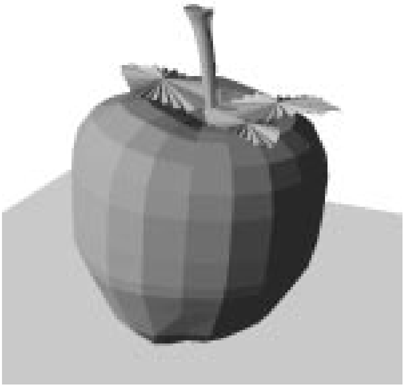
The following rendered examples and comparisons of features is provided to help make better choices of the most appropriate rendering method. A chart is also included after the individual discussions of each renderer.



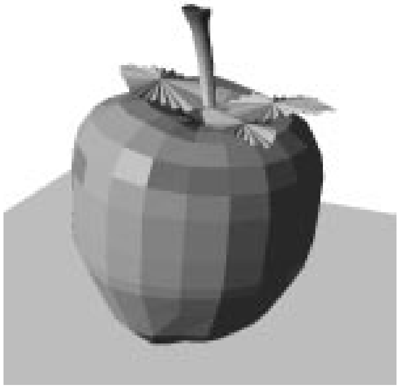
- **Wireframe.** This is the simplest rendering method used by Vision3d. It is primarily a display method for the modeling views, but may also be used for rendering separate images. It generates a line representation of object shapes in a model for displaying on the 2-D views of the screen or outputting to any QuickDraw-compatible printer. No surfaces, lighting, or surface shading are calculated or implied. In wireframe, surface-bearing polygons and hollow polygons are displayed the same way.



- **Hidden Line Removed.** This display method removes the hidden lines from wire frame representations before displaying them in the views or rendering in an image. Wire objects presented this way appear solid, with implied surfaces, but without illumination of the surface. Solid displays make it much easier to discern the orientation of objects.

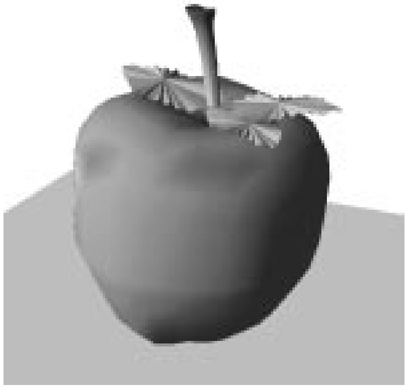


- **Quick Shading.** This display method calculates the actual surface orientation for each polygon. It also shows the illumination intensity for those polygons, providing rudimentary shading of the object(s). Shaded images require the most time to redraw in modeling views whenever changes are made in the model. This method is the highest quality image available in the modeling views. Like the previous two display methods, it can also be used to render images and/or provide QuickDraw-compatible draw-type PICTs that can be saved or printed.



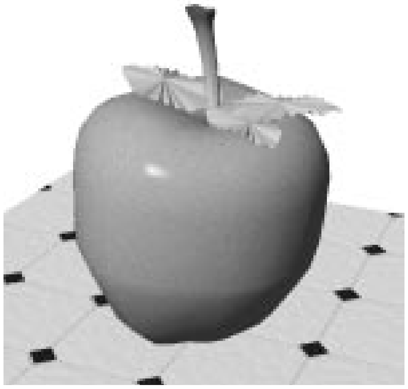
- **Flat Shading.** This rendering algorithm is sometimes called constant shading. It calculates a single color for each face on the surface. The orientation of a surface relative to the light source(s) illuminating it are also factored into the color as an intensity value.

  Flat shading is relatively fast. But, because it uses only one lighting calculation per polygon, the resulting rendering is obviously faceted in appearance. This renderer is most often used for preview-type images and animation development.



- **Gouraud Shading.** Gouraud is an intensity interpolation renderer. Its algorithm calculates the shading of a polygon by interpolating color intensity from the vertices of shared polygons (in triangulated mesh form), then averaging the shading across the surface of each polygon. Rather than using a single polygon normal, like the previous renderers, Gouraud shading uses the normals at each vertex. The resulting image is smoothly shaded, but can have artifacts at the polygon boundaries. The artifacts are most noticeable in areas around specular highlights on the object's surface.

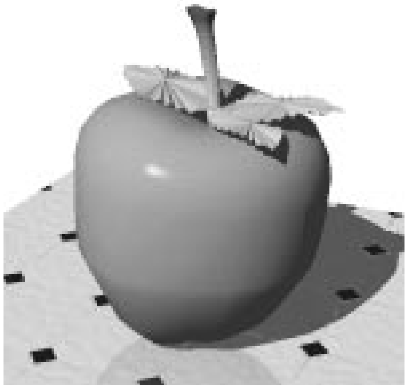
Only surfaces constructed as a triangulated mesh or as bézier, in which vertices are shared by adjacent polygons, have the requisite vertex normals. If no normals are available at the vertex points, the renderer cannot perform a smooth shading interpolation.



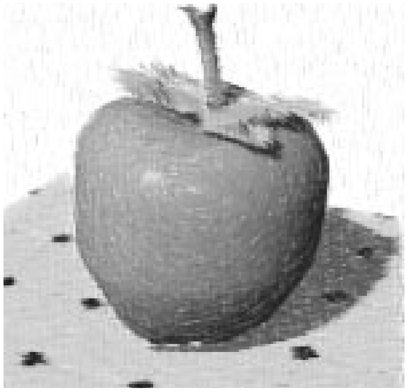
- **Phong Shading.** The Phong rendering algorithm is named for the person who developed this shading technique, Phong Bui Tuong. Phong is a normal vector interpolation shader. It evaluates the position of the vertices, with their associated normals, and interpolates a complete new set of normals across the entire surface of the object. These normals are then used to calculate an intensity for each pixel in the rendered image of that object.

  The ability to calculate each surface pixel's color also allows Phong to include surface mapping and environmental mapping in its rendering effects.

  Phong shading takes longer than the lower level algorithms, but produces a much better result. However, Vision3d's version of the Phong renderer is somewhat faster than traditional implementations because it uses a two-pass approach to the rendering operation. The first pass determines what surfaces are visible from the current camera position so the second pass doesn't waste time rendering hidden surfaces (only to cover them up with other surfaces). The Phong renderer has been further enhanced to support environment reflections on objects and more realistic specular highlights from metallic surfaces.



- **Ray tracing.** This renderer traces rays of light backward through a model to determine how the image should appear from the chosen viewing position. If the light ray comes from a surface, the renderer then calculates where the light that reflected off that surface originated. In some models, it is possible to trace light rays through multiple reflections until finding the source. Ray tracing brings true reflectivity, refractive transparency, and shadows to the imaging calculations. It is more computationally intensive than lower level algorithms, and thus takes longer. The results, however, can be stunning. Vision3d's ray tracer employs the same shading methods as Phong and supports all the surface mapping types.



- **RayPainting.** This algorithm is unique among the rendering methods available to you in Vision3d. It is intended as an artistic and special effects renderer, rather than a photo-realistic one like the others. It supports all of the rendering effects, including smooth shading, surface mapping, reflectivity, refraction, shadows, etc. But also allows you to set an surface (linen, rice paper, canvas, etc.), a brush type (hard chalk, crayon, fur, oil, etc.), a brush size, stroke gap, and more. The result is a rendering method that can let you be very creative. The images appear to have been drawn with 3-D brush strokes, which is exactly what it does.

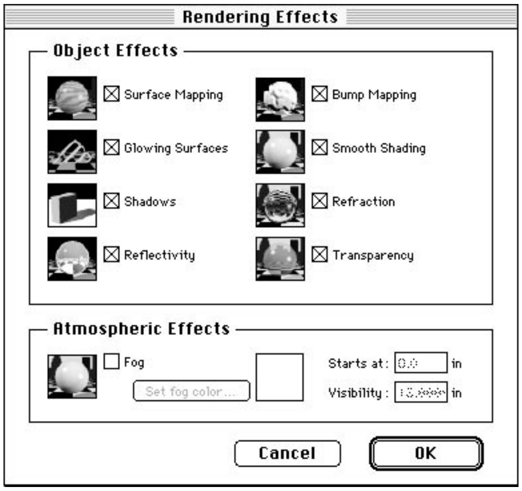
- **Raydiosity.** The photo-realistic effects of lighting and reflection in scenes can be very complicated and subtle. Indirect lighting is particularly difficult to render. This renderer calculates light intensity of each surface in a scene as a function of the intensity of all other surfaces. The effects are then included in the resulting image. This is called "inter-object diffuse illumination." Unlike Raytracing, which works best for highly reflective scenes, Raydiosity takes into account the distribution of light energy throughout the scene and is best suited for images containing mostly matte surfaces and indirect lighting.

  Raydiosity also supports soft edges on shadows from spot and point light sources. The penumbra pattern of shadows is determined by the size and shape of the light sources' icons.

All the renderers are integral with the modeling and lighting tools; making the rendering process a fully interactive tool of equal usefulness to the modeling process. In fact, rendering preview images of specific areas of a scene to adjust maps and lighting can be a very important aspect in the overall scene assembly.

