



Release Notes

Version 3.51
for IRIX and Windows NT®

© Copyright 1988–1996 Microsoft Corporation. All rights reserved.

SOFTIMAGE® is a registered trademark of Softimage Inc., a wholly owned subsidiary of Microsoft Corporation, in the United States, Canada, and/or other countries.

Microsoft®, MS®, MS-DOS®, Windows®, and Windows NT® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks belong to their respective owners and are hereby acknowledged.

This document is protected under copyright law. The contents of this document may not be copied or duplicated in any form, in whole or in part, without the express written permission of Microsoft Corporation. This document is supplied as a guide for SOFTIMAGE 3D products. Reasonable care has been taken in preparing the information it contains. However, this document may contain omissions, technical inaccuracies, or typographical errors. Microsoft Corporation does not accept responsibility of any kind for customers' losses due to the use of this document.

Document No. 97539-1296

Printed in Canada.

Table of Contents

CHAPTER ONE

What's New – IRIX and Windows NT	1
Introduction	3
Top Bug Fixes	4
The Interface	7
Modelling	10
Materials and Textures	13
Channels	15
Tools	22
Standalones	50
Custom Effects	52
DKit Information	60
mental ray Programming	69
Licensing between IRIX and Windows NT	71
Working with both IRIX and Windows NT (linktab.ini)	72
Distributed mental ray Rendering between IRIX and Windows NT	77
Known Problems and Limitations	81

CHAPTER TWO

What's New – Windows NT Only	91
The Interface	93
Channels	94
Tools and Standalones	94
Custom Effects	97

CHAPTER THREE

What's New – IRIX Only	99
Channels	101
Custom Shaders	102
New Sounds	102

APPENDIX A

Correct Preset Shaders – Windows NT Only	A-1
Output Shaders	A-3
Volume Shaders	A-8

C H A P T E R O N E

What's New – IRIX and Windows NT®

Introduction

These release notes contain a detailed list of all the new features of SOFTIMAGE® 3D version 3.51 for the IRIX and Windows NT® operating systems. They also contain an outline of some known problems and limitations for version 3.51.

SOFTIMAGE 3D is a sophisticated software package that lets you create the most advanced effects in computer animation. As is always the case with software this complex, there are inevitably some limitations, minor annoyances, and... bugs.

The problems listed in this document are known to Softimage, and we are endeavouring to fix them for future service packs and releases of SOFTIMAGE 3D.

In the meantime, we are letting you know about these problems so that you can avoid them. In many cases, there is a workaround that lets you accomplish your goal another way.

Other Sources of Information

To be sure that you have the latest information, you should also check the following:

- The Softimage World Wide Web site at:
<http://www.softimage.com/>
- The Softimage mailing lists discussion groups. For information about how to subscribe to these lists, see the *Installation Guide*.
- The *Product Support Services Guide*. This guide describes how to receive technical support for all SOFTIMAGE products.

Reporting Problems

If you encounter a problem that is not listed here or anywhere else, see the *Product Support Services Guide* for information about reporting it.

Top Bug Fixes

This is a list of bugs that have been fixed since SOFTIMAGE 3D version 3.5 on IRIX was released. These bug fixes apply to both the Windows NT and IRIX platforms.

1. Could not deactivate Supra keys in capture mode for channels.

Fix: Keyboard mapping conflicts (Supra keys and Swift keys) have been resolved during channel capture sessions.

2. Supra key problems.

Fix: You no longer need to press a Supra key twice to activate it.

3. Database names were unique up to a fixed amount of characters.

Fix: SOFTIMAGE 3D now recognizes the extra characters in similar database names. For example, if you have /usr/DB in your list of databases, you can now use a browser to enter the database located in /usr/DB1 instead of getting a message that you're entering a listed database (/usr/DB).

4. Named selections could not be selected from the Node selection list in the Info Elements dialogue box.

Fix: You can now select named selections from the NSELECTIONS list that appears when you select the Select Node option in the Info ➔ Elements dialogue box.

5. Editing point lights in Multi selection mode.

Fix: You can now edit point lights properly in Multi selection mode, no matter what the selection order of the lights.

6. Camera DOF Near and Far calculations were incorrect.

Fix: The automatic calculation of the camera's depth of field (DOF) now gives more accurate values. The COC (centre of confusion) value is set to 0.001 which corresponds to most cases in photography. You can change the default COC value using the SI_DOF_COC environment variable. For example:

```
setenv SI_DOF_COC 0.002
```

7. Dopesheet: stretching animation gave incorrect keyframing positions.

Fix: You can now predict more accurately where the keyframes will appear after scaling an animation track in the Dopesheet.

8. Dopesheet usability enhancements.

Fix: There have been several improvements to how the Dopesheet handles long scenes:

- Playback rates are now faster.
- You can grab the box handles more easily.
- Fcurve window now updates when moving through frames.

9. When deleting polygons in POL mode in the Matter module, the model was deleted.

Fix: The model is no longer deleted when performing this operation in the Matter module, only its selected polygons.

10. Polygon performance improvements.

Fix: The speed has been improved when loading large polygon mesh objects and when using Effect → Convert to convert polygon mesh objects.

11. Cardinal and weight interpolations gave bad results on NURBS.

Fix: Proper results are now given when using weighted interpolations for shape animation on NURBS objects.

12. Hiding geometry skeleton ceased its effect on envelope.

Fix: The animation of a hidden child object in a hierarchy is now updated and has an effect on the scene.

13. Envelope speed improvements.

Fix: The performance and speed of interactive manipulations has been improved when working with scenes containing large envelopes. There may still be some delays with very large envelopes, but you can now deactivate flexible envelope relations in the Dopesheet.

14. Plotting deleted original keys.

Fix: Plotting an animation on existing function curves no longer deletes the original keyframes that are outside the plotting frame range.

15. When changing a texture file and entering Paint, texture was not updated.

Fix: The texture file information in the Paint dialogue box is now

updated when you change the texture file information: you don't have to exit the Paint dialogue box or select another texture file.

16. mental ray: 2D local texture XZ mapping of straight grid lines was distorted on patch objects.

Fix: Local 2D texture mapping using an XZ projection is now properly displayed when using mental ray. The mental ray renderer handles local 2D texture mapping in XZ the same way as the Softimage renderer.

17. Refractive value ignored by mental ray when raytracing through more than one object which had a refractive index of 1.1.

Fix: The Refractive value is now properly calculated when raytracing using mental ray under these conditions.

18. Particle didn't find updated version of SOFTIMAGE 3D scene.

Fix: If you have a link to a SOFTIMAGE 3D scene and that scene is updated, Particle will now find the latest version of the original scene.

19. Particle couldn't access or read DatabaseSys.rsrc file upon startup.

Fix: When you start Particle, the DatabaseSys.rsrc file is now accessible and being read.

20. Particle's version number was not synchronized with that of SOFTIMAGE 3D.

Fix: The two program's version numbers are now synchronized.

The Interface

Schematic Window

Open/Close

In addition to using the **Open/Close** option in the Schematic window, you can also double-click to open and close hierarchies.

You can now use the Scaling, Rotation, and Translation menu cells even if the **Open/Close** option is on.

Shared Materials and Textures

Shared materials and textures are always located under the first element that uses them. In other words, you can find materials and textures under the first element to which they are assigned. In the previous version, it was more difficult to determine where shared materials and textures were located.

Spreadsheet Window

Shared Elements

You cannot change the prefix and name of a shared element in the Spreadsheet. This is to prevent problems that might occur if other scenes reference the element using the old name.

Writing Information to a File

You can now write the contents of the Spreadsheet to a file. The file is written in ASCII and retains the layout of the original Spreadsheet. To write the contents of the Spreadsheet to disk:

1. Open a Spreadsheet window.
2. Hold down the left Alt and Ctrl keys, and click the **UPDATE** option in the Spreadsheet title bar. The file is automatically written to disk. There is no way to specify the file's name or directory, so it is written to the default location defined by the environment variables. By default, these are as follows:

IRIX: `/usr/tmp`

Windows NT: `\TEMP`

If these directories are not accessible, the file is written to the current directory.

The Spreadsheet uses the following naming convention:

`sprd_printout<n>`

where <n> is a value between 0 and n. This particular naming convention allows you to write several files to disk in one session (`sprd_printout0`, `sprd_printout_1`, etc.).

Note

WIRE Colour (wrcol) Spreadsheet columns are not written to the file.

Shade View Mode

Several new options have been added to the Shade view mode.

- Wireframe objects can now be viewed with their correct wire colour when the **Show Icons** option is selected. The objects' colours can be modified using the **Preferences → Desktop Colours** command.
- When **Show Icons** is selected, the background colour matches that of the other wireframe windows. The background colour can also be modified using the **Preferences → Desktop Colours** command.
- When **Show Icons** is selected, the axes do not appear in the lower-left corner of the window.
- Ghosted objects are displayed in dotted wireframe.
- **Atmosphere → Depth fading** is enabled in the Shade view mode. This produces the same results as the renderer, but can only be viewed in the Perspective window. When the **Show Icons** option is selected, depth fading is disabled.

SI_SHADE_TEXSIZE_LIMIT Environment Variable

For viewing purposes (in Shade view mode or in SI Live), all texture images are automatically scaled down to the next power of 2. For example, if you have a 250 x 100 texture, the image is automatically scaled down in memory to 128 x 64.

The `SI_SHADE_TEXSIZE_LIMIT` environment variable controls the maximum dimension a texture image is scaled down (this does not affect raytraced rendering). If this environment variable is not set, the default upper limit on the texture dimensions is 256, therefore large images (such as 2000 x 1000) are not scaled down to the next power of 2, but to 256 x 256. If there are many large textures in your scene (more than will fit into the hardware memory of your

machine), you should make this number smaller, such as 128 or 64. However, this reduces Shade view texturing accuracy and quality for a smaller memory footprint and faster texture definition.

Texture images that are already smaller than the specified maximum dimension are not scaled up: the environment variable only specifies the maximum dimension. Instead, they are scaled down to the next power of 2 in each dimension. If `SI_SHADE_TEXSIZE_LIMIT` is set to `OFF`, or to a number that is less than or equal to zero, then the old behaviour (without the extra resizing) is used.

Modelling

Local Envelope Assignment

Vertices cannot be assigned to a skeleton root. As a result, local envelopes can only be applied to a skeleton that has more than one valid model (such as SkRt + Skel). If you try to assign a local envelope to a skeleton with only one valid model, the following message is displayed: “For local envelopes, skeleton needs at least two valid models.”

Web Effect

Four new options have been added to the **Effect** → **Web** command in the Model module:

- **Generate Spline(s)**
- **Process Nodes Individually** (including Sibling Splines and Nulls)
- **Create Symmetrical Model**
- **Use Icon Plane**

Generate Spline(s)

When you select the **Generate Spline(s)** option, the system automatically generates a linear curve (spline) through the selected object using its bounding box as its reference. If **Process Nodes Individually** is selected, curves are generated for each node within the hierarchy (that do not possess a sibling curve); otherwise a single curve is generated for the entire hierarchy. Automatic curve generation improves productivity by allowing you to web an entire hierarchy (or character) in a single operation – you don't have to draw and place curves for each node in the hierarchy to be webbed.

Process Nodes Individually

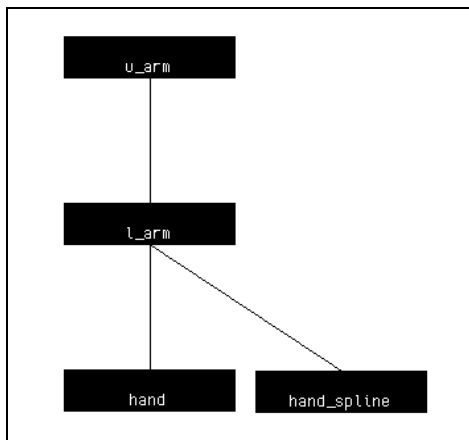
When you select the **Process Nodes Individually** option, the model hierarchy is traversed with each node processed (webbed) individually. One new object is created for each polygon mesh node in the hierarchy.

Sibling Splines

The **Sibling Splines** option allows you to use different curves (splines) for different nodes within the same hierarchy. If **Process Nodes Individually** is selected, the system searches for curves that are

hierarchical siblings of the polygon meshes to be webbed. The siblings must follow this naming convention:

`[NODE_NAME]_spline`



If there is no sibling curve, a curve is either automatically generated (**Generate Splines**), or the originally selected curve is used.

The following steps illustrate how to web an object using a sibling curve:

1. Get or create a polygon mesh object.
2. Create a curve by choosing **Draw → Curve → B-Spline (Non-NURBS)** and position it on the object.
3. Rename the curve following the naming convention and click **Ok**.
4. Open a Schematic window and place the curve at the same hierarchical level as the node to be webbed (its sibling).
5. Choose **Effect → Web**. Select **Process Nodes Individually** and click **Ok**.

You are prompted to first click on the curve and then the model. The Web icon appears and a new webbed model is created.

Sibling Nulls

The **Sibling Nulls** option allows you to exclude certain nodes within a hierarchy from being webbed. If you select **Process Nodes Individually**, the system searches for nulls that are hierarchical

siblings of the polygon meshes to be webbed. If a sibling null is found, the system does not web the corresponding sibling object. The null must use the following naming convention:

`[NODE_NAME]_null`

If no sibling nulls exist, the entire hierarchy is webbed.

Create Symmetric Model

The **Create Symmetric Model** option makes the surface created by the Web effect bilaterally symmetrical. This option uses the global yz plane as its axis of symmetry. You can create a reference point for the effect by manually placing the Web icon at the centre of the object to be webbed (see **Use Icon Plane** below). This gives you greater flexibility and control in converting a detailed model to a surface that represents the exterior of the original object.

To create a bilaterally symmetrical model, the system generates points in V around the model that are equal distances from the centre point and parallel to the symmetry plane.

The following steps describe how to create a bilaterally symmetrical model:

1. Select a model in SOFTIMAGE 3D.
2. Create a curve by choosing **Draw → Curve → B-Spline (Non-NURBS)** and position it on the object.
3. Choose **Effect → Web**. Select **Create Symmetric Model**, **Use Icon Plane**, and **NURBS** as the **Type of Model Generated**. Click Ok.
4. Click on the curve, then the model. The Web icon appears and a new webbed model is created.

When you select the **Use Icon Plane** option, the system uses the Web icon as its point of reference. You can use this option to create bilaterally symmetrical models. When this option is selected, the yz plane of the icon acts as the symmetry plane of the webbed surface.