Chart of features supported (by rendering method)

The effects to be included in a rendering are selected through the Rendering Effects dialog box. This dialog is opened by choosing the Rendering Effects command in the Rendering menu, or by double-clicking on the Camera tool in the Tool palette. The effects available are typically an on/off type of choice. Some choices, such as Fog, require additional parameters which may be set through fields in this dialog.



Rendering Effects

Object Effects

☒ Surface Mapping
☒ Bump Mapping
☒ Glowing Surfaces
☒ Smooth Shading
☒ Shadows
☒ Refraction
☒ Reflectivity
☒ Transparency

Atmospheric Effects

☐ Fog

Starts at: 0.0 in

Visibility : 10.0000 in

Note that at the beginning of each session in Vision3d, all of the rendering effects are turned on except Fog. You must open the Rendering Effects dialog to turn off the effects you do not want or need to use, which can save rendering time, or to turn on Fog. If you disable one or more rendering effects during a session and quit with them off, they are on again at the beginning of the next session.

Wireframe	Solid	Quick shaded	Flat shaded	Gouraud	Phong	Raytracing	RayPainting™	Raydiosity™
• Draw PICT • Anti-aliasing • Glowing surfaces	• Draw PICT • Anti-aliasing • Glowing surfaces	• Draw PICT • Anti-aliasing • Glowing surfaces	• Anti-aliasing • Glowing surfaces • Fog	• Anti-aliasing • Glowing surfaces • Fog • Smooth shading	• Anti-aliasing • Glowing surfaces • Fog • Smooth shading • Surface mapping • Reflectivity • Transparent surfaces • Shadows	• Anti-aliasing • Glowing surfaces • Fog • Smooth shading • Surface mapping • Reflectivity • Transparent surfaces • Shadows • Refraction	• Anti-aliasing • Glowing surfaces • Fog • Smooth shading • Surface mapping • Reflectivity • Transparent surfaces • Shadows • Refraction • Background surfaces • 3-D Brushes	• Anti-aliasing • Glowing surfaces • Fog • Smooth shading • Surface mapping • Reflectivity • Transparent surfaces • Shadows • Refraction • Soft-edged shadows • Inter-object illumination

!

The automatic renderer selection feature applies only to Flat shading, Gouraud, Phong, Raytracing, and RayPainting...

Wireframe, solid and quick-shaded algorithms must be “forced” by manually selecting them in the Renderer Selection field in order to use them for rendering a pixel-based image. The Raydiosity™ algorithm must also be chosen specifically through the pull-down menu.

The charts on the following pages presents the same information as the chart above in a different format. You may find one easier to understand than the other.

	Wireframe	Solid	Quick Shaded	Flat Shaded	Gouraud	Phong	Raytracing	RayPainting™	Raydiosity™
Anti-Aliasing	●	●	●	●	●	●	●	●	●
Draw-type PICTs	●	●	●						
Fog Effect				●	●	●	●	●	●
Glowing Surfaces			●	●	●	●	●	●	●
Inter-object Illumination									●
Pixel-type PICTs	●	●	●	●	●	●	●	●	●
Reflectivity						●	●	●	●
Refraction							●	●	●
Shadows							●	●	●
Soft Shadows									●
Smooth Shading					●	●	●	●	●
Surface Mapping						●	●	●	●
Transparency					●	●	●	●	●
3-D Brushes								●	

Enabling a specific rendering effect is only half the process. You must also choose a rendering method that supports that effect. This is done through either the Rendering Options or Rendering dialogs, as discussed earlier in this chapter.

Alternately, you can choose one of the pre-set rendering options from the Tool palette's pull down menu. They will automatically configure the rendering options for you, based on the type of rendering and quality level you request.

Commands used with rendered images

The following commands are used with the rendering and image/animation display features of the software. The list of commands is alphabetical, and an explanation is included with each item in this section.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

Dither Image command (Edit menu)

The Dither command allows you to choose whether an image being displayed on a monitor or output to a printer is to appear with dithering or no dithering. The Dither command is available whenever a pixel-based color image is contained in the active window. The command is enabled if a check mark appears before the command when the Edit menu is pulled down. To toggle the command on or off, select the command and release the mouse button.

Dithering is a diffusion technique used by 32-Bit Color QuickDraw to allow images of varying color depths to be displayed on monitors of any color depths. The effect is most noticeable when displaying a full-color image (32-Bit, 16.7 million colors) on a monitor with fewer colors available to it, usually 8-bit (256 colors or grays) or monochrome. The dithering process simulates actual colors in the image by mixing white or black patterns with colors available on the monitor.

In Vision3d, you can choose to display an image without dithering, meaning only the pure colors that are available on the monitor at that time will be used. Sometimes this improves the image quality, but most often you will see color artifacts in the image.

Environment Effects command (Rendering menu, ⌘H)

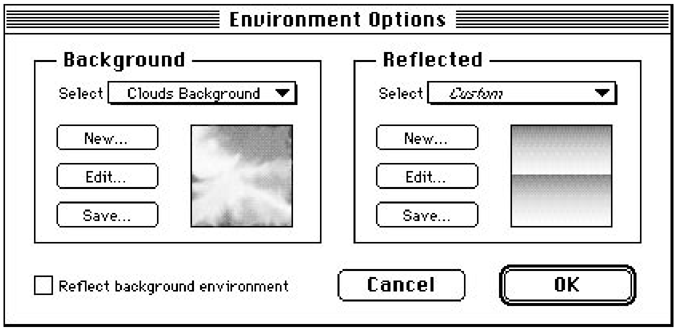
The Environment Effects command allows you to use various colors, gradations, or textures for backgrounds and reflective information in a rendering. This command is available whenever a model is open.

The default background environment for new models is a simple white color; the default reflective environment is a sky using dark colors. You may change the environment settings at any time. The specific environments loaded into a model are saved as part of the model document.

New environments that you create may also be saved as separate files for use with other models. This is done by clicking on the Save button in the Environment Options dialog. The currently-loaded environment — either background or reflected, whichever Save button is clicked — is saved as an environment file.

When an environment is loaded into a model, it is wrapped around the model on an imaginary sphere surrounding all objects; a sphere much like the one used to position directional light sources. The environment cannot be selected directly like an object, but may be managed through the Environment Options command and its various levels of dialogs.

Environments add a surrounding reflection and/or background to the scene. They also provide objects some type of context in the model. By using an environment instead of additional comprehensive modeling, the number of constructed objects or primitives to be rendered can be minimized or even reduced.



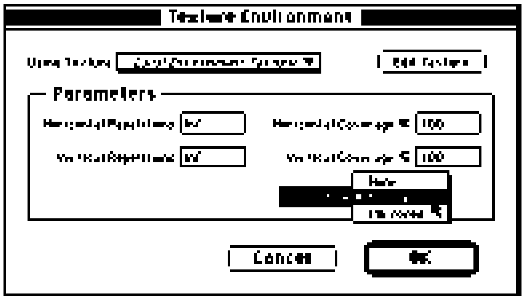
Environment Options

Background
Select: Clouds Background

Reflected
Select: Custom

□ Reflect background environment

When the Environment command is called, a dialog box opens to select and load one or more environment files and specify how they are to be used. The dialog is split into Background on the left and Reflected on the right, allowing you to use one environment for the background and another for the reflections that appear in shiny surfaces. Each half of this dialog works the same.



Texture Environment

Using Texture: [illegible]

Parameters
Horizontal Repetitions: [illegible]
Horizontal Coverage %: 100
Vertical Repetitions: [illegible]
Vertical Coverage %: 100

You should note that texture-based environments will often require different scale and mapping parameters for either of the two sides.

Backgrounds using textures with maps are usually scaled smaller. The scale for a texture-based environment is adjusted through the Edit button in the dialog. You should note that the horizontal and vertical coverage fields are expressed as percentages, i.e. 100% scales the map to cover the entire environment sphere once in the designated direction, a setting of 33% scales the map to cover the sphere three times in the chosen direction. The images appearing in the Environment Options dialog display how a portion of the environment looks as it is mapped onto the sphere. Use them to set the mapping scale larger or smaller and the tiling pattern as needed.

You will find it useful to load some of the environments included with Vision3d and see the scale settings used for them, then adjust you own environment creations in a similar manner.

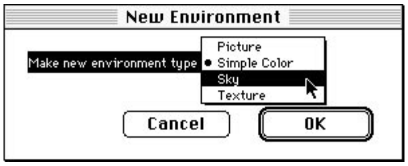
Loading environments

An environment may be loaded from the “Environments” folder using the Select field’s pull-down menu. Available environments found in the folder at the time the application is launched appear in the menu. Chose the desired environment by highlighting its name on the list, then release the mouse button. The selected environment will load into the model.

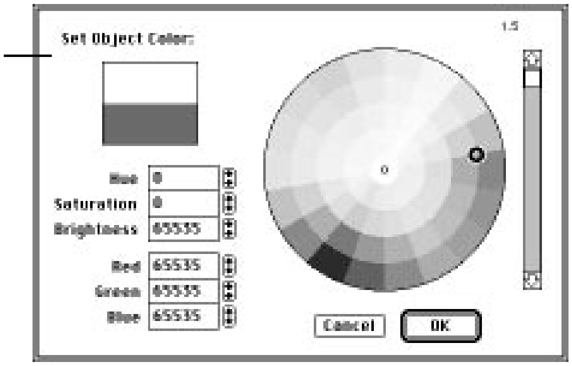
If the environment you want to use is not in the Environments folder at startup, you cannot load it into the model. You must quit the application, move the desired environment into the folder and re-start. No other way is provided to access environments.

New environments

Vision3d supports four types of environments: 1) Simple Color, 2) Sky, 3) Texture, and 4) Picture. Clicking on the New button opens a dialog to choose the type of environment you wish to create.



Then, based on the type of environment, a custom secondary dialog is opened to construct the environment type chosen. The four dialogs that open are shown in the illustration on the following page.



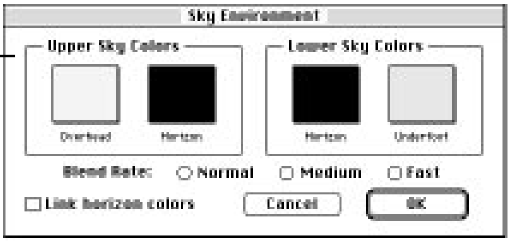
Simple color environment dialog

(A.K.A. the Apple Color Picker™)

Set Object Color:

Hue 0
Saturation 0
Brightness 65535

Red 65535
Green 65535
Blue 65535



Sky environment dialog

Allows you to pick four colors that create a graduated color for sky and ground plane

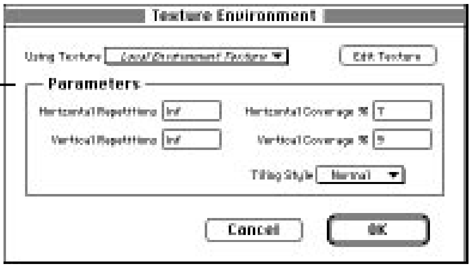
Sky Environment

Upper Sky Colors: Overhead, Horizon

Lower Sky Colors: Horizon, Underfoot

Blend Rate: ○ Normal ○ Medium ○ Fast

□ Link horizon colors



Texture environment dialog

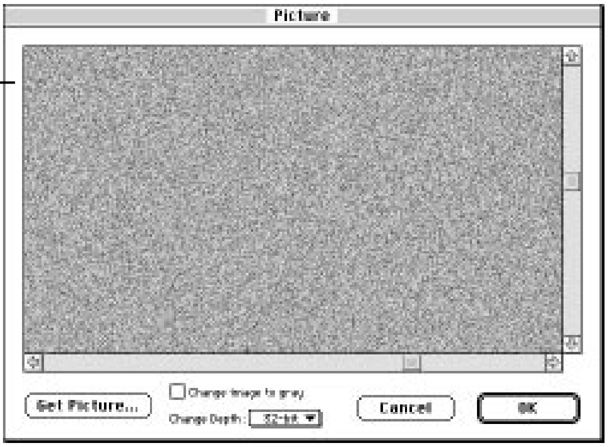
This dialog box appears for environments using a texture to set scale and tiling

Texture Environment

Using Texture: Local Environment Texture

Parameters

Horizontal Repetitions: Inf
Vertical Repetitions: Inf
Horizontal Coverage %: 7
Vertical Coverage %: 5
Tiling Style: Normal



Picture environment dialog

This dialog box appears to select the PICT image you wish to use as a Picture environment

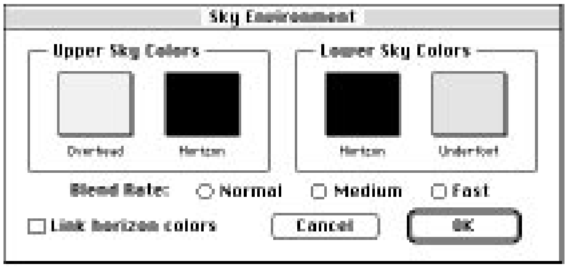
Picture

□ Change image to grey

Change Depth: 32-bit

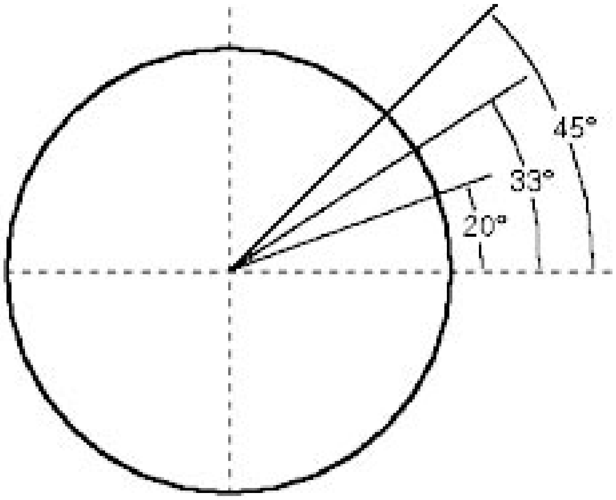
Sky environments

The Sky environment has been given an adjustable blend rate for better results in various situations. This is controlled by radio buttons in the environment's dialog box. An adjustable blend rate allows you to tailor the gradation to better match the lens angle of the view being rendered.



Normal blending uses a linear gradation from the horizon color to the overhead and underfoot colors. The transition from horizon color to sky color is complete by 45° above or below the horizon line. This is useful for wide angle lens settings.

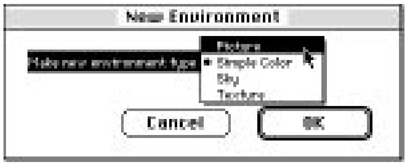
Medium uses an accelerated or "skewed" linear gradation to complete the horizon color's transition by 33° above or below the horizon line. This is useful for normal lens settings.



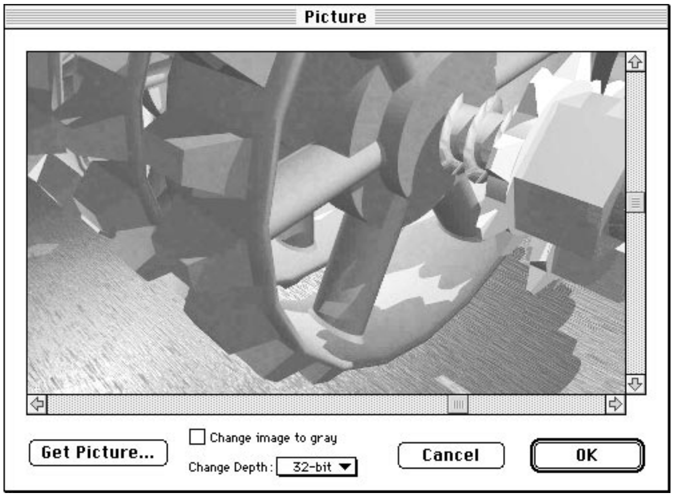
Fast uses a logarithmic gradation to complete the color transition by 20° above or below the horizon line. This is useful for semi-telephoto lens settings. Note that the Orthographic lens settings will not show a gradation of any kind, no matter how fast the transition rate.

Picture environments

The Picture class of environments has been provided to support using background pictures for rendered images. You can prepare a PICT image to be used as a background through the Environment Options command.



When "Picture" is chosen as the environment type and the "OK" button is clicked, the Picture dialog box is presented to choose and prepare the PICT you wish to use as a background.



□ Change image to gray

Change Depth: 32-bit

You may recognize this dialog as the same one used for loading PICT images for surface maps in textures. As before, the option for changing color depth is also available in this use of the dialog.

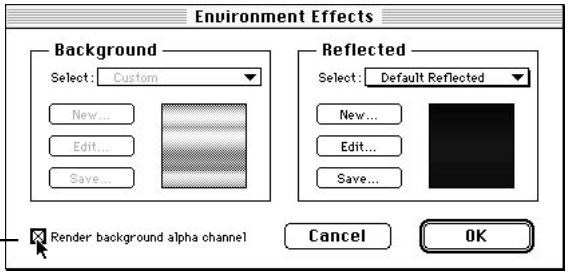
The background picture is positioned and sized to fit the active view — whether in a modeling window or a camera window — from which the snapshot is taken. If the active view is switched, the background picture will automatically re-size to match the active view's extents.

You must make sure the active view has the same proportions or aspect ratio as the PICT being used, or it is possible to distort the background picture. This is something you should watch while preparing the final rendering. If you define a snapshot smaller than the active view, the background picture is also cropped and you will see only that part of the background picture that is visible within the rendered image.

Alpha channel support

Vision3d provides an option to render an Alpha channel mask of the background behind your scene. Enabling this option, which appears as a check box in the Environment Effects dialog, creates a mask that can make the background transparent, if supported by the application in which you will be using the rendered image. This feature is available only if the rendering is saved in full-color (32-bit) PICT format.

The Alpha channel is another eight-bit channel in the 32-bit color image. Eight bits each for the red, green and blue components add up to only 24 bits, so the Alpha channel completes the 32 bits of information contained in the 32-bit color format. You can think of it as a grayscale image that masks the background.