Materials and Textures

Paint

The **Crop** option has been added to the Paint dialogue box. This option defines the portion of the painted picture file to be used as the texture map. This is the offset and scaling of the object's UV coordinates.

When you select the **Crop** option (or press the Supra key **c**), a red crop box is displayed. You can then crop the picture by moving the box's sides or corners. You can also position the crop box anywhere on the picture by dragging its cross-hair cursor. As you modify the size and position of the crop box, the UV coordinates automatically adjust to match the location and dimensions of the newly cropped area.

Note

You cannot use another Paint tool while the **Crop** option is selected.

Palette

Normalized Option

The **Normalized** option has been added to the Palette dialogue box, which is accessible from the **Material**, **Texture**, **Ambience**, and **Light** menu cells. This new option lets you define the range of the RGBA text boxes allowing you to define colour selection in two different ways. When the **Normalized** option is selected, the RGBA text box values range from 0.0 to 1.0. When **Normalized** is not selected, the RGBA text box values range from 0 to 255.

This new integer-based colour selection tool can help simplify the palette manipulation process. It can also be very useful when the number of bits per colour channel has been modified using the Colour Reduction and Paint commands, resulting in colours that do not fall within the “normalized” ranges.

By default, the **Normalized** option is active. To deactivate it, click its button and click Ok.

Note

The values entered in the RGBA text boxes must fall within the range specified. If you enter a value that falls outside that range, the system automatically uses the current minimum and maximum range values.

EDIT options

Three new options have been added to the EDIT menu in the Palette dialogue box. These options provide you with more control over Palette selection.

- **Select All** lets you select all the colours in the palette. You can also press Ctrl-a to do this.
- **Toggle Select** lets you switch between colours that are selected and those that are not selected. You can also press Ctrl-t to do this.
- **Unselect All** lets you deselect all the colours in the palette.

The Supra key for the **Paste After** option, also in the EDIT menu, has been changed from Ctrl-a to Ctrl-f.

2D Textures

Cropping a Sequence

A sequence of image files can be cropped in much the same manner as a single image, but there is one important distinction: after you manipulate the cropping rectangle that appears around the picture file, you must click Next, Previous, Paint, or Ok to accept the cropping values for the entire sequence. If you do not click one of these buttons, the cropping values will apply only to the first frame of the sequence.

Once you have accepted these new cropped values, they become the default for any new texture you apply to your object.

Channels

sampleCompute driver

This driver is provided as sample source code for those who want to develop their own channel drivers. It applies a sample computation to degrees of freedom. See the *Channel Developer's Guide* HTML file on the On-line Documentation CD for more information.

Take Recording

After you have captured your motion data using the Take Recording feature in either SOFTIMAGE 3D or the SOFTIMAGE Live viewer, you can use the ASCII or binary files that are created on an IRIX or Windows NT system. These files are interchangeable between the two operating systems.

In addition, by right clicking in the middle of a take run, you can exit “take” mode, which automatically opens the Take Load dialogue box.

Remote Drivers

Channel drivers can be started and operated across machine architecture boundaries. For example, you can now do the following:

- SOFTIMAGE 3D Windows NT or SOFTIMAGE Live launches the remote driver on another Windows NT machine or an IRIX machine.
- SOFTIMAGE 3D IRIX or SOFTIMAGE Live launches the remote driver on another IRIX machine or a Windows NT machine.

Environment Variables for Remote Drivers

There are two environment variables which may be particularly useful when running remote drivers.

The `SI_DRIVER_TIMEOUT` environment variable is generally used to limit the amount of time that SOFTIMAGE 3D waits for a driver to start. In the case of remote drivers, especially across architecture boundaries, more startup time may be needed. You may want to increase this value from the default of 7 seconds.

If you set the environment variable `SI_DRIVER_REMOTE_USER`, remote drivers are launched on remote machines with the specified user name (otherwise, the local user name is used). In either case, the specified user must be able to run commands on the remote machine without being prompted for a password.

For example, if `SI_DRIVER_REMOTE_USER` is set, you must be able to use **rsh** to connect to the remote machine using the remote user name and execute a command:

```
rsh remote_machine -l remote_username command
```

If `SI_DRIVER_REMOTE_USER` is not set:

```
rsh remote_machine command
```

Using Remote Drivers on Windows NT

To make use of remote drivers on Windows NT, you must have some sort of remote shell mechanism set up and installed correctly to allow remote process execution between machines. Typically, you'll need to do the following:

1. Make sure that all the machines you want to spawn drivers on/ from are on the same network, can access each other, and are running networking software.
2. Install an rsh service/daemon on your Windows NT machine, along with an rsh client. A free service called **rshsvc** is available in the Windows NT resource kit and an rsh client program is shipped along with Windows NT.

You should be able to use any of the several available commercial solutions that support the rsh protocol. Check with your vendor for details.

Note

If you only want to launch remote drivers on an IRIX box from your Windows NT machine, you need only to have the rsh client program. If, however, you want to spawn remote drivers on another Windows NT machine or on your Windows NT machine from an IRIX one, you need to install an rsh service.

3. Set up your network and **rhosts** files (see the relevant networking documentation) properly so that the machines have access to each other. Check to see if you have the correct access to launch a driver on a remote machine by typing:

```
rsh <target_machine> dir
```

No password prompt should appear for the test to work.

4. Broadcast and listen drivers (see next section) do not require the setup of rsh clients or servers. As such, you can make use of this mechanism in cases where you do not have rsh software

available. To do so, start the broadcast driver on the target machine, and run a listen driver (or a driver in listen mode) on the local machine. This allows you to get data from the target machine's broadcaster.

Broadcast and Listen Drivers

Broadcast/listen drivers are now a supported feature of SOFTIMAGE 3D on both the IRIX and Windows NT platforms (they do not require the presence of any rsh remote process execution software, unlike remote drivers).

Any driver can run in broadcast mode, whereby the driver communicates its data to any number of sessions of SOFTIMAGE 3D or the SOFTIMAGE Live viewer running concurrently. The other sessions "listen in" on the broadcaster using either the special listen channel driver or any other driver running in listen mode.



Only input drivers currently work as broadcast drivers, and the only commands that are broadcast are SetDofTime, SetDofValue, and the GetOp/SendData pair.

This new feature allows you to do the following:

- Plug a device into a machine, use a channel driver on the machine, and see the results on another session of SOFTIMAGE 3D on either Windows NT or IRIX, as well as in the local SOFTIMAGE Live viewer. Many devices allow only one driver to read them at a time, so the broadcast mechanism allows several sessions to access the data.
- Create scenarios using two machines, where devices on those machines control a character on both the local and remote machines. This allows you to create interaction between characters using channel drivers. For example, you could create a virtual fight game between two characters – one character originating on Windows NT and one originating on IRIX. Both characters, however, appear in the viewers of both machines. A joystick on the first machine controls one character in both the local and remote viewer, and a joystick on the other machine controls another character in both viewers!
- Run a driver on one machine and access the data on other machines.
- If you do not want to automatically use the remote driver feature to

start up a driver on another machine, you can start the driver “manually” in broadcast mode on that machine, and simply listen in on the local machine.

There are two main ways in which the broadcast/listen feature can be used:

- Manual/standalone: the broadcast driver is started manually from the command line and the special listen channel driver is used.
- Automatic/online: a special “registration” file is used which allows the driver to broadcast from the context of a SOFTIMAGE 3D or viewer channel run. It also supplies the local session with the appropriate data. The machine which is listening in uses the same driver that is used when running locally, but it is switched to the new “Listen” mode. Because of the registration file, it knows automatically how to connect to the broadcaster. All you need to do is specify the broadcast machine name.

The following two sections (*Manual/Standalone* and *Automatic/Online*) describe these two main modes in more detail.



All drivers can run in these new modes without any prior modifications. A complete list of the possible driver modes can be obtained by typing `<driver> -SI_RUN -h`

Manual/Standalone

In the manual/standalone mode of operation, you start any input channel driver from the command line manually. You must then use the special listen channel driver in the sessions of SOFTIMAGE 3D or the SOFTIMAGE Live viewer where you want to listen in.

To use broadcast/listen drivers:

1. Run the input driver you want to broadcast with by typing:

```
mouse -SI_RUN -mode BROADCAST <port_number>
```

where `<port_number>` is a TCP port that falls within the user range (such as 2000). This should be different for each broadcast driver on the same machine. The driver then waits for listen drivers running within SOFTIMAGE 3D or the SOFTIMAGE Live viewer to connect.

2. To connect to this broadcast driver from a session in SOFTIMAGE 3D, load the listen channel driver. Adjust the number of DOFs to match that of the broadcast driver

(optional), type in the port number and the machine name (it can be on a different machine than the one you are on), then use that driver as you would any normal channel driver.

When you start any kind of channel run, the listen driver listens in to the broadcast driver, and you get the broadcast driver's data. Multiple instances of SOFTIMAGE 3D can be listening to the same broadcast driver at the same time.

Note

It does not matter in which order you start the listen and broadcast drivers. If you start a listen first, and the broadcast driver isn't running, the channel run proceeds as usual, periodically checking to see if the broadcaster has started. The same features work from the SOFTIMAGE Live viewer.

You can try doing loopback listens/broadcasts between two instances of SOFTIMAGE 3D.

Since SOFTIMAGE 3D is not running to supply any custom data to the broadcast driver (which was started as a standalone from the command line/prompt), if the driver is expecting custom data, it won't get it.

You can work around this by:

- Using only **.drv** broadcast drivers, which don't have custom dialogue boxes.
- Piping the data that a **.cus** driver would need as standard into the broadcast driver. See a **.chn** file in the **SETUP_CHANNELS** chapter of the database for an example of the format that a driver expects. You should pass the information from the **_DATA** until the **_END** for a given driver. For example, for the keyboard driver, you could have a small file called **keyboard_feeder** which contains:

```
KEY_7 1
KEY_8 1
G_ATTACK 1.000000
G_DELAY 1.000000
G_DECAY 1.000000
_END
```

You can then start the keyboard driver in broadcast mode by typing:

```
keyboard -SI_RUN -mode BROADCAST 2000 < keyboard_feeder
```

Note

Ports 2000 to 2050 are “reserved” for broadcast drivers and are not used by any other drivers running in remote mode. You are free to use any port in the user range (see networking documentation), but be aware that remote drivers may use ports falling outside the 2000 to 2050 range.

Map File Option

The listen driver has the option of using a map file to map degrees of freedom names to degrees of freedom IDs. This allows you to view the names of the connections in the Connection Setup dialogue box, which lets you see to what each listen driver degree of freedom refers.

You can create a map file like the following:

```
# Example map file
[# Ignore comment lines starting with #, and any blank
lines]
[Dummy # names of the original driver creating the
mapping]
4 # Number of dofs
one
two
three
four
```

This map file creates four degrees of freedom for the listen driver named `dummy_one`, `dummy_two`, `dummy_three`, and `dummy_four`.

Automatic/Online

A driver can be started and run in normal mode (supplying data to the local session of SOFTIMAGE 3D or the SOFTIMAGE Live viewer that started it), and simultaneously broadcast data to any other sessions that want to listen in.

Broadcasting

To broadcast, you must place a special registration file in the directory pointed to by your `SI_CHNLDRIVER` environment variable. This file is named `<driver_name>.brd` (for example, for the mouse driver it would be `mouse.brd`) and looks similar to this:

```
SOFT-BROADCAST <driver_name>
on
2001
```

For example, say that `<driver_name>` is “keyboard.” When you start

a channel run, if a **keyboard.brd** file exists, the keyboard driver is started as a normal channel driver, but also listens periodically for connections from other sessions that want to listen in. The second line in the **.brd** file must state “on.”

Listening

To listen in on other sessions of SOFTIMAGE 3D or the SOFTIMAGE Live viewer, you must also set up a keyboard driver to do a channel run. The keyboard driver should have the same custom setup as the one awaiting connections as mentioned previously. In addition, make sure that the same **.brd** file is installed on the machines in question, especially if the **SI_CHNLDriver** environment variable differs between the various machines.

To do this, go into Channel Setup, select the keyboard driver, click Options, and switch the instance of the driver to Listen mode. You also need to specify the name of the machines from which the driver will listen.

When you start a channel run, the driver tries to connect with the first driver. If it can't do so initially, it will keep trying until it is successful. This occurs whenever the broadcast driver is started later than the listener.

More about the .brd File

The first line of the **.brd** file has the magic identifier followed by the driver name.

The second line contains the string “on” or “off.” If set to “off,” the channel drivers run as they normally would and not in local and broadcast mode. However, any driver in listen mode knows how to communicate with broadcast drivers of that name running on another machine.

The third line contains a number which should be unique for each driver that has a **.brd** file. It is the TCP port that is used for communication. As mentioned earlier, the range 2000 to 2050 is reserved by channels for broadcast drivers. This number allows the channel driver running in broadcast and local mode to speak to any other machines listening in. It also allows drivers running in listen mode to establish a connection between machines. Therefore, both broadcast/local and listen drivers need to have access to the registration file so they can agree upon the port number.

Tools

Video → Abekas/Accom

The **Starting Location** text box and the **Linear** and **Time Code** buttons in the Abekas/Accom Record and Grab dialogue boxes have been replaced by a single **Time Code** text box. You don't need to toggle between Linear and Time Code anymore because SOFTIMAGE 3D recognizes the format by the way the numbers are entered. As a result, you can enter the time code or frame number directly into the **Time Code** text box. As in previous versions, the numbers entered into this text box define the starting location of the video record or grab. If you want to use frame numbers as your starting point, type the frame number into the text box (such as 30). If you wish to use time code, enter the time code in this format **mm:ss:frame** (such as 00:01:20 for 1 second, 20 frames).



In **Video → Accom** and **Abekas**, Accom works on both the Ethernet and SCSI connections, but Abekas works only for an Ethernet connection.

To change the default device name in the Abekas dialogue box from Abekas to the actual DDR host name, do the following:

1. Make sure the environment variable `SI_IODEVICE_RSRC` is pointing to the path of the configuration file `SI_IODDevice.cfg` (the `\rsrc` directory), such as `C:\SOFT3D_3.51\3D\bin\rsrc`.
2. Make sure the environment variable `SI_IODEVICE_PATH` is pointing to the IO device DLLs, such as:

`C:\SOFT3D_3.51\3D\dso\SI_IODDevice.`

3. In the Devices section of `SI_IODDevice.cfg`, the line:

`Abekas = 7,1 Abekas (ethernet)`

has the number 7 pointing to the transport device in the Transport Device section and the number 1 corresponding to the video board. If necessary, edit this line as follows:

```
SI_TransportDevice7=Abekas Disk Recorder,  
SI_AbekasEthernet, SI_AbekasEthernet.dll, diskus,  
525, 1.0.
```

where `diskus` refers to the host name (make sure that you can “ping” the device host name).