Select: Custom

Select: Default Reflected

☒ Render background alpha channel

Environment Effects dialog

When alpha channel option is turned on, the background environment is disabled

The alpha channel allows easy placement of a rendering into a page layout document and facilitates the "compositing" of renderings with graphical elements from other software packages.

When the Alpha channel option is enabled, the background areas of the image are automatically identified during rendering. It doesn't matter what the background color is, because the background will be masked to be transparent. Note that enabling the alpha channel disables the background environment. The background of the image renders as black, with the gray scale mask in a separate channel to make the black area transparent.

To use the alpha channel option, select the Environment Effects command under the Rendering menu. Click on the check box in the lower left corner of the dialog window. The background environment side of the dialog grays out so it cannot be used, even if an environment has been previously loaded there. Click on the OK button, then render the scene as usual.

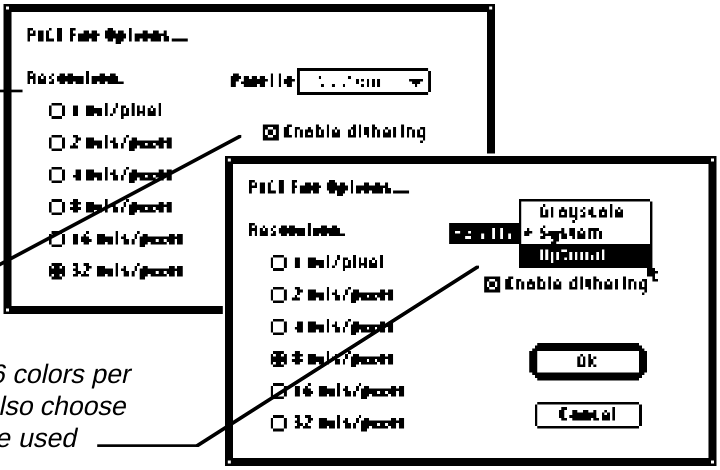
NOTE: *To realize the potential of alpha channel support, the finished rendering with background mask must be used in a software package that supports the alpha channel. Alternately, the mask must be converted to a clipping path in an image processing package like Adobe® Photoshop™ for use in packages that do not offer direct alpha channel support. Consult the user manual of the software package in which you plan to use your rendering with background mask to determine whether alpha channel support is present…and if so, how to use it.*

Make Optimal Palette command (Edit menu)

The Make Optimal Palette command generates an alternate set of colors that more closely match the colors used in a specific image. This command is available when a pixel-based color image is present in the active window.

Note: *Under System 6.0, only color palettes are optimizable. Gray scale palettes continue to use the standard system palette. Under System 7.0, both color and gray scale palettes may be optimized.*

Calculating and attaching an optimal palette to an image allows it to be displayed on monitors of reduced color depth with less dithering. Since the colors used to display the optimized image are more closely aligned with the colors the image actually requires, the effect is a cleaner, truer color display of that image. The optimal palette is present only while the target image is open in the active window. Closing the image removes the optimized palette from the System.


PICT File Options…

Resolution

○ 1 bit/pixel
○ 2 bits/pixel
○ 4 bits/pixel
○ 8 bits/pixel
○ 16 bits/pixel
● 32 bits/pixel

Palette: System

☒ Enable dithering

PICT File Options…

Resolution

○ 1 bit/pixel
○ 2 bits/pixel
○ 4 bits/pixel
● 8 bits/pixel
○ 16 bits/pixel
○ 32 bits/pixel

Palette:

☒ Enable dithering

When saving images, you can choose the color depth

…and whether the image will be “dither-able” when using QuickDraw to display it.

For color depths of 256 colors per pixel or less, you can also choose the type of color palette used

You may permanently attach an optimized palette to an image as part of the save operation. This is discussed in chapter 1 under the Save As command. Images saved with an optimized color palette can use that palette when they are read into Vision3d again or in other applications that support this feature. The mechanism that allows this process to work across applications resides in the System software and 32-Bit Color QuickDraw. An image with an optimized color palette read into software that uses QuickDraw to display its images can have the System replace the standard system color palette with the attached optimized palette.

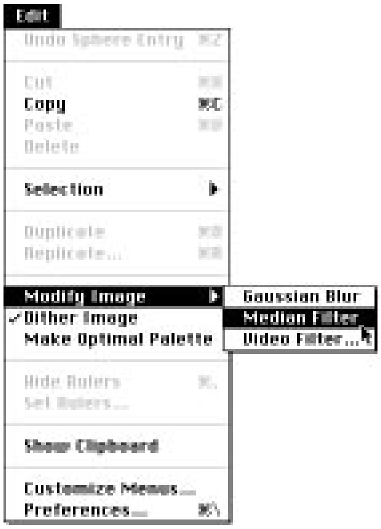
Modify Image command (Edit menu)

The Modify Image command allows the application of filters to an image. This command is available when a window containing an image is the active window. The image may be a completed rendering or one from another source read into Vision3d using the Open command in the File menu.

Vision3d provides two built-in filters: Gaussian Blur and Median Filter. With these filters, you can conveniently clean up visual noise or lightly blur an image.

Third-party filters

In addition, Vision3d supports extension filters designed by Strata Inc. or third-party developers specifically for Vision3d. Once installed, extension filters are accessed by this command. All filter commands operate on the entire image. Filters that have dialog boxes associated with them appear with an ellipsis (…) after their name in the sub-menu. Each time a filter is chosen that requires parameters, you must re-enter the values.

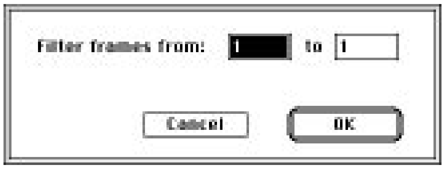


Vision3d also supports Adobe PhotoShop's plug-in filters sold by third-parties. To use, they must be copied into the Extensions folder within Vision3d's application folder before program startup. Under system 7, you can use aliases in the Extensions folder so you do not have to have a completed set of duplicate filters in two places on your hard-disk drive.

Choosing the Modify Image command displays a sub-menu, from which you can open specific filters for processing the image in the active window.

Using filters with animations

Both still images and animations can be post-processed. Both PICS and QuickTime Movie animation formats are supported. This feature may be used on animations from any source, not just Vision3d.



Filter frames from: 1 to 1

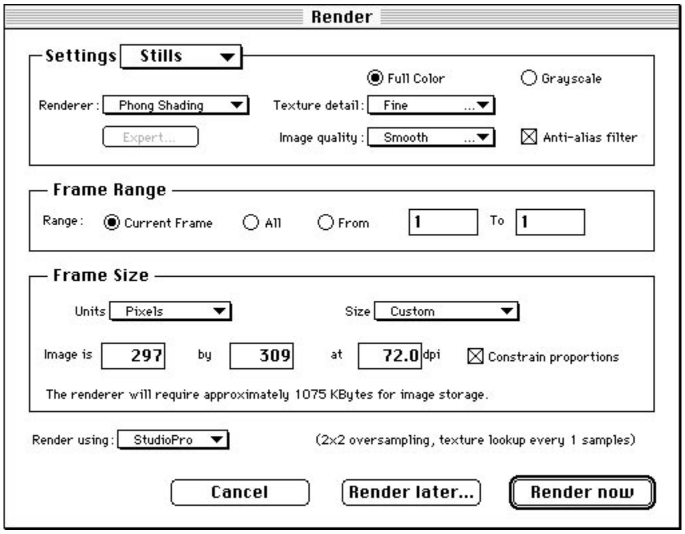
When using the Modify Image command, if a filter is applied to a single image, the process proceeds in a straightforward manner. If the active window contains an animation (in either QT Movie or PICS format), a dialog appears asking the range of frames within the animation to which you wish the filter effect applied.

The dialog always opens with only the current frame specified. You can change the dialog as needed before clicking on the "OK" button or pushing the Return key.

Render command (Rendering menu, ⌘Y)

The Render command initiates a rendering. Render is available whenever a model, shape, or camera window is active. Selecting the Render command opens a dialog box allowing you to choose the renderer, image quality, set the frame or number of frames to be rendered, and the image size and resolution.

Clicking "Render Now" or pushing the Return key initiates a snapshot of the model from the currently active view. Alternately, you can choose the Render Later option to save a suspended rendering directly from this dialog.



Settings: Stills

● Full Color ○ Grayscale

Renderer: Phong Shading

Expert...

Texture detail: Fine

Image quality: Smooth

☒ Anti-alias filter

Frame Range

Range: ● Current Frame ○ All ○ From 1 To 1

Frame Size

Units: Pixels

Size: Custom

Image is 297 by 309 at 72.0 dpi

☒ Constrain proportions

The renderer will require approximately 1075 KBytes for image storage.

Render using: StudioPro

(2x2 oversampling, texture lookup every 1 samples)

When the Render command is selected directly from the Rendering menu, the currently active view is used as the viewpoint for the rendering, as well as for size and proportion. The view's focal length setting also becomes the perspective used for the image.