Note

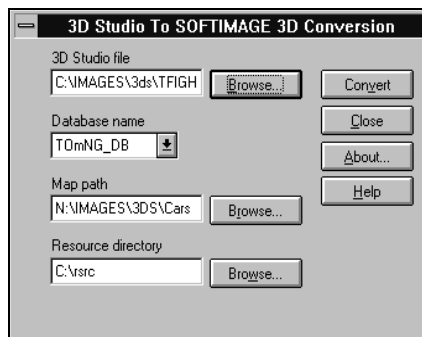
The environment variables can be added or modified in the SETENV.BAT file. This file is automatically created when SOFTIMAGE 3D Windows NT 3.51 is installed, and can be opened from the SOFTIMAGE 3D Windows NT 3.51 window in the Program Manager.

Get → 3DS2Soft Is Now Import → Objects → 3DStudio

The **Get → 3DS2Soft** command has been replaced by the **Import → Objects → 3DStudio** command in the Tools module. This command converts files from 3D Studio (3DS version 3.0) to SOFTIMAGE 3D format. It converts an extensive subset of geometry, lights, cameras, materials, 2D textures, and animation.

The files written by 3D Studio version 2.0 contain information not present in later versions. This information is ignored, resulting in defaults for certain parameters.

When you choose the **Import → Objects → 3DStudio** command, the following dialogue box is displayed:



Parameters

3D Studio File

Enter the file that is to be converted or click the **Browse** button to navigate to any directory and select a file with the extension .3DS for conversion. The directory last used is retained and is used as a starting point each time the application is run (if the directory is not available, the current directory is used).

Database Name

This drop-down menu displays the names of the databases specified

in the **DatabaseDir.rsrc** file. Select the name of the file to which you want the converted model written. The database last selected is automatically selected the next time the application is run.

Map Path

Enter the directory that contains the texture maps referenced by the converted model.

You can also use the **Browse** button to navigate to the map directory. This directory is retained and used by the application the next time it is run (if the directory is not available, the current directory is used).

Resource Directory

Enter the directory that contains the **Relations.rsrc** file. You can also select the **Browse** button to navigate to the desired directory. This directory is retained and used when the application is next run (if the directory is not available, the current directory is used).

Convert

Initiates the conversion process. It is ghosted until a 3D Studio file and a valid database name have been chosen. Status information relating to the progress of the conversion is displayed at the bottom of the window. When the converter has finished converting a file, the dialogue box remains active and the status line displays the message Ready. If an error occurs during the conversion process, a message box is displayed and you must click Ok to continue the process.

About

Displays version and development information.

Help

Displays a help file which describes each of the parameters in the dialogue box.

Name Mangling

When the object converter performs a conversion between an item (such as light, camera, material, or polygon mesh) in one format to an equivalent item in another format, it attempts to match the name of the converted item as closely as possible to that of the original item. Since permitted name lengths and format differ between the SOFTIMAGE 3D and 3D Studio environments, and since a name

may already be in use, the object converter may be forced to alter names during conversion. This process is termed *name mangling*.

3D Studio Import Conversion

The following describes the information that is converted:

- Ambient Light: This is converted to **Atmosphere → Ambience** in SOFTIMAGE 3D. RGB colour maps are converted directly to RGB colour.
- Global Shadow Parameters: These are applied to each light, if local settings are not present for that light.
- Fog/Depth Cueing: This is converted to **Atmosphere → Depth Fading**. The 3D Studio initial % setting is ignored (no equivalent), the start distance, end distance and fog RGB colour are mapped directly, and the final % setting maps to Transparency.
- Background: 3D Studio backgrounds have no direct equivalent in SOFTIMAGE 3D and are not converted.
- Viewport Information: Viewport information is not compatible between 3D Studio and SOFTIMAGE 3D and is not converted.
- Cameras: The first camera encountered in the 3D Studio file is read and converted to the camera in SOFTIMAGE 3D. Camera position, interest, hidden status, near plane clipping, and roll are converted directly. Far plane clipping is not available. The remaining camera parameters are set to the defaults of SOFTIMAGE 3D

Note

If no cameras exist in the 3D Studio file, a camera is created with the SOFTIMAGE 3D default parameters.

Any hierarchical relationship defined in the 3D Studio file between cameras and other objects is ignored (the SOFTIMAGE 3D camera cannot be part of a hierarchy).

Lights

- 3D Studio omni lights are converted to SOFTIMAGE 3D point lights of the same colour and position with no shadow set. If attenuation is enabled for a light, the SOFTIMAGE 3D falloff parameters are set according to the 3D Studio file range values specified for that light. If objects are included from the light, the light is set as a selective and exclusive light. The excluded objects are associated with the light. The umbra, penumbra, and shadow parameters are set to SOFTIMAGE 3D defaults.

- 3D Studio spot lights are converted to SOFTIMAGE 3D spot lights of the same colour, position, interest, and cone/spread angles. If shadows are enabled for the light, they are set to either raytraced or depth-mapped, accordingly. Depth-mapped shadow parameters are set for map resolution and filter size, when applicable. If attenuation is enabled for a light, the SOFTIMAGE 3D falloff parameters are set according to the 3D Studio file range values specified for that light. If objects are excluded from the light, the light is set as a selective and exclusive light. The excluded objects are associated with the light. The umbra and penumbra parameters are set to SOFTIMAGE 3D defaults.
- 3D Studio rectangular and projector spot lights are converted to normal spot lights in SOFTIMAGE 3D, as previously described.

Note Any hierarchical relationship defined in the 3D Studio file between lights and other objects is ignored.

Materials

Each valid material in the 3D Studio file is converted to one material and zero or more associated 2D textures. The name of the material definition is the same name used in the 3D Studio file. 2D textures associated with a material use names derived from the material name. Since SOFTIMAGE 3D does not support independent mark maps, these are not converted from 3D Studio materials.

Materials are converted as follows:

- Wire-frame and flat-shaded material types are converted to Constant-shaded materials.
- Gouraud, Phong, and Metal materials are converted to Phong-shaded materials.
- For all material types other than Phong and Metal, the RGB values for ambient, diffuse, and specular components are converted directly to their equivalents in SOFTIMAGE 3D.
- For Phong and Metal materials, the ambient and diffuse components are converted directly to their equivalents in SOFTIMAGE 3D, while the specular component colour intensity is attenuated by the 3D Studio Shininess Strength values. The 3D Studio Shininess value is scaled and converted to the Specular Decay setting.

- The 3D Studio Transparency value is scaled and converted to the Transparency setting in SOFTIMAGE 3D.
- In 3D Studio, the Phong Soften, Transparency Falloff, Reflection Blur, Self-Illumination and Face-Map settings have no equivalent in SOFTIMAGE 3D and are not converted.

2D Textures

Due to the differences between 3D Studio and SOFTIMAGE 3D in the way that materials are defined and assigned to objects, one 2D Texture definition is created for each distinct mapping file/mapping assignment combination in 3D Studio.

- 3D Studio material map definitions do not correspond exactly with those in SOFTIMAGE 3D. The default settings for 2D textures may be modified, as described in the sections for each map type.
- The Picture Name field is set to the base name of the corresponding 3D Studio map file.
- The Animation setting is determined by the type of file used in the corresponding 3D Studio map. If the map file type is .FLI or .FLC, the animation type is set to Sequence; if the map file type is .IFL, the animation type is set to Script; otherwise the animation type is set to Static.
- The UV Scale, Offset, and Repeats settings are assigned the corresponding values from the 3D Studio map and object. Note that if the correct parametric information for mapping is not present in the 3D Studio file (for example, if the file was created by an earlier version of 3D Studio, or if objects are deformed after mapping is assigned), the mapping conversion cannot be performed accurately.
- The Tiling mode is set to “non-alternating” for both u and v, unless the Mirror option is set for the 3D Studio bitmap, in which case the tiling is set to alternating for both u and v.
- The Swap UV option is deselected.
- The Pixel Interpolation option is selected.
- The reflection Map Rotation field and Intensity slider are set to 0.0.
- The Mapping Method is set to XY, Cylindrical, or Spherical, depending on the mapping method used for this material’s application in 3D Studio.

- The Map Component selection is set to RGB Intensity and the Transparency, Reflectivity, and Roughness sliders are set to 0.0.
- All Cropping values are set to 0.0.
- The Blending option is set to Without Mask, and the Overall Blending, Ambient, Diffuse, and Specular sliders are set to 0.0.
- For Texture maps (Texture1, Texture2), a 2D texture is generated with the Blending control set to Alpha Channel Mask, unless the 3D Studio texture has the Ignore Alpha button set, in which case the Blending control is set to Without Mask. The Overall Blending slider is set to the scaled value from the 3D Studio Amount slider for the appropriate map and the Ambient, Diffuse, and Specular sliders are all set to 1.0.
- For the Specular map, a 2D texture is generated with the Blending control set to Alpha Channel Mask, unless the 3D Studio texture has the Ignore Alpha button set, in which case the Blending control is set to Without Mask. The Overall Blending slider is set to the scaled value from the 3D Studio Amount slider for that map and the Specular sliders are set to 1.0.
- For the Reflection map, a 2D texture is generated with the Reflection map intensity set to the scaled value from the 3D Studio Amount slider for the Reflection map.
- For the Opacity Map, the Roughness slider is set to the Amount slider for the Opacity map.
- For the Bump Map, the roughness slider is set to the scaled value from the 3D Studio Amount slider for the Bump map.
- The 3D Studio Shininess Map, Self-Illumination maps, and SXP (procedural) textures cannot be converted to SOFTIMAGE 3D.

Object Geometry

- Each polygon mesh object from the 3D Studio file is converted to a polygon mesh object in SOFTIMAGE 3D and assigned the same name as the 3D Studio object. Automatic Discontinuity is set for each polygon mesh, using the SOFTIMAGE 3D default angle of 60 degrees. This setting is overridden using the discontinuity edge flag to simulate the effect of 3D Studio's smoothing groups.
- Materials created are associated with the appropriate polygon mesh objects.

- Any hierarchical relationship defined in the 3D Studio file for objects is recreated in SOFTIMAGE 3D. 3D Studio instance objects are expanded in SOFTIMAGE 3D.

Animation

- 3D Studio animations for polygon mesh objects, lights, and cameras using scale, rotation, and translation (SRT) are converted directly to their equivalents in SOFTIMAGE 3D. Keyframes are generated on a one-to-one basis, with function curve shapes being determined by keyframe settings. Additional animated parameters for lights (colour, hotspot, and falloff) and cameras (roll and field of view) are similarly converted.
- 3D Studio morphing of objects is converted by using SOFTIMAGE 3D shape animation techniques (such as creating a series of different key shapes with the same number of vertices and assigning them to the appropriate keyframes).
- 3D Studio material morphing is converted by applying the converted material definition to the appropriate object/keyframe combination.
- Pivot points for objects in 3D Studio are converted to object centres in SOFTIMAGE 3D.
- 3D Studio hide keys are converted to an equivalent hidden function curve in SOFTIMAGE 3D.
- Dummy objects in 3D Studio are converted to Null objects in SOFTIMAGE 3D.

Export → Objects → direct3DExport

The **Export → Objects → direct3DExport** command is written with SOFTIMAGE's DKit and allows you to export polygon mesh objects from SOFTIMAGE in direct3D XOF file format version 2.0 (a binary file identified by an .xof extension). This operation manipulates the specified SOFTIMAGE 3D models (geometry, materials and textures) to properly express the data in the XOF format. The command does not export animation data.

Note

The **direct3DExport** command is available for SOFTIMAGE 3D version 3.0 and later on the following platforms: SGI (IRIX), PC (Intel), PC (Alpha), and PC (MIPS).

Installing direct3D Export

Installing **direct3DExport** is identical to standard custom effects. It should be located in the **tools** directory (both the **.cus** and the executable files). For example:

- IRIX: **/usr/softimage/custom/tools**
- Windows NT: **c:\softimage\custom\tools**

You must also update your **SI_CUSTOM_TOOLS** environment variable to indicate the location of the appropriate directory:

- IRIX: **setenv SI_CUSTOM_TOOLS /usr/softimage/custom/tools**
- Windows NT: **set SI_CUSTOM_TOOLS
c:\softimage\customs\tools**

On IRIX, it is recommended that you put this environment variable definition in the **.softimage** file.

Conversion Process

Since the XOF format does not support hierarchies, the converter needs to either merge all objects in the SOFTIMAGE 3D hierarchy into a single model or generate multiple XOF files (one per model in the hierarchy). The **Merge Polygon Meshes** option in the **direct3D Export** dialogue box allows you to specify which approach will be used. If you select the **Merge Polygon Meshes** option, a user-specified file name is used to save the converted data on disk; otherwise, the **.xof** files are named after the objects themselves.

Scaling and Viewing Problems

By default, SOFTIMAGE 3D uses non-classical scaling transformations, which is a different approach than most 3D systems (you can toggle between classical and non-classical modes by choosing the **Preferences → Classic Scaling** command). The difference between the two approaches is the order in which the global transformations are applied on a model within a hierarchy:

Non-classical:

$$S_{model} \times S_{parent} \times R_{model} \times T_{model} \times R_{parent} \times T_{parent}$$

Classical:

$$S_{model} \times R_{model} \times T_{model} \times S_{parent} \times R_{parent} \times T_{parent}$$

Unfortunately, there is no automatic way of converting non-classical transformations to classical ones. This means that there will be a discrepancy between the SOFTIMAGE 3D and direct3D scenes, unless the SOFTIMAGE 3D models use “identity” scaling (that is, 1.0, 1.0, and 1.0).

To avoid this problem, you can select the **Freeze Transformations** option in the direct3D Export dialogue box. This option applies the transformations on the model vertices and then resets the transformations to the default setting. This option should not be used, however, if you want to keep the original model definition and its transformations as two distinct pieces of information.

Another option is to use the **Effect → Freeze → Scaling** command in SOFTIMAGE 3D on the hierarchy before exporting it to direct3D.

Materials and Textures

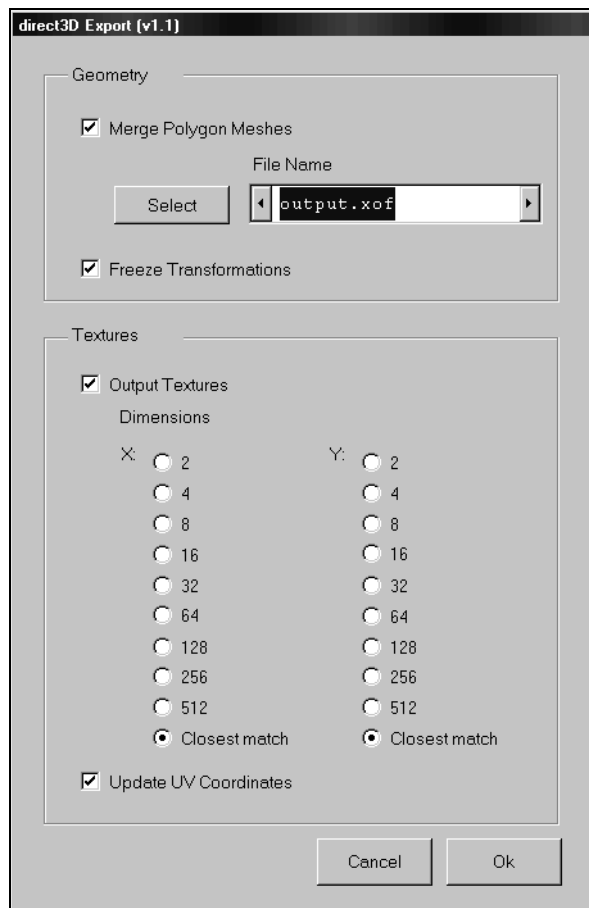
SOFTIMAGE 3D allows you to apply multiple materials (one global and multiple locals) to an object. Each material can also have multiple textures applied to it using various projection methods: XY, XZ, YZ, cylindrical, spherical, and UV-mapped. The direct3D effect v2.0, however, only supports one texture per polygon. Unfortunately, the converter does not actually blend the multiple textures applied to the polygon into a single texture. It simply considers the first global 2D texture (if any), then it looks at the first 2D texture on each local material. In the direct3D Export dialogue box, you can specify texture dimensions either as absolute values in X and Y (they can be different) or as values equal to the closest upper-bound power of 2 (**Closest match**). Texture conversion does not involve bilinear interpolation for smoothing: it uses the closest neighbour texture sampling technique instead.

Limitations

A current limitation of the converter is that cylindrically or spherically mapped textures in SOFTIMAGE 3D will not be exported properly for polygons that are under the texture wrap. A work-around to this problem is to convert the texture projection method to UV in SOFTIMAGE 3D by choosing **Txt_Oper → Info UV Coord** in the Matter module, clicking OK, and specifying a name for the new texture map, required because SOFTIMAGE 3D expands the texture map horizontally (or vertically if the **Swap UV** option is selected).

Using direct3D Export

When you choose the **Export** → **Objects** → **direct3DExport** command in the Tools module, the direct3D Export dialogue box is displayed.



The direct3D Export dialogue box is organized in two main sections: Geometry and Textures. The default values for each option in these section are set to those most commonly used.

Geometry

- **Merge Polygon Meshes** merges all the SOFTIMAGE 3D objects contained in a hierarchy are merged into a single file. In this case, the file name is taken from what is specified in the **File Name** text box. When selected, each object is saved separately in its own file,

which is named after the object names in SOFTIMAGE 3D (with an **.xof** extension).

- **File Name** specifies the name of the exported direct3D files. Type the file name in this text box or click the **Select** button to open a file browser and select an existing file name.
- **Freeze Transformations** applies global transformations to every vertex of each polygon mesh object and the object transformations are reset to their default values (that is, identity). If this option is deselected, the objects and their transformations remain unmodified.

Textures

- **Output Textures** converts the original SOFTIMAGE 3D **.pic** texture image files to **.ppm** format and stores them in the same location as the **.xof** files. If this option is deselected, the image file conversion does not occur, but the texture file reference is stored within the object data structure in the **.xof** file.
- **Dimensions** allows you to specify the dimensions of output maps in x and y. You can choose any of the available powers of 2 (2, 4, 8, 16, 32, 64, 128, 256, or 512) or use the **Closest Match** option (the default) to prompt the system to use the closest upper-bound power of 2.
- **Update UV Coordinates** composes the texture's UV coordinates from the texture projection characteristics in SOFTIMAGE 3D and stores the information with its respective object. If this option is deselected, the UV coordinates of the objects are not updated, so only the UV-mapped textures defined as such within SOFTIMAGE 3D appear correctly inside direct3D.

Tips and Tricks

- Don't forget to choose **Effect → Freeze → Scaling** in the Model module or use the **Freeze Transformations** option in the direct3D Export dialogue box if you have scaled your SOFTIMAGE 3D models using the non-classical scaling mode.

Note

The effect is applied to the whole scene, not just selected objects. It is write only, so it doesn't modify the SOFTIMAGE 3D scene in any way.

- Convert cylindrically or spherically mapped textures to UV by choosing **Txt_Oper → Info UV Coord** in the Matter module before

exporting to the .xof format. This ensures correct texture mapping for the polygons under the texture projection wrap.

- Use the default options because they usually give the best results. However, you might want to change the texture slice dimensions to lower resolutions to save disk space and system memory when working with direct3D.

Export → Objects → direct3DXExport

The **Export → Objects → direct3DXExport** command is written with SOFTIMAGE's SAAPHIRE and lets you export polygon mesh objects from SOFTIMAGE 3D to direct3D X file format (an ASCII file identified with an .x extension). This operation manipulates the specified SOFTIMAGE 3D models (geometry, hierarchies, materials and textures, UV Coordinates, and animation).



The **direct3D X Export** command is available for SOFTIMAGE 3D version 3.51 on the following platforms: SGI (IRIX), PC (Intel), PC (Alpha), and PC (MIPS).

Installing direct3D X Export

Installing the **direct3D X Export** command is identical to standard custom effects and should be located in the **tools** directory (both the .cus and the executable files). For example:

- IRIX: /usr/softimage/custom/tools
- Windows NT: c:\softimage\custom\tools

You must also update your SI_CUSTOM_TOOLS environment variable to indicate the location of the appropriate directory:

- IRIX: setenv SI_CUSTOM_TOOLS /usr/softimage/custom/tools
- Windows NT: set SI_CUSTOM_TOOLS
c:\softimage\customs\tools

On IRIX, it is recommended that you put this environment variable definition in the .softimage file.

Materials and Textures

The **direct3D X Export** command exports local and global materials and textures. If a global material is assigned to an object, every polygon that does not have a local material is assigned the global material (and texture).

Textures

The **direct3D X Export** command exports UV Coordinates. Extra processing is done on the object to duplicate the vertex with UV discontinuities because the .x file format accepts only one coordinate pair per vertex. This modifies the geometry of the object, but it can be deselected in the Effects dialogue box.

If a polygon mesh has a projected texture (not UV mapped), the **direct3D X Export** command assigns proper UV Coordinates to the object (or its descendant that doesn't have a material). However, if the projection mode is either cylindrical or spherical, the polygons with texture seams are mapped incorrectly (UV wrapping is not yet supported).

Note

To convert cylindrically or spherically mapped textures to UV mapping, choose the **Txt_Oper → Info UV Coord. (local or global)** command in the Matter module.

Animation

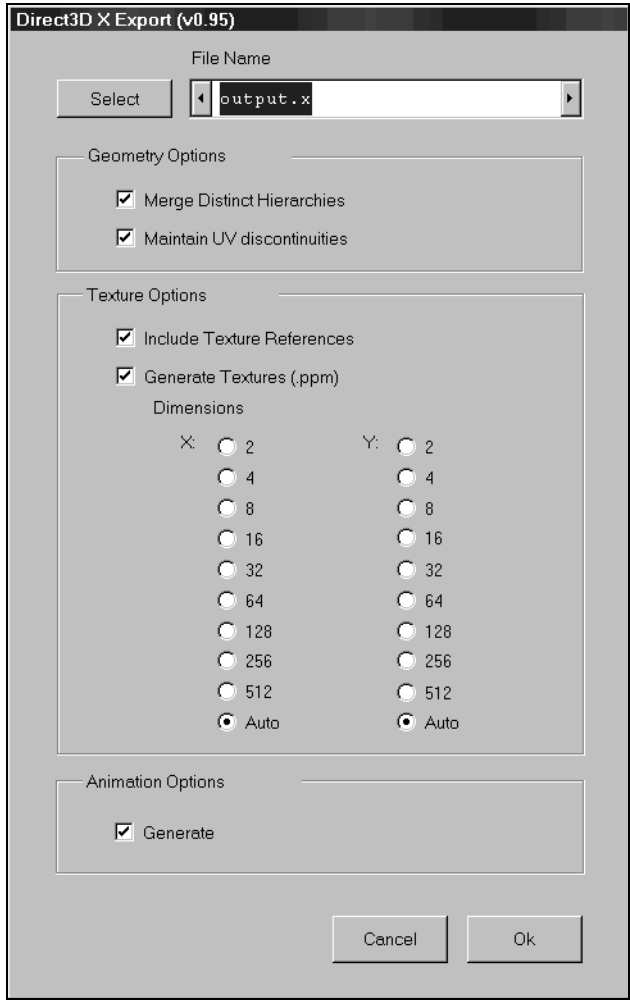
The **direct3D X Export** command exports animation (translation, rotation, and scaling) for every model in the hierarchy and all frames specified in the playback settings (such as frame start and frame end).

Note

The **direct3D X Export** command does not convert geometry animation, such as envelopes, shape/cluster animation, etc.

Using direct3D X Export

When you choose the **Export → Objects → direct3D X Export** command in the Tools module, the following dialogue box is displayed.



The direct3D X Export dialogue box is divided into three main areas: Geometry, Texture, and Animation options. The default values for each option are set to those most commonly used.

File Name

Enter the file name in this text box or click the **Select** button to select a file. If a file already exists, it is overwritten.

Geometry Options

- **Merge Distinct Hierarchies** merges all the hierarchies under a single root in the .x file (a new empty polygon mesh object is not automatically created: it must already exist). If you want to cut and paste entire hierarchies from the exported file to another file, deselect this option. Non-polygonal models are replaced with an empty polygon mesh object, such as an object with one polygon with three points at (0,0,0).

Note

You cannot load a file with several hierarchies with the current Direct 3D Viewer.

- **Maintain UV Coordinates** allows geometry alteration (adding a vertex) to keep the UV discontinuities of the object (**direct3D X Export** allows only one UV coordinate per vertex). If you deselect this option, the geometry is not affected, but the UV mapping may be distorted if there are discontinuities.

Texture Options

- **Include Texture References** includes references to the texture files from the materials.
- **Generate Textures (.ppm)** converts the .pic bitmap files to their .ppm equivalent. To reduce or enlarge the bitmap, select the dimensions in X and Y. If **Auto** is selected (it is by default), the next highest valid power of 2 value is used (bitmap dimensions must be power of 2 values).

Animation Options

- **Generate** includes the animation in the file.

Tips and Tricks

- When creating a model, make sure that there are no “invalid” values, such as infinite normals, etc. Such values make the exported file unreadable by the viewer.
- Make sure that the objects’ scaling values are positive. If the scaling values are negative and you select a polygon in the direct3d viewer, polygons on the other side of the object are selected. The mouse movements and model rotations applied through a hierarchy will also be inverted if negative values are used.

- If you want to view the object with the direct3d viewer, use **Open Frame** to load the object without animation. **Open Mesh** fails even if there is only one polygon mesh because the polygon meshes are encapsulated in Frame templates. Use **Open Animation** to view the meshes with animation. To make the direct3d view output look more like SOFTIMAGE 3D, select the **Gouraud**, **RGB** model, and **Dithered** options in the Renderer menu.

Export → Objects → saturnExport



You must be a SEGA Registered developer to use this command.

The **Sampling Density** slider in the Saturn Export Setup dialogue box has been changed from an integer value slider to a floating value slider ranging from 0.0 to 200.0 (measured in texels per SOFTIMAGE unit). This new floating value slider allows you to specify the texture slicing values more precisely, reducing possible breaks in the texel density across polygons.

The **saturnExport** command has also been modified so that it can detect shared texture maps between polygons without generating the wrong texture name reference in the Saturn **.mdl** files.

Viewer → Saturn Viewer



You must be a SEGA Registered developer to use this command.

The **Detection Timeout** slider has been removed from the Saturn Viewer Setup dialogue box and the timeout has been coded to a value of 2 seconds.

A new option has been added to the Saturn Viewer Setup dialogue box which affects the Cartdev Viewer mode. In version 3.5, you could only toggle the **Cartdev Viewer** option on and off. Now you have three options:

- **Inactive:** The Cartdev Viewer is inactive.
- **Active (no info):** The Cartdev Viewer is active. The scene is displayed on screen.
- **Active (with info):** The Cartdev Viewer is active. The scene is displayed on screen with relevant information (such as current

frame number and total number of polygons). This option is selected by default.

Sony PlayStation Converter/Viewer (PSX)

The Sony PlayStation Converter converts image files between SOFTIMAGE 3D and the Sony PlayStation. It converts polygon mesh objects, materials, and 2D textures between the two formats.

There are three new commands in the Tools module that are used with the PlayStation:

- **Export → Objects → psxExport** (see next section)
- **Import → Objects → psxImport** (see page 44)
- **Viewer → psxViewer** (see page 47)

Note

You must be a Sony Registered developer to use these commands.

Export → Objects → psxExport

The **Export → Objects → psxExport** command allows PlayStation game designers to export a hierarchy of models and textures as PlayStation data in TMD binary or RSD ASCII file formats.

Conversion Process

Geometry

Polygons that have more than four vertices are reduced to quads or triangles. Self-intersecting polygons are not identified, so they are undefined.

Generally, objects are converted to Gouraud-shaded primitives, unless the face's normals are pointing in the same direction, in which case they are converted to flat-shaded faces.

If no vertex normals exist, they are automatically calculated for each vertex in the object. Face models are converted to flat-shaded primitives. For objects using automatic discontinuity, the faces are converted to Gouraud.

If the **Merge Scene in One File** option in the PlayStation Export dialogue box is not selected, each model in the scene is saved as a separate file. This option is valid only for the TMD file format. If this option is selected, each model in the scene is merged into a single file.

Materials

Materials assigned to each polygon are converted using the diffuse area of illumination. If the original object does not have a material, the converted object is assigned the default colour.