Dimensions for a snapshot can be adjusted directly in this dialog. They may be expressed in pixels, inches or centimeters. This is chosen through the Units field pop-up menu. You may set the resolution as a function of the units chosen. This will be the finished image's resolution after the anti-aliasing interpolation has been performed, if used.

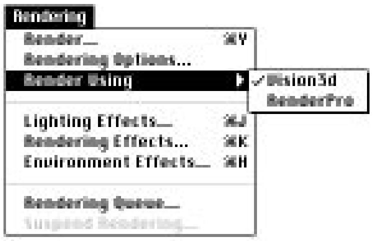
You may choose from a set of pre-defined sizes or enter your own custom size for the rendered image. Based on the size and resolution settings, a field at the bottom of the dialog provides feedback on the effective resolution (including anti-aliasing, if turned on), and how much disk space will be required while the image is being rendered.

Note that the size requirement shown is the absolute minimum, and may be more. You must have at least this much space free, either in RAM or for the virtual memory cache on your hard disk. Depending on the format chosen for saving the image when the rendering is complete, the saved file may be much smaller.

NOTE: *You may also access this dialog by holding down the Shift key while using the Camera tool to define a snapshot. This allows you to crop the scene being rendered, while also accessing the numerical control for image size, resolution, etc. provided by this dialog. When using the drag-define method with the Camera tool, adjustments made to the image size in the dialog apply to the area defined by the drag marquee.*

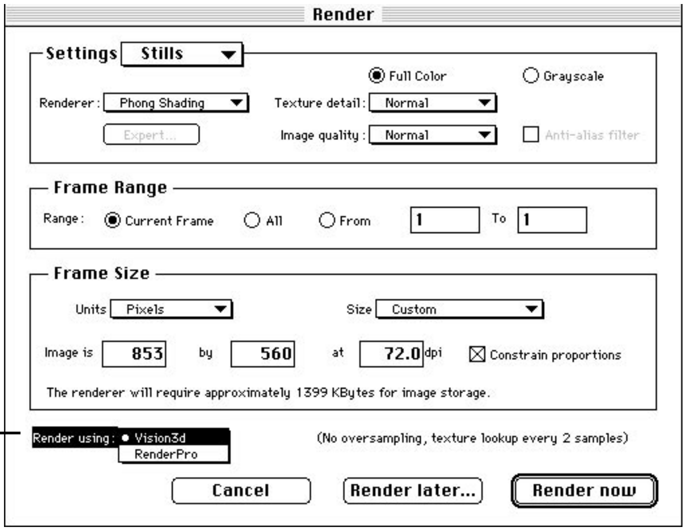
Render Using command (Rendering menu)

Use this command to select the source for rendering images. Vision3d defaults to using its own built-in rendering algorithms each time it is run. To use an alternate rendering "engine" or application, you must select it from this command's sub-menu. Only one can be chosen at a time.



Only those rendering alternatives specifically supported by Vision3d appear in the sub-menu. Some choices may communicate transparently with Vision3d; returning the finished rendered image back to Vision3d for displaying or saving. Others may require Vision3d to create a separate suspended-rendering or export file that must be manually opened in the external rendering application. The specific process for using another renderer is explained in the documentation that accompanies that application.

Render using pull-down menu



Render

Settings: Stills

● Full Color ○ Grayscale

Renderer: Phong Shading

Texture detail: Normal

Image quality: Normal

□ Anti-alias filter

Frame Range

Range: ● Current Frame ○ All ○ From 1 To 1

Frame Size

Units: Pixels

Size: Custom

Image is 853 by 560 at 72.0 dpi ☒ Constrain proportions

The renderer will require approximately 1399 KBytes for image storage.

Render using: ● Vision3d / RenderPro

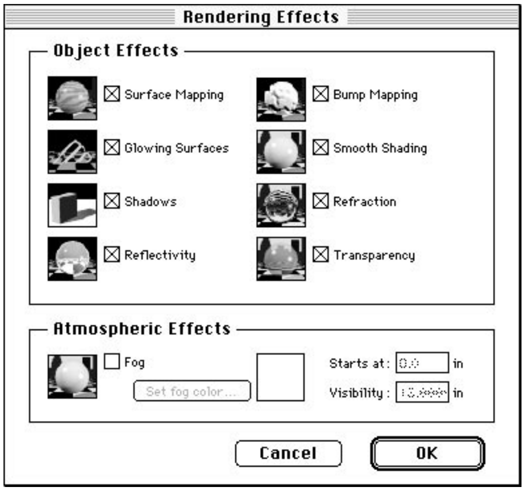
(No oversampling, texture lookup every 2 samples)

The choice of built-in or alternate renderers can also be made through the Render command's dialog box. A pull-down menu duplicates the “Render Using” command’s sub-menu in that dialog box. It shows the currently-selected rendering source and allows you to change it, if needed, at that level too.

The primary use of the Render Using command is to support RenderPro, Strata Incorporated’s network distributed-rendering system. The Render Using command is where you would choose to use RenderPro™ rather than rendering images locally. It allows you to harness the processing power of multiple Macintoshes connected on a network to render an image or animation sequence significantly faster than a single machine can do it alone.

Rendering Effects command (Rendering menu, ⌘K)

The Rendering Effects command is used to turn the rendering effects on or off globally. This command is available at all times.



Rendering Effects

Object Effects

☒ Surface Mapping

☒ Bump Mapping

☒ Glowing Surfaces

☒ Smooth Shading

☒ Shadows

☒ Refraction

☒ Reflectivity

☒ Transparency

Atmospheric Effects

☐ Fog

Set fog color...

Starts at : 0.0 in

Visibility : 10.0000 in

Choosing this command opens the Rendering Effects dialog. This is where you enable or disable the effects you wish to see in a rendering. Each time Vision3d is launched, all the rendering effects are turned on, except Fog. This is the default configuration.

You must consciously open this dialog and choose to turn off the effects supported in Vision3d. Turning off effects that you do not need or want in a rendered image or animation can improve the rendering performance.

The effects available in this dialog are as follows:

- **Surface Mapping:** This field enables Vision3d’s mapping feature. Two types of mapping are supported in Vision3d: Surface and Solid.

Both types have their strengths and weaknesses. Surface mapping is the process of using two-dimensional PICT images to alter the surface of objects. Surface mapping is the most usable and flexible, and often produces the most realistic images because the maps are usually taken from real objects. Solid uses a mathematical formula to generate a pattern that is volume dependent. They must be compiled externally before use in Vision3d, but are more memory efficient and generally render faster. Some solid textures are shipped in the Vision3d product. More solid textures are available from Strata Inc. in the form of commercial libraries.

Surface mapping is available for use with the Phong, Raytracer, RayPainter, and Raydiosity algorithms.

- **Glowing Surfaces:** Objects with this attribute have a luminescence greater than the light shining on them. This effect is often used to give an object or group of objects a neon-like appearance. The actual amount of glow for each object must be set in the texture applied to that object.

  Glowing surfaces is available with the Gouraud, Phong, Raytracer, RayPainter, and Raydiosity algorithms.
- **Shadows:** This effect enables calculating shadows in the rendered image. All objects cast shadows unless an object's shadow is specifically turned off with the "Passes light" field in the texture applied to the surface of that object. Lights may also be given the option to not cast shadows through the Adjust Object dialog for the light source.

  The shadows effect is available with the Phong, Raytracer, RayPainter, and Raydiosity algorithms.
- **Reflectivity:** Objects with a reflectivity component in their surface produce a mirror-like appearance. The degree of reflectivity is variable with a specific object's texture settings. Enabling this rendering effect performs the calculations necessary to show reflections in reflective surfaces.

  Reflectivity is available with the Phong (limited to environments), Raytracer, RayPainter, and Raydiosity algorithms.
- **Bump mapping:** This effect allows any bump-type surface maps present in textures to influence the appearance of object surfaces. Turning this effect off disables all bump maps throughout a model during rendering.

  The bump mapping effect is available with the Phong, Raytracer, RayPainter, and Raydiosity algorithms.
- **Smooth Shading:** This effect determines whether object surfaces are rendered smooth or faceted. Most curved surfaces are modeled as a

collection of polygons. In the lower level rendering methods, objects will appear faceted. Faceting is the result of sudden changes in light intensity from one polygon to the next, represented in a rendered image as a change in color value.

The more sophisticated renderers can obtain a smoother surface appearance on objects by averaging the lighting change across curved surfaces, but only if the object is a primitive or its surface has been "smoothed," i.e. defined by a triangulated mesh with vertex normals present.

The smooth surfaces effect is available with the Gouraud, Phong, Raytracer, RayPainter, and Raydiosity algorithms.

- **Refraction:** Light shining through transparent objects is "bent" by the object's material. This phenomenon, called refraction, is observed often in the real world. As a transparent object is being rendered, its surface must be modified to include contributions from light in back of the object. In most cases, this light is reflected from surfaces of other objects. Some of Vision3d's renderers can take refraction into account, depending on the lighting and placement of images behind them. This option adds refraction into the imaging calculations.

  Refraction is available with the Raytracer, RayPainter, and Raydiosity algorithms.

- **Transparency:** Objects with this attribute have a crystalline or glass-like appearance. The degree of transparency is variable with a specific object's texture settings.