2D Textures

During the conversion process, SOFTIMAGE 3D **.pic** files are converted to **.TIM** format. The colour depth of a **.TIM** file is 16 bits (5, 5, 5) as opposed to the 24-bit colour depth (8, 8, 8) of a **.pic** file. The conversion from 24 to 16 bits is performed by truncating the last three bits of each colour component in the **.pic** file.

If a **.pic** file is larger than 64 x 256 pixels, the file is resized to fit within these dimensions. For example, an image 128 x 128 is resized to 64 x 128 to fit within the predefined dimensions.

If more than one texture is applied to a material, only the 2D texture (global or local) with the highest overall blending is converted to **.TIM** format. All other textures for that material are ignored. If more than one texture has the same transparency value, the system selects one at random.

It is necessary to assign textures to VRAM pages so that they can be mapped to the VRAM area which is then referenced by the model itself. Textures are assigned sequentially to the available VRAM pages using the **VRAM Pages Assignment** option in the PlayStation Export dialogue box.

Textures are assigned sequentially to the available VRAM pages (there are 32 VRAM pages in total). If all the available VRAM pages have textures assigned to them but there are still textures remaining, the system returns to the first available VRAM page and starts the cycle again overwriting existing textures.

For information on VRAM page assignment, see the **VRAM Pages Assignment** option described in *Texture Options* on page 43.

UV Coordinates

The Sony PlayStation supports only the UV coordinates mapping method. If you use UV coordinates, the textures are converted directly. If you use one of the other texture mapping methods (XY, XZ, YZ, Cylindrical, or Spherical), the texture is converted to UV coordinates using a “closest match” approach. As a result, the texture may not be exported properly.

A workaround to this problem is to convert the texture projection method to UV in SOFTIMAGE 3D by choosing **Txt_Oper** → **Info UV Coord** in the Matter module, clicking OK, and specifying a name for the new texture map.

Lights, Camera, Animation!

Lights, the camera, and animation are not currently supported.

File Name Processing

All file names are output in lowercase and modified to conform with DOS file name specifications.

Scaling and Viewing Problems

If you export a hierarchy in which models have scaling values other than (1, 1, 1), and use the **Viewer** → **psxViewer** command, the result looks different on the SOFTIMAGE 3D display than with the **psxViewer**. This is because SOFTIMAGE 3D uses non-classical transformations by default, while PlayStation uses classical transformations. The difference between these systems is the order in which the translation, rotation, and scaling matrices are applied:

- Non-classical

$$S_{model} \times S_{parent} \times R_{model} \times T_{model} \times R_{parent} \times T_{parent}$$
- Classical

$$S_{model} \times R_{model} \times T_{model} \times S_{parent} \times R_{parent} \times T_{parent}$$

Unfortunately, there is no easy way to convert non-classical transformations into classical ones. When SOFTIMAGE 3D exports to PlayStation format, it generates new local transformations so that objects appear to be positioned and scaled correctly on the PlayStation.

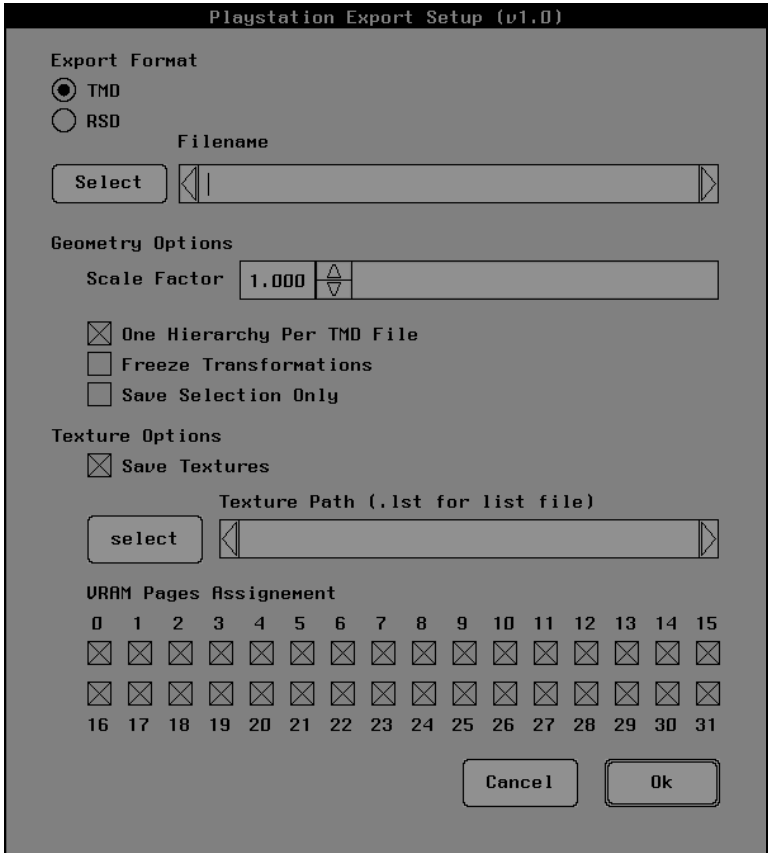
There are two workarounds for this problem:

- Use the **Effect** → **Freeze** → **Scaling** command in the Model module before exporting.
- Model with classical transformations by choosing the **Preferences** → **Classic Scaling** command.

Using psxExport

When you choose the **Export** → **Objects** → **psxExport** command, the

PlayStation Export Setup dialogue box is displayed.



Export Format

In the Export Format section, you can select either the TMD or RSD file format option.

- **TMD:** The output mode is PlayStation binary and one or more files are written with the .TMD extension. By default, all objects in a scene are saved in one file, but if you deselect the **One Hierarchy Per TMD File** option, a file is created for each model.
- **RSD:** The output mode is ASCII and four files are created for each model with the following extensions: **RSD**, **.PLY**, **.MAT**, **.GRP**. If your model contains textures, files with the extension **.TIM** are created.

The RSD file format describes the relationships between PLY, MAT, and GRP files: PLY describes positional information on vertices of a

polygon, MAT describes material information on a polygon, and GRP describes grouping information.

- **Filename:** Type a file name for the exported PlayStation files in the **Filename** text box, or click the **Select** button to open a file browser and select an existing file name. Existing file name extensions are replaced by Sony PlayStation related extensions.

Geometry Options

- **Scale Factor** allows you to resize models to minimize the loss of precision that occurs when converting floating point values (SOFTIMAGE 3D) to fixed point values (Sony). The vertices have a maximum value of 32, 767 and a minimum value of -32, 767. If you exceed these values, the following warning message is displayed: “The scaling factor has been set to [*new scale factor*].”

The warning message provides you with the accurate maximum/minimum scaling factor for your model. This value is used when importing and exporting your model.

- **One Hierarchy Per TMD File** creates one TMD file per hierarchy. This option is selected by default. If you deselect this option, one file is created per model.
- **Freeze Transformations** sets the current scaling, rotation, and translation values as the default values. It basically resolves problems occurring from object transformations and computes absolute coordinates for each vertex. This option should be deselected if your object contains animation.
- **Save Selection Only** saves only the selected object. If this option is deselected, all objects are saved.

Texture Options

- **Save Textures** converts texture picture files to TIM format. If no path is specified in the **Texture Path** text box, the TIM files are stored in the directory specified in the **Filename** text box. When you specify a path, all the textures are stored in that directory creating one file per texture. You can also generate an **.lst** file that creates one file per model containing a list of that model’s textures. Converted textures take on the name of the SOFTIMAGE 3D texture (t2d1) and not the picture file associated with that texture. If this option is deselected, the picture files are not converted.

If you don’t add the **.lst** extension, a file is not created.

- **VRAM Pages Assignment:** Video RAM (VRAM) is memory used to display models, materials, and textures. VRAM pages act as the frame buffer for the Sony PlayStation. It is within these pages that the display information is stored. There are 32 VRAM pages available resulting in a maximum possible resolution of 1024 x 512. Depending on the resolution of your image, a portion of the VRAM pages are already used by the PlayStation application's display. Once you have determined which VRAM pages are already in use, you can deselect those pages and use the rest to display the various textures which have been applied to your objects. The converter can only handle one texture per page, limiting the number of textures that can be exported with your models.

Import → Objects → psxImport

The **Import → Objects → psxImport** command allows you to import PlayStation files (models, materials, textures) into SOFTIMAGE 3D.

Conversion Process

Geometry

When importing PlayStation models, they are converted to polygon mesh models. Normal information is assigned to the model. Flat-shaded objects are converted to face objects and all other objects are converted using the automatic discontinuity default of 60 degrees.

Materials

RGB information is assigned to each polygon using its diffuse area of illumination. The Specular component is set to 1, and the Ambient component is set to 0.

If the original object does not have a material, the converted object is assigned the default colour (Diffuse=0.7, Ambient= 0.5, and Specular Decay=50).

2D Textures

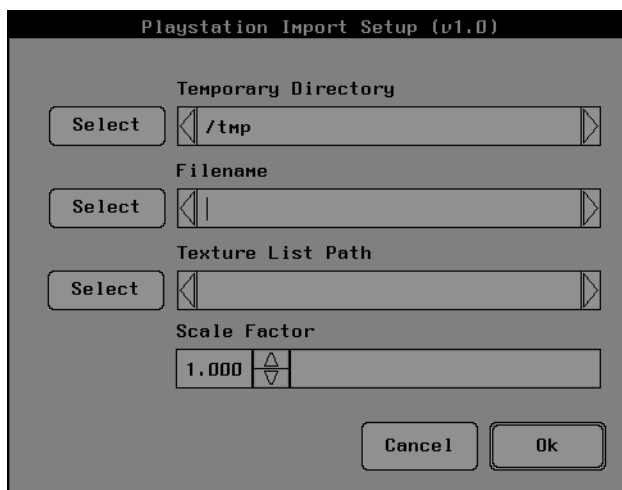
The image file converter converts .TIM files to SOFTIMAGE 3D .pic files. An alpha channel is generated for the .pic file with each alpha pixel set to 0, 128, or 255, depending on the state of the STP bit and the RGB values for the corresponding .TIM file pixel. Texture transparency is set to 100% and the source field is set to alpha, meaning that the texture's transparency depends entirely on its alpha channel.

UV Coordinates

The Sony PlayStation uses only UV coordinates as its texture mapping method. The UV coordinates are imported directly into SOFTIMAGE 3D.

Using psxImport

When you choose the **Import** → **Objects** → **psxImport** command, the PlayStation Import Setup dialogue box is displayed.



Temporary Directory

This defines a location for the converted texture files. Type a path in the text box or click the **Select** button to open a file browser and select a directory.

Filename

Type the name of the PlayStation file you want to import in this text box. Make sure to include the .RSD or .TMD file extension. You can also click the **Select** button to open a file browser and select an existing file.

Texture List Path

This option specifies a directory or file from which textures are imported from the corresponding PlayStation TIM files. If left blank, all the associated textures in the directory specified in the **Filename** text box are processed. If you specify a particular .lst file, only those

textures in that Texture List will be processed. If you specify a path without an extension, all the TIM files in that directory are processed. Alternatively, you can click the **Select** button to open a file browser and select a file.

Note

RSD files create specific texture files for each model. As a result, when a model is imported from the PlayStation its texture information is accessible and can be easily processed.

Unlike **RSD** files, **TMD** files have no specific way of referencing the TIM files. As a result, all the textures in the specified directory or .lst file must be processed. Once all the TIM files have been loaded, a VVRAM (virtual VRAM) space is created to match the textures to their respective polygons. For example, a model consisting of two polygons is imported into SOFTIMAGE 3D from the PlayStation in TMD file format. There are five textures in the texture list path specified, but only two of the textures actually relate to this particular model. Because the system has no way to reference which textures should be processed, it loads all five textures and then maps them into the VRAM space. It then begins to match the model's various polygons to their respective textures. Once the matching process is complete, all unmatched textures are discarded.

Scale Factor

This option allows you to resize models in order to minimize the distortion that occurs when converting models created using fixed point values (Sony) to models created using floating point values (SOFTIMAGE 3D). The vertices have a maximum value of 32, 767 and a minimum value of -32, 767. The same Scale Factor should be used when both importing and exporting your models.

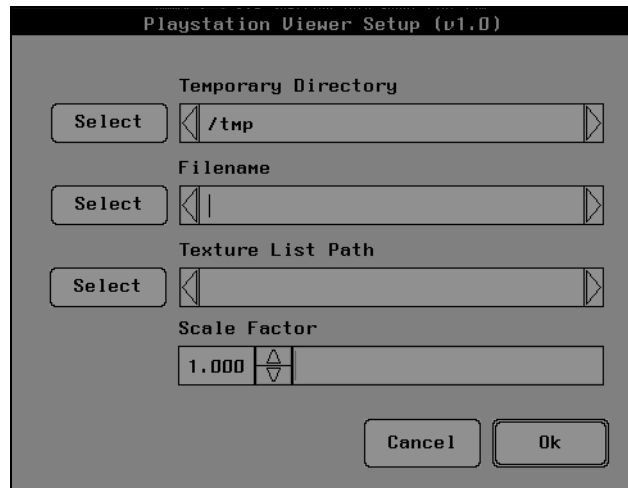
Viewer → psxViewer

The **Viewer → psxViewer** command in the Tools module allows PlayStation game designers to view PlayStation models (encoded in PlayStation TMD or RSD file format) in a viewer in SOFTIMAGE 3D.

Using the Viewer

1. Choose the **Viewer → psxViewer** command.

The PlayStation Viewer Setup dialogue box is displayed.



2. Type the name of the PlayStation file you want to load in the **Filename** text box. Be sure to include the appropriate file extension (TMD, RSD). You can also click the **Select** button to open a file browser and select a file.
3. Type the name of the PlayStation texture list file you want to load in the **Texture List Path** text box. Be sure to include the .lst extension. You can also click the **Select** button to open a file browser and select a file.
4. Click **Ok**. The Scene Viewer opens.

You can use the **z**, **o**, **p**, and **a** Supra keys with the mouse to respectively zoom, orbit, dolly, and frame all objects in the usual manner. Use the SOFTIMAGE 3D playback control box to play animation, if any. For a description of the other options in this window, see *Scene Viewer Parameters (for both Saturn and PlayStation)* on page 49.