  The Wireframe, Hidden Line and Quick Shading algorithms do not support transparency. Flat shading, Gouraud and Phong rendering methods implement transparency using a "screen door" effect to see through an object's surface to those behind it. The amount of transparency is fixed in levels of zero, 25%, 50%, 75% and 100%. Ray tracing, RayPainting, and Raydiosity can render transparent objects in fully variable amounts corresponding to the values in the transparency field for the surface on each object.

  Transparency is available with the Flat shading, Gouraud, Phong, Raytracer, RayPainter and Raydiosity algorithms.

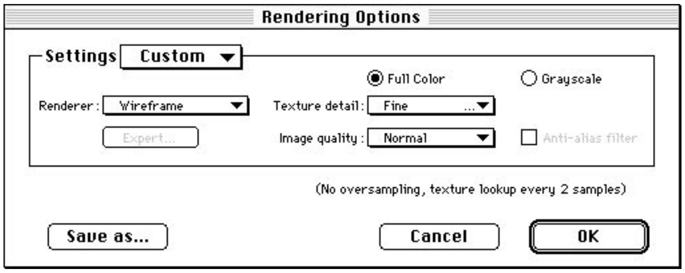
- **Fog:** This effect enables a fog-like appearance in the rendered image. While not a true volumetric effect based on spatial density, it does simulate the effects of fog based on distance from the viewing plane.

  Fog is available with the Flat Shaded, Gouraud, Phong, Raytracer, RayPainter, and Raydiosity algorithms.

- **Fog Color, Starting Distance, and Visibility:** When the fog option is enabled, secondary fields become active to set the color of the fog, the distance from the viewing position when it starts in the scene and how dense the fog appears. There are no default or recommended settings for these fields.

Rendering Options command (Rendering menu)

The Rendering Options command is used to set (and optionally save) the parameters for a rendering. This command is available whenever a model is open. In this dialog, you can manually choose the rendering method or, based on the setting chosen, Vision3d can select the most appropriate rendering algorithm to accomplish the task.



Settings: Custom

● Full Color ○ Grayscale

Renderer: Wireframe

Texture detail: Fine

Image quality: Normal

□ Anti-alias filter

(No oversampling, texture lookup every 2 samples)

Choosing this command opens a dialog to select options desired for the next rendering you initiate. While this dialog is open, you may also save various combinations of options by name for use as pre-defined settings with the pull-down menu in the Tool palette or in this dialog and the Render dialog.

The Rendering Options dialog includes the following fields:

Gray scale rendering

The gray scale option renders the model in 256 shades of gray. This option is useful for rendering images that must reproduce well using a B&W output device, such as a laser printer or high-resolution typesetter, or for use in B&W printing applications.

Selecting gray scale forces both the screen image and the internal image to represent colors in shades of gray, even if the model has multiple-color attributes assigned to objects. The resulting image is saved in 8-bits per pixel format

Full-color rendering

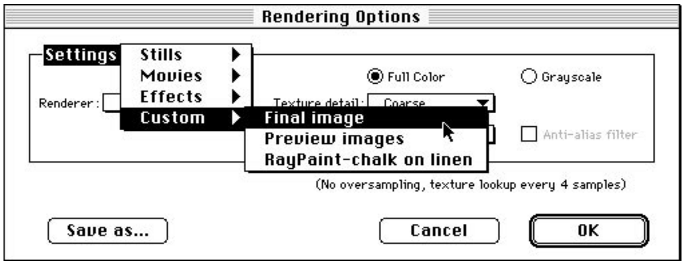
Selecting full color renders images in a format compatible with 32-bit Color QuickDraw. The image is rendered in full 32-bit color, even if the monitor displaying the rendering window is not capable of showing that much color depth. Color QuickDraw provides a standard way of handling images, no matter what device is used to reproduce or display it — from B&W laser printers to color film recorders, from gray-scale monitors to high resolution video film.

The 32-bit Color QuickDraw format supports B&W, 8-bit, 16-bit and 24-bit color devices, allowing full-color images to be displayed or output in one color, 256 colors, thousands or millions of colors. Approximation of an image's actual colors at lower resolutions is accomplished via dithering.

The rendering window displays the image at the best capability of the monitor. To see the full range of colors contained in a 32-bit color image at the time of rendering, you will need a full color monitor. The resulting image is saved in 32-bits per pixel format.

Settings

This field allows you to set the current options and values in the dialog to a previously-defined configuration that was saved through the Save As button. Previously-defined rendering options are saved between sessions in the Preferences file and are available to all models. However, the Preferences file is not updated with any newly-saved rendering options until Vision3d is quit. If Vision3d is exited other than through the Quit command, the newly-saved rendering options will not be available at the next session.



Rendering Options

● Full Color ○ Grayscale

Texture detail: Coarse

□ Anti-alias filter

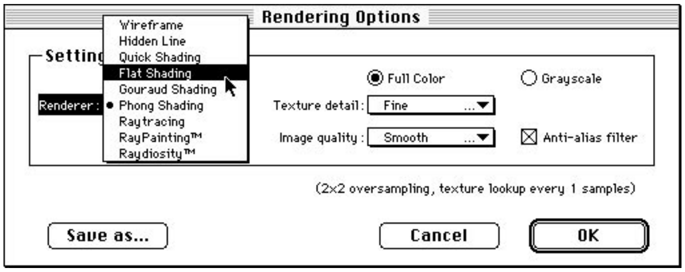
(No oversampling, texture lookup every 4 samples)

The pre-defined options list appears in the pull-down menu for selection. Picking an entry affects all the check boxes, numbers and settings in the Rendering Options dialog, including those in the Raytracing, RayPainting and Raydiosity Esoterica dialogs. Thus, whatever values are in these fields at the time a configuration is saved will be re-entered into those same fields upon selecting the choice again from the Settings pull-down menu.

The list of pre-defined rendering options is also available in the Tool palette and the Render dialog.

Renderer selection

Clicking on this field displays a pull-down menu. Vision3d uses the rendering mode chosen even when the rendered image desired doesn't require it or, conversely, when a higher level of rendering may actually be preferred.



Renderer: ● Phong Shading

◉ Full Color ○ Grayscale

Texture detail: Fine

Image quality: Smooth

☒ Anti-alias filter

(2x2 oversampling, texture lookup every 1 samples)

The following choices appear in the menu:

- **Wireframe:** With Wireframe selected, only the outlines of objects are drawn, without indication of surfaces. This is one of the methods Vision3d uses to represent objects in the modeling windows, but when chosen here for rendering, the Anti-aliasing capability is also available.
- **Hidden Line:** This setting produces a rendering of wireframe objects, but with surfaces implied by the removal of hidden lines. Again, this is a display method used for the modeling windows, but is also available here.
- **Quick Shaded:** This setting renders an image with simple shaded objects. In addition to showing object surfaces, lighting direction and the resulting shading of objects is calculated. This is a simple surface rendering, using only one lighting calculation per facet.
- **Flat Shading:** Selecting Flat Shading renders with solid surfaces and colors. Transparency is supported using a screen door appearance in increments of zero, 25%, 50%, 75% and 100% transparency.
- **Gouraud:** Gouraud is a shading-oriented rendering algorithm that refines how an objects' surfaces are shaded. This rendering mode can execute the smooth shading option if the object is either a primitive type or defined as a triangulated polygon mesh with appropriate vertex normals.
- **Phong:** Phong rendering is a light-oriented algorithm, with the capability to render smooth surfaces, reflectivity (environments only), shadows, transparency, and surface maps.

- **Raytracing:** This rendering method calculates how rays of light react to object surfaces within the model before reaching the viewer's eye. Raytracing creates enhanced photo-realistic images with true reflectivity, shadows, refraction, transparency, etc.
- **RayPainting:** This rendering method provides an artistic flare that diverges from the photo-realistic intentions of other high-level renderers. RayPainting creates images with 3-D brush strokes on textured backgrounds.
- **Raydiosity:** This setting produces images of unparalleled realism. This rendering method considers the effects of inter-reflections among objects in its calculations. Raydiosity is especially useful for scenes with indirect lighting.

The choices discussed in this menu are presented in lowest-to-highest order of rendering sophistication. Depending on the rendering method chosen, a finished image can take from just a couple of seconds to several hours. You must balance the need for realism with the time the higher-quality rendering algorithms require to perform their work.

Image quality

The Image Quality field sets the rendering detail for the desired image quality. Predictably, each increase in the level of image quality increases the quality of the rendered image and also requires longer to render. The options are represented in this dialog with descriptive titles.

- **Anti-aliasing:** The "stair-stepped" appearance visible on diagonal edges and lines can be minimized or eliminated with this effect turned on. This is done by choosing either the Smooth or Extra Smooth image quality setting.

  Anti-aliasing is independent of any specific rendering algorithm. It may be used with any set of rendering effects and/or rendering algorithm. The level of anti-aliasing chosen depends on the need. The Smooth setting is recommend for all but the most critical imaging situations.
- **Anti-aliasing Filter:** When the Smooth or Extra Smooth option is used in the Image Quality field, the "Anti-alias filter" checkbox becomes active. This enables an additional pixel averaging step that further enhances the appearance of a rendered image.

Texture detail

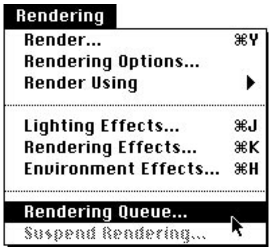
Texture detail is set independently of the rendering algorithm or the image quality. Of course, not all rendering methods can use the information provided by textures. As the level of detail increases, the quality of the surface detail increases, with a corresponding increase in rendering time.

This field sets the rate at which surface maps are sampled. A setting of Fine means the maps are sampled every pixel. A setting of Normal samples every other pixel; Medium samples every third pixel; and Coarse samples every fourth pixel. The pixels in between the sampled pixels are extrapolated and filled in by Vision3d.

The sampling rate is a global setting, affecting all objects in the scene being rendered. The sampling rate chosen depends on the desired detail of the maps. It involves the usual trade off between image quality and rendering speed. The lower the detail, the fewer times a map must be sampled during the rendering process. Typically, you would use the lower detail settings for preview renderings during the modeling and scene-assembly phase, then use a sampling rate of one or two (fine or Normal) for the finished image.

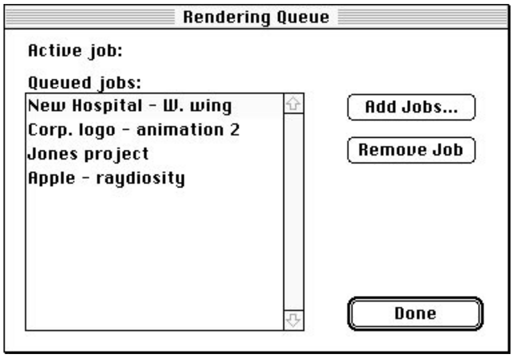
Rendering Queue command (Rendering menu)

This command allows you to add suspended renderings to or remove them from the rendering queue. The Rendering Queue command is available all the time.



A queue is established when one or more suspended renderings are added to the job list in the Rendering Queue dialog. Only one rendering is active at a time. The order in which the jobs are added to the queue determines the order they are rendered. Clicking on the Done button begins the loading and rendering of the first rendering in the queue.

When a queued rendering is closed, either because it has completed and been saved or it has been terminated by clicking on the close box, the next rendering in the queue (if any) loads and begins rendering.



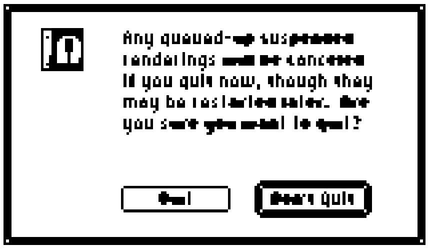
Rendering Queue

Active job:

Queued jobs:

- New Hospital - W. wing
- Corp. logo - animation 2
- Jones project
- Apple - raydiosity

If Vision3d is quit while renderings are in the queue, they are deleted from the queue. Vision3d does not remember the queuing of renderings between sessions. An alert is displayed announcing that renderings are still present in the queue, and asking if you are sure you want to quit.



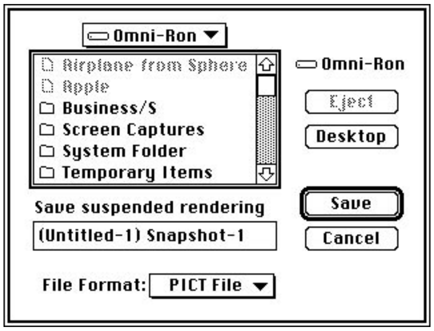
Any queued-up suspended renderings will be canceled if you quit now, though they may be restarted later. Are you sure you want to quit?

If you want to render more than one image at a time, you must open them manually. The rendering queue has no provision to support multiple renderings at one time. Remember, with more than one rendering in progress at a time, you must send them all to the background so they will share CPU time. Otherwise, the one in the active window (foremost on the desktop) takes priority and all the CPU time…just like using the rendering queue.

Suspend Rendering command (Rendering menu)

The Suspend Rendering command allows you to save a rendering in progress, which can then be restarted at a later time. Suspend Rendering is available when a rendering window is the active window, and the rendering is in progress.

All the model data, as well as the rendered image at its current state of completion, are saved as a self-contained file. You do not need to have the original model open to restart the suspended rendering.



Save suspended rendering
(Untitled-1) Snapshot-1

File Format: PICT File

Suspended renderings are re-started using the Open command in the File menu. Upon restarting a suspended rendering, its rendering window re-opens to monitor the rendering's progress just as before.

Suspended renderings are useful when you need to schedule more involved Vision3d projects around other tasks requiring the full attention of the computer. Suspended renderings may be restarted, allowed to render for a time, suspended again, then restarted again…as time permits and as often as necessary to complete the rendering.

Additionally, the Rendering Queue command is available for queuing several renderings to load, render, and save automatically. As mentioned in that command's discussion, it works only with suspended renderings.

GLOSSARY

- **Anti-aliasing.** A mechanism to prevent or remove the jagged appearance of diagonal lines or edges in an image. Anti-aliasing can be achieved in rendered images by averaging adjacent pixels with sharp variations in color and brightness or by increasing the resolution of the image to meet or exceed the resolution of the device displaying or printing it.
- **Bitmap.** Also known as a pixel image. An image composed of pixels. May be any resolution or color depth. Rendered images may be saved as a bitmap file in either PICT or TIFF format. Surface maps are contained in textures as bitmaps.
- **Bounding Box.** An imaginary rectangle that exactly encloses the geometry of an object or group of objects. When an object is selected, handles appear at the corners and midpoints of the sides of this rectangle. When an object is manipulated using one of the object-handling tools, its bounding box appears to interact with the tool and provide visual feedback during the tool operation.
- **CAD.** An acronym that stands for Computer-Aided Design.
- **Click.** The act of positioning a tool cursor in a view, and usually on an object, then pressing the mouse button. This event initiates a process or operation related to the tool's function.
- **Click-Drag.** The process of clicking the mouse button and, while holding the mouse button down, dragging the cursor in the view. For example, this procedure would be used to move an object from one location to another in a view.
- **Cursor.** An icon indicating the current tool selected and/or the current focus of the program. Vision3d uses different cursors to indicate what action is occurring.
- **Database.** The area of memory within the program while it is running that is set aside to keep track of objects within a model.
- **Default.** A parameter or setting pre-defined in the program which may be changed by the user.
- **Diffuse Reflection.** That component of the light reflecting from a surface caused by its dull or matte nature. Dull or Matte surfaces

reflect the light striking them in random angles over a large area; giving the surface an equally-bright appearance from a wide range of viewing positions.

- **Double-click.** The act of positioning the cursor on an object in the active view or an icon in the interface, then pressing and releasing the mouse button twice in rapid succession.
- **Extrusion.** A method of creating a 3-D object using a 2-D template; giving depth to the 2-D shape.
- **Flat Shading.** Shading by filling each facet of a surface with a single color.
- **Free-form modeler.** Modeling editor that uses spline patches to define the geometric of an object. Tools are provided to move spline handles to form the geometry as desired. Multiple views allow manipulating the object from different directions to more easily see and modify the shape.
- **Gouraud Shading.** A rendering method that generates a smooth-appearing surface similar to the Phong shading method, but uses normals attached to the vertices for its shading calculations.
- **Group.** A collection of objects that act as one.
- Hierarchical Object. Creating an object shape in a separate work space, which when instantiated in the model retain a link to the shape. Changes made to the shape are automatically applied to the instantiations.
- **Lathing.** A method of creating a 3-D object by revolving a 2-D profile about a designated axis.
- **Marquee.** A rectangular dotted box created by dragging the cursor. Marquees appear in the programs windows or views to provide visual feedback during various tool operations.
- **Mesh Surface.** A surface that has common vertex points between adjacent polygons. Mesh surfaces are typically used to define complex forms. The surface also provides smoothing information for the rendering algorithms.
- **Normals.** The mathematical value that indicates the direction a surface is facing in 3-D space. Normals may be attached to individual polygons or to the vertex points that define the polygons.
- **Orthographic.** A method of displaying objects in a view where parallel lines do not converge. Orthographic projections are used to compare object's absolute dimensions, without the confusion of relative distance from the view.