5. When you have finished, click **Exit** in the Viewer.

Note

If you view a hierarchy exported with **Export → Objects → psxExport** and the models have scaling values other than (1.0, 1.0, 1.0), it will look different in the Scene Viewer than on the PlayStation console. This is because SOFTIMAGE 3D uses non-classical transformations by default, while PlayStation uses classical transformations. For a description of some ways to avoid this situation when exporting, see *Scaling and Viewing Problems* on page 41.

PlayStation Viewer Setup Parameters

Temporary Directory

Specifies a temporary location for the loaded texture files. Type a path in the text box or click the **Select** button to open a file browser and select a directory.

Filename

Type the name of the PlayStation file you want to load in the **Filename** text box. Make sure to include the **.RSD** or **.TMD** file extension. You can also click the **Select** button to open a file browser and select a file.

Texture List Path

Loads the textures from the corresponding PlayStation **TIM** file, if any. If left blank, all the associated textures are loaded. If you specify a particular **.lst** file, only those textures in that texture list are loaded. You can also click the **Select** button to open a file browser and select a file.

Scale Factor

Resizes models to minimize the distortion that occurs when converting models created using fixed point values (Sony) to models created using floating point values (SOFTIMAGE 3D). The vertices have a maximum value of 32, 767 and a minimum value of -32, 767. The same scale factor should be used when importing, exporting, and viewing your objects.

Scene Viewer Parameters (for both Saturn and PlayStation)*Show Icons*

Displays icons such as curves and lights.

Enable Hardware Texture

Uses the hardware renderer to show the model's texture. When this option is selected, the following options are available:

Texture Blending Approx

Specifies the method used to blend the material colour with the texture colour:

- **Without Mask** ignores the alpha content of the composition.
- **Alpha** interprets the alpha component of the texture as a proportion of texture colour to use against the material colour.

Texture Repeat

Controls whether textures are repeated (tiled):

- **None** disables the repeat factor.
- **Grid** enables the repeat factor.

Texture Type

Determines which textures are displayed: **Local** or **Global**.

Display Optimization

Controls how the model appears when you use the **z**, **o**, and **p** Supra keys with the mouse to zoom, orbit, and dolly:

- **None** displays the scene in Shade view with textures.
- **Shade** displays the scene in Shade view without textures.
- **Wireframe** displays the scene in Wireframe view.

Standalones

liveOGL in the SOFTIMAGE Live Viewer (LiveGL/LiveOGL on IRIX)

The **-e** command line option expands all instances in a scene. This option works the same way as the **Duplicate → Expand → All** command in SOFTIMAGE 3D.

Keyboard Options

- **R** – Resets the camera to its original settings that were saved in the scene.
- **d** – An option for using display lists. For some platforms, using this option improves the performance of the display.
- **k** – Controls frustum culling. Only objects in the camera's field of view are drawn. This option can help you improve the performance in the viewer.

The culling test checks for an intersection between the camera's field of view and the bounding box of the object. Two bounding boxes come into play, one for the object itself and another for the entire branch (the object and all its children). The culling test uses the branch bounding box as its starting point. The branch bounding box can be either:

- Completely outside the field of view (SOFTIMAGE Live ignores the branch).
- Completely inside the field of view (SOFTIMAGE Live draws the entire branch without further testing).
- Partially inside the field of view (the branch bounding box provides insufficient information to make a decision so SOFTIMAGE Live then begins a more detailed examination by checking the bounding box of the top-level object within the branch). The object's bounding box can be either:
 - Completely outside the camera's field of view (SOFTIMAGE Live ignores the object).
 - Completely inside the camera's field of view (SOFTIMAGE Live draws the object and then performs the test on the children).
 - Partially inside the camera's field of view (SOFTIMAGE Live draws the object and then performs the test on the children).

Optimizing the Culling

Here are some tips to optimize the culling feature:

- Expand all instances that are unnecessary. This is because instances are not eligible for the culling test and they use the triangle display code as opposed to the faster Tmesh display code. Be careful which instances you expand: an instance whose master does not have any animation can be expanded without a problem, but an instance whose master has a geometry deformation animation makes the playback slower if expanded.
- Do not leave unused animations in you scene. Animated objects are not eligible for culling.
- Do not leave unused channel connections in your scene. Channel animated objects are not eligible for culling.
- Group objects that are close to one another in the same hierarchy.
- Large objects should be divided into smaller ones.

soft -R

Two new command line options have been added to the SOFTIMAGE rendering standalone (**soft -R**). They are:

```
[-p <pre-frame script path>]  
[-P <post-frame script path>]
```

- p Specifies the pre-frame script that is executed.
- P Specifies the post-frame script that is executed.

soft2everest

The **-R** option no longer exists for the **soft2everest** standalone.

Custom Effects

Installing Custom Effects in Different Places

The following environment variables now support multiple paths so that you can install custom effects in more than one location.

- SI_CUSTOM_ACTOR
- SI_CUSTOM_MATTER
- SI_CUSTOM_MODEL
- SI_CUSTOM_MOTION
- SI_CUSTOM_TOOLS

For example, you can set SI_CUSTOM_MOTION to point to the location of the motion effects shipped with SOFTIMAGE 3D and the sample effects shipped with SOFTIMAGE SDK.

On IRIX:

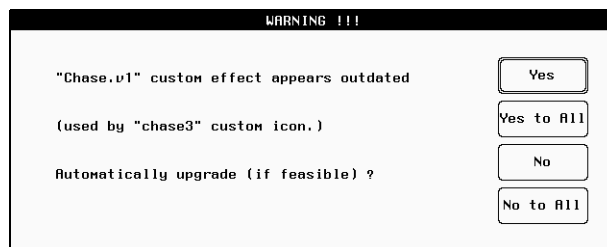
```
setenv SI_CUSTOM_MOTION /usr/softimage/3D/motion:  
/usr/softimage/SDK/examples/motion
```

On Windows NT:

```
set SI_CUSTOM_MOTION=C:\softimage\3D\motion;  
C:\softimage\SDK\examples\motion
```

Upgrading SOFTIMAGE SDK Custom Effects

When you load a scene that contains custom effects written with SOFTIMAGE SDK, SOFTIMAGE 3D automatically detects if more recent versions of the effects are available. If so, it gives you the option of upgrading the effects or keeping the older versions.



- To upgrade all instances of the effect, click **Yes to All**.
- To upgrade instances one by one, click **Yes** (to upgrade) or **No** (to keep old version).

- To keep the scene as is (keep old versions of all effects), click **No to All**.

Note

Custom effect .cus file names have the format **effect.vn.cus**, where **n** is the version number. In SOFTIMAGE 3D, the name of SDK-based effects appear in the menu, but not the version number. The menus always use the most recent version of an effect.

Installing SOFTIMAGE SDK Custom Effect Plug-ins

This section describes how and where to install custom effect plug-ins created with SDK. SDK custom effect plug-ins must be built specifically for each operating system; that is, you cannot run an IRIX custom effect on Windows NT, or vice versa. As well, custom effect plug-ins must be specifically built for Windows NT architecture; that is, you cannot run an Intel custom effect on an Alpha machine, or vice versa.

Custom Effect Plug-in Components

An SDK custom effect plug-in must consist of either two or three files:

- The custom effect interface: **<the_effect>.cus**
- The custom effect binary: **<the_effect>.so** (for IRIX) or **<the_effect>.dll** (for Windows NT)
- The custom effect icon: **<the_effect>.hrc** (only for persistent effects)

Persistent Custom Effects

Persistent custom effects are effects that “stay alive” with the scene, such as the Explode effect. Because persistent effects stay in the scene and can be modified, they need an icon to represent them. Only persistent custom effects have an icon, so they have three files associated with them (interface, binary, and icon).

Immediate Custom Effects

Immediate custom effects are effects which happen once in the scene, such as the Breakup effect. Because immediate effects do not “live” in the scene, they do not need an icon, so they only have two files associated with them (interface and binary).

Environment Variables

SOFTIMAGE 3D finds custom effects according to a number of environment variables. For IRIX, all environment variables for SOFTIMAGE 3D are defined in the `.softimage` file in your home directory. To see the current environment variable values, open an IRIX shell and type `env`.

For Windows NT, all environment variables for SOFTIMAGE 3D are defined in the `setenv.bat` file in the `%SI_LOCATION%\3D\bin` directory (for example, if you installed SOFTIMAGE 3D in its default directory, the location would be `C:\SOFTIMAGE 3D-v3.51\3D\bin\setenv.bat`). To see the current environment variable values, double-click on the **Edit SETENV.BAT** icon in the SOFTIMAGE 3D program group in the Windows NT Program Manager.

Custom effects are grouped together by module: all Model module effects are together, all Motion module effects are together, etc. Each module has an associated environment variable that points to the directories where the effects' interface (`.cus`) files and optional icon (`.hrc`) files are located.

You must determine the value of the `SI_CUSTOM_<MODULE>` environment variable that corresponds to your custom effect plug-in. For example, if the custom effect is a Model effect, the `SI_CUSTOM_MODEL` environment variable is associated with it. This is the list of possible environment variables:

`SI_CUSTOM_MODEL`
`SI_CUSTOM_MOTION`
`SI_CUSTOM_ACTOR`
`SI_CUSTOM_MATTER`
`SI_CUSTOM_TOOLS`

Installing the Custom Effect Plug-ins

1. Install the custom effect plug-in interface file (and optionally icon file)

To install the custom effect plug-in, choose one of these options for its location:

- To keep the custom effect plug-in in the same location as your existing custom effects, place the plug-in's `.cus` file (and `.hrc` icon file for persistent effects) in the directory specified by the

appropriate `SI_CUSTOM_<MODULE>` environment variable.

or

- To put the custom effect plug-in in a separate location, modify the value of the `SI_CUSTOM_<MODULE>` environment variable. Simply replace the existing path value with the path of the plug-in's location.
 - On IRIX, use a colon (:) to separate multiple directories in a path statement. For example:

```
setenv SI_CUSTOM_MODEL "usr/softimage/3D/custom/  
model:/myeffects/model"
```
 - On Windows NT, use a semicolon (;) to separate multiple directories in a path statement. For example:

```
SI_CUSTOM_MODEL=C:\SOFTIMAGE3D\3D\custom\model;  
C:\myeffects\model
```

2. Install the custom effect plug-in binary file

The custom effect binary file (the “brain” of the effect) is stored in another file. The file extension is `.so` for IRIX and `.dll` for Windows NT.

On IRIX, the custom effect binary file must reside in one of the directories listed in the `LD_LIBRARY_PATH` environment variable. On Windows NT, the custom effect binary file must reside in one of the directories listed in the `PATH` environment variable.

- If the custom effect plug-in is in the same location as your existing custom effects:
 - On IRIX, place the plug-in's binary (`.so`) file in one of the directories specified in the `LD_LIBRARY_PATH` environment variable.
 - On Windows NT, place the plug-in's binary (`.dll`) file in one of the directories specified in the `PATH` environment variable.

or

- If the custom effect plug-in is in a separate location:
 - On IRIX, modify the value of the `LD_LIBRARY_PATH` environment variable by adding the path of the plug-in's binary (`.so`) file. Use a colon (:) to separate multiple directories. For example:

```
setenv LD_LIBRARY_PATH "usr/softimage/3D/dso:  
/myeffects/dso"
```

- On Windows NT, modify the PATH environment variable by adding the path of the plug-in's binary (.dll) file. Use a semicolon (;) to separate multiple directories. For example:
PATH=%PATH%;C:\SOFTIMAGE3D\3D\bin;C:\myeffects\bin

3. Restart SOFTIMAGE 3D

On IRIX, open a shell and type `source ~/.softimage` to reinitialize the environment variables.

On both IRIX and Windows NT, you must restart SOFTIMAGE 3D for the changes to take effect.

Ambulate Custom Effect

Ambulate is an SDK-based persistent custom effect which allows you to create walking animation by manipulating a series of footsteps. The effect interpolates the feet of a model between the positions and orientations of the controlling footsteps. To create interesting “ambulating” animations, the footsteps can be positioned and oriented in any way. The torso of the character is also animated based on the motion of the feet.



This effect can be found in the SOFTIMAGE 3D version 3.51 **gifts** directory. These gifts are provided freely “as is” and without any warranty or support. Softimage assumes no liability whatsoever and makes no representations in regard thereto.

Files

The Ambulate effect requires the following files, and should be installed in the directory pointed to by the `SI_CUSTOM_MOTION` environment variable.

For IRIX:

- Ambulate.cus
- ambulate.so
- ambulate.hrc

For Windows NT:

- Ambulate.cus
- ambulate.dll

- ambulate.hrc

You can create the footstep hierarchy any way you wish. You can use **Duplicate** ➔ **Repetition** or any other method you choose. To help you out, a modelling effect called *Stepmaker Custom Effect* (on page 58) is available. This effect uses a curve to generate the footstep hierarchy required by the Ambulate custom effect.

Using Ambulate

Pick Left Footprints Hierarchy

The footprint hierarchies for the two feet are expected to be flat hierarchies. There should be a number of models (representing the individual footsteps for one of the feet) grouped together under a single parent node. You can pick any footstep or the parent of the hierarchy. The footsteps are traversed in the order in which the footstep objects were placed in the hierarchy.

Pick Right Footprints Hierarchy

The right footprints hierarchy should be organized the same way as the left one – with all footsteps as children of a single parent node.

Pick Left Foot

Choose a model which represents the left foot. The two feet should be grouped under a common ancestor node, which receives animation, as does the feet. Typically, a foot might be the end effector of an inverse kinematic chain that represents a leg of the character, but any object can be selected as a foot.

Pick Right Foot

Choose the model which represents the right foot.

Pick Torso Root

Choose a node which is an ancestor of both feet. In the “zero” state (prior to applying the effect), this node should be positioned between the feet at foot level (not at waist level).

Parameters

Move Time Per Step

Indicates the length of time (in seconds) that the moving part of each step should take. A smaller time value represents a faster walking speed.

Idle Time Per Step

Indicates the length of time (in seconds) that the immobile part of each step should take. Usually, these first two parameters (move time and idle time) would be equal, giving the walk a 50% duty cycle.

Initial Step Offset (Right and Left)

Control how far ahead or behind each foot should be from the start of the cycle. To create a normal walk, set these parameters to 0.5 for the “leading foot” and 0.0 for the “trailing” foot.

Hip Sway

Governs the side-to-side swaying motion of the torso over the course of a step cycle. A value of 1.0 indicates that the torso swings as far to the side as the foot that is currently “down.” Values larger than 1.0 signify that the torso sways further to the sides than the width of the stance.

Maximum Step Height

Controls how high (in SOFTIMAGE 3D units) each foot rises above the level of the footsteps when at the highest point of each step. The path of each footstep follows a parabolic curve. Later versions of this custom effect may allow you to create a “prototype” step curve.

Stepmaker Custom Effect

Stepmaker is an SDK-based immediate effect which is to be used with the Ambulate persistent custom effect. It allows you to create a series of footsteps from a curve of any kind, .



This effect can be found in the SOFTIMAGE 3D version 3.51 **gifts** directory. These gifts are provided freely “as is” and without any warranty or support. Softimage assumes no liability whatsoever and makes no representations in regard thereto.

Files

The Stepmaker custom effect requires the following files, and should be installed in the directory pointed to by the SI_CUSTOM_MODEL environment variable.

For IRIX:

- Stepmaker.cus

- stepmaker.so

For Windows NT:

- Stepmaker.cus
- stepmaker.dll

Using Stepmaker

Create a linear or NURBS curve and make sure it is selected. Two flat hierarchies, containing a series of left and right footsteps (represented by nulls), are created using the parameters specified in the custom dialogue box.

Parameters

Step Length

Represents the distance (in SOFTIMAGE 3D units) taken by each step. In other words, it is the approximated distance (along the curve, not necessarily in a straight line) between two subsequent steps taken by the same foot.

Stance Width

Indicates how far apart (in SOFTIMAGE 3D units) the feet should be positioned when the character is standing normally.

Flat Feet

Indicates whether the footsteps should follow the tangent of the curve exactly, or whether all the steps should be created parallel to the xz plane. This is somewhat like the difference between walking up a hill and walking up stairs.

DKit Information

DKit for Version 3.51

The version of DKit distributed with SOFTIMAGE 3D version 3.51 is the same version shipped with SOFTIMAGE 3D version 3.01. DKit includes no new features, just bug fixes for version 3.51 (see the next page).

This release of DKit provides a transition period for developers before they begin to using SOFTIMAGE SDK, which is the next-generation replacement for DKit.

3.51 Model Files

DKit does not support the new tokens introduced into model files by SOFTIMAGE 3D version 3.51:

- Node matte
- Branch matte
- Projected NURBS curves
- NURBS curves as paths

When a DKit program loads in a model file, it skips these tokens and does not save them back into the file.

Custom Commands

Replace *target source* {-all | -geometry | -material} [-n]

- | | |
|-----------|---|
| -all | Replaces the geometry, materials, and textures of <i>target</i> with the model data in <i>source</i> . |
| -geometry | Replaces the geometry of <i>target</i> with the model data in <i>source</i> . |
| -material | Replaces the materials and textures of <i>target</i> with the model data in <i>source</i> . |
| -n | By default, the model data in <i>source</i> must include valid polygon normals. However, if you include the -n option, SOFTIMAGE 3D automatically recomputes the normals. |



When you replace a model's materials and textures, they are deleted from the scene (unless they are used by some other model in the scene).

Backward Compatibility

DKit no longer supports SOFTIMAGE 3D version 2.5.

Bug Fixes

- DKit handles shape animation of NURBS curves and surfaces properly.

Previously, DKit (DK - dbSceneSave) would reset Additive shape animation to Average when saving a scene.

Note that DKit does not support the shape animation mode for a model's ASCII file format, only for .hrc files.

- DKit copies trimmed curves of NURBS surfaces, so custom effects like Duplicator can handle NURBS surfaces.
- 3.0 DKit custom effects can be applied to 3.5 NURBS models. DKit skips unsupported NURBS tokens.
- DKit saves the name of spot light interests.

Previously, the DKit (DK - dbSceneSave) would remove constraints between a spot light interest and another element because the spot light interest was not saved properly.

- DKit properly handles relations (such as constraints) between elements with similar names.

Previously, if two models in a scene had the same name (for example, if prefixes had been removed from `prefix1-name` and `prefix2-name`), DKit could not always save the scene properly. For example, if a constraint existed between the two models, DKit would not set the constraint properly because of the model names.

Replace Custom Command

If a DKit effect uses the Replace command without first executing a Read command, the effect may not update properly. The effect may appear to be frozen, but it is actually working – just not updating.

The workaround is to always execute a Read command before a Replace:

```
Read foo proc
Replace foo proc -normal
```

Channel Drivers DKit

DK prefix changed to CDK

The names of the functions, types, and files in the Channel Development Kit now use the CDK prefix. For example, the function `DK_driverInit()` is now named `CDK_driverInit()`.

Broadcast and Listen Drivers

Broadcast drivers broadcast the data transferred between SOFTIMAGE 3D and a driver by the following functions:

- `CDK_driverSetDofValue()`
- `CDK_driverSetDofTime()`
- `CDK_driverGetOp()`
- `CDK_driverSendData()`

Connecting to the Local X Server (IRIX Only)

Channel drivers cannot rely on the `DISPLAY` environment variable being set correctly when they run on a remote machine (the CDK no longer resets the `DISPLAY` variable for the remote process using `env`).

If a driver needs to access X resources, it is responsible for opening a connection to the local X server.

Channel drivers that use GL may need to call `dglopen()` if running remotely.

 Channel drivers can use `CDK_driverGetInfo()` to determine if they are running locally or remotely.

`CDK_driverPortGetState()` and `CDK_driverPortSetState()`

`CDK_driverPortGetState()` and `CDK_driverPortSetState()` provide a platform-independent interface for configuring serial ports.

```
CDK_Boolean CDK_driverPortGetState
(
    FILEHANDLE f,
    CDK_DriverPortState *state
);
```

```
CDK_Boolean CDK_driverPortSetState
(
    FILEHANDLE f,
    CDK_DriverPortState *state
);
```

The state structure encapsulates the configuration of a serial port:

```
typedef struct
{
    CDK_DriverPortBaud baud; /* baud rate */
    CDK_DriverPortDataBits dataBits; /* number of data bits */
    CDK_DriverPortParity parity; /* none, odd, even */
    CDK_DriverPortStopBits stopBits; /* number of stop bits */
    CDK_Boolean flowControl; /* RTS/CTS flow control */
    int readTimeout; /* in milliseconds */
} CDK_DriverPortState;
```

Field	Possible Values
baud	CDK_DRIVER_PORT_BAUD_9600 CDK_DRIVER_PORT_BAUD_19200 CDK_DRIVER_PORT_BAUD_38400 CDK_DRIVER_PORT_BAUD_56000 CDK_DRIVER_PORT_BAUD_57600 CDK_DRIVER_PORT_BAUD_115200
dataBits	CDK_DRIVER_PORT_DATABITS_5 CDK_DRIVER_PORT_DATABITS_6 CDK_DRIVER_PORT_DATABITS_7 CDK_DRIVER_PORT_DATABITS_8
parity	CDK_DRIVER_PORT_PARITY_NONE CDK_DRIVER_PORT_PARITY_ODD CDK_DRIVER_PORT_PARITY_EVEN
stopBits	CDK_DRIVER_PORT_STOPBITS_1 CDK_DRIVER_PORT_STOPBITS_2
flowControl	TRUE to enable flow control, FALSE to disable.
readTimeout	Length of time (in milliseconds) to block during read operations. Set to 0 for non-blocking reads.

CDK_driverPortReadNum()

This is a version of `CDK_driverPortRead()` that tells you how many bytes of data it was able to read from a serial port.

```
CDK_Boolean CDK_driverPortReadNum
(
    FILEHANDLE f,
    char *buf,
    int num,
    int tries,
    int *numRead
);
```

Tries to read `num` bytes from a serial port into `buf`. The read operation is repeated `tries` times. The actual number of bytes read from the serial port is stored in `numRead`.

Returns `FALSE` if `num` is not equal to `numRead`.

CDK_driverGetDofValue() and CDK_driverGetDofTime()

Both these functions use an output argument to store data retrieved from SOFTIMAGE 3D. The function return value (a status code of type `CDK_ChnStatus`) indicates what is stored in the output argument.

Return Value	Description
CDK_CHN_SUCCESS	A new value/time was retrieved and is stored in the output argument.
CDK_CHN_UNCHANGED	No new value/time to retrieve, and the last value/time retrieved from SOFTIMAGE 3D is stored in the output argument.
CDK_CHN_ERROR	An error occurred, and 0 is stored in the output argument.

Writing Portable DKit Applications

Naming Custom Effects

Don't use the names of platform-specific commands as the names of custom effects. For example, if you use the name "label" for a custom effect, when you try to run the effect on Windows NT, SOFTIMAGE 3D will execute the `label` command.

Executing Shell Commands

A `.cus` file can use the `Shell` and `NT_Shell` commands to execute platform-specific commands:

```
Shell rm tmpfile
NT_Shell del tmpfile
```

The IRIX version of SOFTIMAGE 3D executes the `Shell` command, while the Windows NT version executes the `NT_Shell` command.

Temporary Files

Portable custom effects should use temporary files. Custom effects create temporary files whenever custom commands use file names that do not contain an explicit path. For example, this `Write` command creates a temporary file:

```
Write input tmp_file
```

If the `Write` command specifies an explicit path such as `./data_file` or `/usr/home/james/data_file`, then the file is not a temporary file (and the effect is not portable since it uses a platform-specific path.)

The location of temporary files depends on the platform:

- `/usr/tmp` on IRIX
- `%TEMP%` on Windows NT

If `/usr/tmp` is inaccessible on IRIX, or if `TEMP` is undefined or `%TEMP%` is inaccessible, then temporary files are put in the current directory.

To determine the location of temporary files, custom effects can use the `DK_temporaryDirectory()` function:

```
const char *DK_temporaryDirectory( void );
```

This function returns the path to the temporary directory used by

SOFTIMAGE 3D to store temporary files created by custom effects. If the temporary directory is inaccessible, `DK_temporaryDirectory()` returns "".

`DK_temporaryDirectory()` always appends a path separator (a “\” or “/” character) to the path name, so you can concatenate the file name to the path name as follows:

```
char buf[FILENAME_MAX];
sprintf(buf, "%s%s", DK_temporaryDirectory(),
        fileName );
```

Or:

```
char buf[FILENAME_MAX];
strcpy( buf, DK_temporaryDirectory() );
strcat( buf, fileName );
```

Byte Swapping

SOFTIMAGE 3D always writes out information in “big endian,” the byte ordering of the RISC processors. All DKit functions have been modified to properly byte swap the information read from a file or pipe.

Reading Tagged Point Information

To read the tag information output by the `Write` command, use the function `DK_byteswap_fread()`.

```
size_t DK_byteswap_fread
(
    void * buffer,
    size_t size,
    size_t count,
    FILE * stream
);
```

`DK_byteswap_fread()` reads up to count items of size bytes from the input stream, and stores them in buffer.

For example, the following command writes tag information into the temporary file `tag_info`:

```
Write SourceModel tag_info -tag
```

Tag information is written in binary format, and the first four bytes of the file specify the number of bytes to follow. Each subsequent byte indicates whether or not a point is tagged.

To read the first four bytes:

```
DK_byteswap_fread( &nb_of_points, 4, 1, fp );
```

`DK_byteswap_fread()` swaps the bytes if necessary.

Initializing Pipes on Windows NT

On Windows NT, use `DK_streamInitialize()` to initialize pipes so that they work with SOFTIMAGE 3D:

```
main( int argc, char *argv[] )
{
...
#ifdef _WIN32
    DK_streamInitialize();
#endif
...
}
```

Calling `Dk_dbInit()`

DKit applications that work with scenes and scene elements must call `DK_dbInit()`.

Location of the `soft.status` File

On IRIX, DKit applications write their error messages in `/usr/tmp/soft.status`.

To locate the `soft.status` file on Windows NT, SOFTIMAGE 3D:

- Uses the `TEMP` environment variables.
- If both environment variables are undefined, `soft.status` is opened in the current working directory.
- If the current working directory cannot be used, then `stderr` is used.

Model User Data

The type identifier string for user data nodes can be any length (previously, the type string was limited to a maximum of four characters).

This fix is backwards compatible, so version 3.5 scenes containing long type names can be loaded into version.0 (which will ignore the custom nodes with long type names).

Incompatible Model files (v3.1 Windows NT to and from v3.x IRIX)

DKit v3.1 Windows NT and SOFTIMAGE 3D v.3.1 Windows NT did not perform byte swapping correctly for some internal data structures, so model files containing user data attached by Windows NT versions are incompatible with the IRIX version of SOFTIMAGE 3D. You cannot load version 3.1 Windows NT model files containing user data into v3.x and IRIX.

Trying to load an incompatible model file will corrupt the model data.

Dialog Editor

- You cannot create more than one level of submenus (a submenu cannot contain its own submenus).
- The Dialog Editor now supports a new Text/Edit type: numeric integers that are positive and non-zero.

mental ray Programming

Message Routines

Message routines do not append new lines to messages.

mi_call_shader()

The function prototype for `mi_call_shader()` is:

```
miBoolean mi_call_shader(
    miColor *result,
    miShader_type type,
    miState* state,
    miTag shader);
```

mi_light_info()

```
void mi_light_info(
    miTag tag,
    miVector *org,
    miVector *dir,
    void **paras);
```

A light source can have an origin (`org`) or a direction (`dir`), but not both, so the unused vector is set to a zero vector. This allows you to identify directional (infinite) light sources because their `org` pointer is set to zero.

Since `mi_light_info()` always returns a value in the `org` and `dir` parameters, they cannot be NULL pointers.

mi_lock() and mi_unlock()

The function prototypes for `mi_lock()` and `mi_unlock()` are:

```
void mi_lock(
    const miLock *lock);

void mi_unlock(
    const miLock *lock);
```

Use these functions to surround a critical section as follows:

```
miLock lock;

mi_lock( &lock );
if (!is_init) { /* critical section */
    is_init = miTRUE;
    initialize();
```

```
}  
mi_unlock( &lock );
```

mi_matrix_invert()

The function `mi_matrix_invert()` takes two parameters:

```
miBoolean mi_matrix_invert(  
    miMatrix r  
    miMatrix a);
```

mi_par_thread()

The *mental ray Programmer's Guide* incorrectly refers to the `mi_par_thread()` function as `mi_par_nthreads()`.

Scene Description

The syntax for describing vertices is:

```
v indexint  
[ n indexint ]  
[ t indexint ]  
[ m indexint [ indexint indexint ] ]
```

State Variables

The section of the *mental ray Programmer's Guide* that discusses state variables refers to the include file `mi_shader.h` when it should refer to the file `shader.h`.

Licensing between IRIX and Windows NT

The interoperability between SOFTIMAGE 3D version 3.51 on Windows NT and IRIX has been enhanced to include cross-platform licensing.