- **Parent/Child Link.** An internal connection between two objects, in which the child is linked to the parent for constraint purposes. Various uses exist in different applications for linked objects, but are usually related to animation.
- **Perspective.** A depth cue available in Vision3d in which parallel lines converge to align with a designated vanishing point. Perspective is a product of several parameters in the displayed view: lens focal length, camera size, and distance from objects.
- **Phong Shading.** A rendering method that creates a smooth surface by blending the boundaries between adjacent polygons of a mesh surface. Shading calculations are based on the normals attached at the center of each polygon. Phong Shading supports surface and volumetric mapping.
- **Pixel.** An acronym that stands for Picture Element. It is the smallest component which makes up the display on a computer monitor. Each dot on the screen is a pixel. Many images displayed on the screen are likewise stored in a pixel form that is mapped to the screen pixels for viewing.
- **Primitives.** Basic geometric elements from which complex objects can be built.
- **Refraction.** The change in direction of light as it passes from one transparent material to another. This causes an apparent shift in the image showing through the transparent material.
- **Specular Reflection.** That component of the light reflecting from a surface caused by its shiny or glossy nature. Shiny surfaces reflect light striking them in clearly defined angles of incidence; resulting in "hot spots" corresponding to the direction of the light sources providing the illumination.
- **Surface Mapping.** A process in which a PICT image is used to define an object's surface properties. A separate map is used to define color, texture, reflectivity, transparency.
- **Sweep.** A method of creating a 3-D object from a 2-D template using a combination of lathe and extrude. The template is both pushed through space and revolved around an axis at the same time to create a geometrically complex shape. This type of modeling would be used to create the threads on a bolt, for example.
- **Vertex.** A point of intersection of two vectors or a point used to define a polygon.
- **Volumetric Mapping.** A process in which a map is defined as a 3-D volume, but which is only visible on the surface of an object. The object

has the appearance of being carved out of the volume. Volumetric maps may also be used to define multiple surface characteristics.

- **Window.** An element of the Macintosh desktop interface that contains one or more views of the 3-D model. There are several types of windows used in Vision3d. the window frame is maintained by the operating system; the contents within the window is maintained by Vision3d.
- **Wireframe.** A fundamental rendering method that represents 3-D objects with connecting lines. No surfaces are shown, but may be implied if hidden lines are removed for the resulting image.
- **X axis.** The axis that describes the horizontal position of an object in 3-D space. This may be expressed in absolute coordinates or view-relative coordinates.
- **Y axis.** The axis that describes the vertical position of an object in 3-D space. This may be expressed in absolute coordinates or view-relative coordinates.
- **Z axis.** The axis that describes the depth position of an object in 3-D space. This may be expressed in absolute coordinates or view-relative coordinates.

BIBLIOGRAPHY

Here are reference materials pertaining to the rendering algorithms used in Vision3d. The books in the list discuss the theory of three-dimensional graphics and explain many concepts pertaining to 3-D rendering. They should provide a good background in understanding how Vision3d's rendering algorithms work. You should be aware that the material in these books can be very technical. You may also find that they are dated. The technology and software available in the field of three-dimensional modeling/rendering is progressing very rapidly.

- **"An Introduction to Ray Tracing,"** Academic Press © 1989. ISBN #0-12-286160-4. Edited by Andrew S. Glassner.
- **"Computer Graphics for Designers and Artists,"** Van Nostrand Reinhold Company Inc. © 1986. ISBN #0-442-24712-5. Written by Isaac V. Kerlow and Judson Rosebush.
- **"Computer Graphics - Systems & Concepts,"** Addison-Wesley Publishing Company © 1987. ISBN #0-20-114656-8. Written by Rod Salmon and Mel Slater.
- **"Computer Vision3d,"** Prentice-Hall, Inc. © 1982. ISBN #0-13-165316-4. Written by Dana H. Ballard and Christopher M. Brown.
- **"The Mathematical Structure of Raster graphics,"** Academic Press © 1989. ISBN #0-12-257960-7. Written by Eugene L. Fiume, University of Toronto, Ontario, Canada.
- **"Computer Graphics: Principles and Practice,"** Addison-Wesley Publishing Company © 1990. ISBN #0-201-12110-7. Written by James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes.
- **"A Progressive Refinement Approach to Fast Radiosity Image Generation,"** *COMPUTER GRAPHICS* (SIGGRAPH '88 Conference Proceedings), Vol. 22, number 4, August 1988. ISBN #0-89791-275-6. Written by Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg, Cornell University.
- **"A Ray Tracing Solution for Diffuse Inter-reflection,"** *COMPUTER GRAPHICS* (SIGGRAPH '88 Conference Proceedings), Vol. 22, #4, Aug. 1988. ISBN #0-89791-275-6. Written by Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear, Lawrence Berkeley Laboratory.

Of course, each of these books contains a large Bibliography that can point you to other books, publications or graduate-level theses on the subjects of computer modeling and three-dimensional rendering.

Textures Settings for Common Materials

	Ceramic	Paint	Paper	Plastic	Plexiglass	Rubber	Titanium	Chrome	Glass
Ambient Fraction	0.55	0.35	1.00	1.00	0.00	0.15	1.00	0.80	0.20
Bump Amplitude	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Diffuse Fraction	0.60	0.50	1.00	1.00	0.20	0.70	0.00	0.80	0.20
Glow Factor	0.00	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00
Reflectivity	0.25	0.15	0.00	0.15	0.15	0.00	0.30	1.00	0.20
Refraction Index	1.00	1.00	1.00	1.00	1.20	1.00	1.00	1.00	1.50
Specular Fraction	0.95	0.33	0.05	0.90	1.00	0.05	1.00	1.00	0.80
Transparency	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.90
First Smoothness	800	256	30	512	1024	6	900	1100	850
2nd Smoothness	0	0	0	0	0	0	6	10	0
Second Weight %	0	0	0	0	0	0	65	25	0
Enable shadows	yes	yes	yes	yes	yes	yes	yes	yes	yes
Link Spec/Diff	no	no	yes	yes	no	yes	yes	yes	no
Prop. scale maps	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

REFRACTION: LIST OF MATERIALS

The following list of indexes of refraction was compiled and edited from the 63rd edition of the "Handbook or Chemistry and Physics" published by CRC press. This list is offered as a guide only in applying custom refractive indexes to textures which will be simulating these materials.

Minerals

Amber	1.455
Amethyst	1.544
Argonite	1.530
Azurite	1.730
Bentonite	1.757
Beryl	1.565
Borax	1.4466
Brucite	1.560
Calcite	1.658
Calomel	1.973
Celestite	1.621
Chlorite	1.570
Chromite	2.160
Diamond	2.4175
Dolomite	1.679
Emerald	1.576
Erythrite	1.626
Fluorite	1.433
Gypsum	1.519
Halite	1.544
Helvite	1.728
Hematite	3.220
Lazulite	1.604
Leucite	1.508

Magnetite	2.020
Moss Agate	1.540
Onyx Marble	1.486
Opal	1.450
Quartz	1.544
Rose Quartz	1.550
Rhodonite	1.711
Ruby	1.766
Sapphire	1.770
Serpentine	1.530
Smoky Quartz	1.550
Spinel	1.719
Sulfur	1.958
Talc	1.539
Topaz	1.606
Turquois	1.610
YAG Rhodo chrosite	1.830
Zircon	1.923

Inorganic Compounds

Aluminum Chloride	2.700
Aluminum Oxide	1.665
Barium Oxide	1.980
Calcium Borate	1.540
Calcium Carbide	1.750
Calcium Sulfite	1.590
Copper Chloride	1.644
Common Glass	1.600
(ranges from 1.5 to 1.9, depending on composition)	
Ice	1.305
Iron Sulfate	1.802
Lead Nitrate	1.782
Lithium Bromide	1.784
Lithium Chloride	1.662
Lithium Oxide	1.644

Manganese Borate	1.617
Manganese Chloride	1.555
Mercury Cyanide	1.645
Mercury Iodide	2.748
Potassium Cyanide	1.410
Potassium Fluoride	1.352
Sodium Bromide	1.479
Sodium Cyanide	1.452
Sodium Nitrate	1.587
Sodium Phosphate	1.440
Strontium Fluoride	1.442
Strontium Oxide	1.870
Tin Iodide	2.106
Water	1.333
Zinc Bromide	1.545
Zinc Silicate	1.616

Organic Compounds

Methanol	1.326
Ethyl Ether	1.352
Acetone	1.357
Ethanol	1.359
Ethyl Acetate	1.370
Acetic Acid	1.370
Proprionic Acid	1.385
Butyric Acid	1.397
Sulfuric Acid	1.427
Ethelene Glycol	1.434
Chloroform	1.444
Fluorobenzene	1.463
Toluene	1.494
Benzene	1.498
Methyl Benzoate	1.515
Benzyl Alcohol	1.538

Realistic Gems

Another excellent source of information on the optical properties for various materials is “GEMSTONES of the World,” by Walter Schumann. ©English-language edition: Sterling Publishing Co. Inc., New York 10016. ISBN 0-8069-3089-6.

You should note, however, that in order for polygonal objects in Vision3d to realistically render with the type of complex crystalline appearance of precious and semi-precious gems, they must contain an interior structure of transparent surfaces through which the light can interact and refract realistically. Additionally, you must pay special attention to the following points:

1. You must model the correct geometry for the gem(s) you wish to simulate, i.e. — does the gem have a Cubic, Tetragonal, Hexagonal, Trigonal, Orthorhombic, Monoclinic, or Triclinic crystalline shape?
2. You must create and apply a texture that uses a second smoothness value corresponding to the internal crystalline structure. (This is definitely one of those trial and error type of projects.)
3. You must increase the value in the transparency recursion field of the Raytracing Esoterica dialog box (opened through the Rendering Options command’s dialog box) so that the multiple transparent surfaces will be fully noted and rendered by the raytracer.

In the case of rendering realistic gems, the raytracer is definitely the algorithm of choice.