You can now establish either a Windows NT or IRIX machine as your FLEXlm license server and have either platform type designated as client machines. For example, you can instruct any Windows NT machine as a client to take its license from an IRIX license server on the same network, or an IRIX machine to take its license from a Windows NT license server.

Note

For cross-platform licensing to work, make sure that SOFTIMAGE 3D version 3.51 is installed on both platforms, and that FLEXlm version 4.1b is running on all license servers.

To support cross-platform licensing, simply follow the installation process for licensing in the *SOFTIMAGE 3D Installation Guide*. Then when prompted by the Install program for the name of the FLEXlm server (if you are installing as a FLEXlm client configuration), type in the name of that machine, whether it is a Windows NT or IRIX machine. The Install program will then set the following accordingly:

LM_LICENSE_FILE = 744@<name of FLEXlm license server (IRIX or Windows NT)>

If you need to make any changes to this information, it can be found in the list of environments that SOFTIMAGE 3D sources before starting:

- On IRIX: `$SI_LOCATION/.softimage`
- On Windows NT: `$SI_LOCATION\3D\bin\SETENV.BAT`

Working with both IRIX and Windows NT (linktab.ini)

Issues when Using Both Platforms

There are several issues to be aware of when working with SOFTIMAGE 3D on both IRIX and Windows NT machines:

- Windows NT is not case sensitive. It mixes upper and lower case file names. For example, FILENAME.txt is considered to be the same as filename.txt. IRIX, however, is case sensitive.
- There are many providers of NFS for PCs: make sure to read and understand the settings regarding case handling of file names. The syntax used by your provider to identify a mounted file system is also important. Some use the UNC convention:

`\\machinename\mount_point`

while others use a UNIX style name:

`machinename:/mount_path/mount_point.`

- SOFTIMAGE 3D handles both forward and backward slashes on Windows NT.
- In a DOS shell, backward slashes are retained as they are typed. For example, if you type:

`<<cd USERS>>`

the shown path is in upper case. If you type:

`<<cd users>>`

it is shown as lower case. At this point, starting SOFTIMAGE 3D retains the path as it is, including upper and lower case as shown in the path.

Interoperability Problems

Interoperability problems may occur in the following situations:

- Distributed rendering using mental ray.
- Moving a scene from one machine to another.
- Sharing databases (or user accounts) between IRIX and Windows NT machines.
- Resolving IRIX symbolic links from the Windows NT machine.

- Resolving installation or configuration mismatches between machines.

When a scene is moved from one platform to another or accessed from a remote platform using NFS, there is the potential for interoperability problems concerning files outside a database.

Using the **linktab.ini** file (as described in the next section) transparently resolves these issues and helps you move from one architecture to another effortlessly.

The linktab.ini File Can Help You

In SOFTIMAGE 3D, all files composing a scene are normally placed in a given database. Problems occur when references to files outside of the database are kept within a scene, such as picture files taken from another location. SOFTIMAGE 3D needs a way to resolve those references (find the files) from any machine on the network. This is done using the **linktab.ini** translation table (or equivalent).

The **linktab.ini** file is a simple translation table. It is quite simple to use and once put in place, it can improve how SOFTIMAGE 3D works between IRIX and Windows NT.

The **linktab.ini** file is an ASCII file that is used to specify one-to-one relations between IRIX and Windows NT paths. Each line of the **linktab.ini** file represents such a relation. By convention, the paths maintained in the file are IRIX style, therefore, the **linktab.ini** file is not required on IRIX hosts, only on Windows NT hosts.

The **linktab.ini** file provides SOFTIMAGE 3D with the necessary information to locate and load files that are either on a remote machine or are installed at a different location on the current machine. This same information is also used when saving scenes from a Windows NT machine. In the case of distributed rendering, it also helps associate shaders between IRIX and Windows NT.

How Does linktab.ini Work?

When SOFTIMAGE 3D finds a **linktab.ini** file, it uses the file extensively to ensure that scenes remain compatible across platforms. When SOFTIMAGE 3D is launched, it looks for the **linktab.ini** file in the directory pointed to by the `SI_LOCATION` environment variable. If it finds the file, it builds an internal table of path translation (Windows NT to IRIX and IRIX to Windows NT). Internally, SOFTIMAGE 3D builds the list of all network drives

currently connected to the Windows NT machine as well as the associated mount point. This list in conjunction with the **linktab.ini** file provides the necessary information to translate paths to and from IRIX.

Path preferences are translated according to the following rules:

- When loading a scene:
 - IRIX paths are translated (if possible) to the equivalent mount point and then to the corresponding network drive letter.
 - UNC references are translated to the corresponding network drive letter.
- When saving a scene:

Network drive letters are translated to the given UNC mount point and then to the corresponding IRIX path (if an entry is found in the **linktab.ini** file table).

For example:

If there are two accounts on an IRIX machine called:

`/home/user1`

`/home/user2`

that are in fact a logical link to:

`/mnt1/allusers/user1` and `/mnt1/allusers/user2`

on the Windows NT machine, a network drive is connected (using NFS) to `sgi1:/mnt1/allusers` and in this case, drive `h`.

Then you can build a **linktab.ini** file that contains the following information:

```
sgi1:/mnt1/allusers!user1    /home/user1
sgi1:/mnt1/allusers!user2    /home/user2
sgi2:/usr/people/user3       /usr/people/user3\
\\pc_server1\share_location  /pc_server1
```

When SOFTIMAGE 3D encounters the path `/home/user1/...` at load time, it translates it to `h:/user1/...` which is the equivalent Windows NT path. In this example, the true mount point is `sgi1:/mnt1/allusers` and the exclamation point (!) indicates that only that portion of the path should be looked up in the current list of NFS network drives.

However, when saving a scene, the path is translated again to the proper IRIX path `/home/user1/...` so that scenes remain accessible from both platforms.

The first path argument of each line is the Windows NT path while the second path argument on each line is the IRIX equivalent.

In the **linktab.ini** file example, the third and fourth line indicates other situations. The third line indicates that the path `/usr/people/user3` comes from machine `sgi2` at the same location. On the other hand, the fourth line presents a situation where a Windows NT file server is being used by both Windows NT and IRIX machines. This line indicates that on IRIX machines, references to files located on the `pc_server1` remote file server should be kept using the IRIX path `/pc_server1`, or in other words, that `\\pc_server1\\share_location` is mounted as `/pc_server1` on all IRIX machines.

Maintaining the linktab.ini File

The **linktab.ini** file is not maintained by SOFTIMAGE 3D. When changes are made to the network configuration or data is moved from one location to another, you must update this file accordingly. It is recommended that users familiar with the Windows NT and IRIX operating systems, such as system administrators, perform this task.

Troubleshooting

The translation process that SOFTIMAGE 3D does can be put in a DOS shell by previously setting the `SI_LINKTAB_DEBUG` environment variable to “yes”. This trace lists all the current translation information that is being used by SOFTIMAGE 3D to match corresponding IRIX and Windows NT file names.

Remember that the matching process is done from top to bottom. This means that for a given path reference, the line entries of the **linktab.ini** file are searched in order. A rule of thumb is to place the longest matching IRIX path first because they are the most restrictive and should be processed first. This way you reduce the chances of building a table with conflicting rules. The following shows an example of a **linktab.ini** file containing conflicting rules:

```
sgi1:/mnt1/allusers!user2    /home
sgi2:/home/people/user3      /home/people/user3
```

In this example, when SOFTIMAGE 3D loads a scene that references the path `/home/people/user3`, the first is applied which results in an invalid path, such as `h:/user2/people/user3`. This is why the longest IRIX path should be placed first in the table.

There are numerous examples of conflicting rules and the person maintaining the **linktab.ini** file should always look for and reorder the rules or change the conflicting names, if possible. Never match upper and lower case letters in a path that is to be exported to other platforms. This reduces the chance of case mismatching problems.

Checklist for Interoperability Problems

- Verify that all remote locations are mounted on both Windows NT and IRIX machines.
- Double check permission access to all remote locations.
- Double check that mount point syntax looking for mixed upper and lower case names.
- Use the `SI_LINKTAB_DEBUG` environment variable to trace the process.
- Look for conflicting rules in the **linktab.ini** file.

For more information, see *Distributed mental ray Rendering between IRIX and Windows NT* on page 77.

Distributed mental ray Rendering between IRIX and Windows NT

The mental ray rendering executable is able to distribute rendering tasks to multiple hosts. The machine which starts the rendering process (the rendering “client”) reads the scene from a text file which is either read from the disk or piped directly into mental ray by SOFTIMAGE 3D during interactive rendering. The client then sends the scene to the other machines that participate in the rendering (the rendering “slaves”). This works well when all the machines have the same architecture and all see the file system the same way (either by having exact duplicates on each machine or by having an NFS mounted network). However, this situation breaks down when the machines have different architecture and both the setup and the file name conventions are different.

Localization and File Naming Problems

The mental ray renderer must reference the shaders’ dynamic libraries and support files using an absolute path. This information is passed on to the rendering slaves by the rendering client. All the rendering slaves have to load the shaders’ dynamic libraries in order to render their part of the final image.

The problem is two-fold when a mixed Windows NT and IRIX network is involved for slave rendering: one problem is how each file system names its absolute paths and another problem is where SOFTIMAGE 3D is installed so that mental ray can locate the shaders.

File Naming

By convention, all file references that SOFTIMAGE 3D is generating in the scene description to mental ray (the MI file) follow the IRIX format. This ensures that all the rendering slaves interpret the path names the same way. Another reason is that the MI file is platform independent so that an MI file created using a Windows NT version of SOFTIMAGE 3D can be used on the IRIX version. The SOFTIMAGE 3D to mental ray translator also converts the file extensions to ensure cross-platform compatibility. The standard is in IRIX format, so the file extensions related to dynamic libraries and shader compilation are converted as follows:

`.dll → .so`

.lib → .a
.obj → .o

File Localization

The correspondence between the shaders' libraries on Windows NT and IRIX is made using the **linktab.ini** file mechanism since both versions are installed differently on both platforms. The installation script installs all the related components of SOFTIMAGE 3D in a single directory tree. This tree is pointed to by the environment variable **SI_LOCATION**. The following is the current directory structure (in part) after installation on a Windows NT machine:

```
$SI_LOCATION\3D\  
  \bin  
    \rsrc  
  \flexlm  
  \setup  
  \mental_ray  
    \bin  
    \inc  
    \lib  
    \man  
    \MR_Shaders
```

The default installation on Windows NT puts this tree in `c:\SOFT3D_3.51\3D` and in `/usr/softimage/3D` for the IRIX version.

Setup

All the Windows NT machines, the rendering client, and the rendering slaves which are used as part of the distributed rendering must have a **linktab.ini** file located in the directory pointed to by the **SI_LOCATION** environment variable.

The **linktab.ini** file is an ASCII file that is used to define one-to-one relations between IRIX and Windows NT paths. Each line of the **linktab.ini** file represents such a relationship and contains two entries. The first entry on the left is the Windows NT path, and the entry on the right is the IRIX equivalent.

On the Windows NT machine, make sure that the **SI_LOCATION**

environment variable points to the location where SOFTIMAGE 3D is installed, which is `c:\soft3d_3.51` by default. This variable should be present in the `%SI_LOCATION%\3D\bin\SETENV.BAT` file. To modify or see the contents of this file, click the Edit Setenv.bat icon in the SOFTIMAGE 3D 3.51 program group.

Create a file in the directory which `SI_LOCATION` points to and name it **linktab.ini**. This file contains path correspondence between Windows NT and IRIX for the current machines. In this file, type the following commands (each of these should be on one line):

```
C:/SOFT3D_3.51/mental_ray/MR_Shaders/  
                /usr/softimage/mental_ray/MR_Shaders/  
C:/SOFT3D_3.51/3D/bin/rsrc/usr/softimage/3D/rsrc
```

It is assumed that the installation of SOFTIMAGE 3D is located in `/usr/softimage`.

The first line ensures that the shaders' dynamic libraries are visible on all the platforms. The second ensures that the correspondence between the `rsrc` directory on IRIX and Windows NT, especially that the **softimage.mi** and **noIcon.pic** files are looked up correctly. The **noIcon.pic** is only used to replace pictures that are not found during rendering.

Repeat these steps for every Windows NT machine that you wish to include in the distributed mental ray rendering.

Windows NT Only Networks

The **linktab.ini** file is not needed on an homogeneous network of Windows NT machines for which the installation is identical. This means that each Windows NT machine has SOFTIMAGE 3D installed in exactly the same location on the same drive. If the default installation is used, then all machines have their own copy of the SOFTIMAGE 3D software located in `c:/soft3d_3.51`. If the drive or path name is changed on one machine, then all the machines must have a **linktab.ini** file similar to the previous sample to map each other's installation correctly and thus be able to perform mental ray distributed rendering.

While the **linktab.ini** file is not required for a homogeneous network, its presence does not make a difference to normal operation.

Debugging

On a Windows NT machine, generate an MI file that contains a default sphere with a material shader on it and examine its contents. (any material shader will do). All file names should be in IRIX format. If there is one path that is not in IRIX format, then there is a typographical error in the **linktab.ini** file. To generate an MI file, choose the **Render** menu cell in the Matter module. In the Render Setup dialogue box, select the mental ray renderer and select **Options**. The MI file generation controls are located in the lower right corner of the dialogue box.

Known Problems and Limitations

The following list of known problems and limitations applies to both the Windows NT and IRIX versions of SOFTIMAGE 3D version 3.51.

If a problem occurs for only one of the platforms, it is indicated as *Windows NT only* or *IRIX only* in the title.

The Interface

1. **Can't reduce enlarged window after GL Hardware render (*IRIX only*).**

After a GL Hardware render is completed, any window other than the Perspective window that has been enlarged does not return to its previous size (the view changes to Wireframe).

2. **Moving handle in proportional setup box causes black debris (*Windows NT only*).**

When moving the handle in the proportional setup box, black debris appears around the dialogue box.

3. **Rectangle not visible in rectangular zoom in rotoscopy shade view (*Windows NT only*).**

When doing a rectangular zoom (Shift-z) in the rotoscopy shaded view, the rectangle is not visible.

4. **There are no double frames to indicate keyable values (*Windows NT only*).**

There are no double frames around parameter buttons in dialogue boxes to indicate keyable values.

Modelling

5. **SOFTIMAGE 3D crashes with Proportional Edit on NURBS models.**

With some NURBS models, SOFTIMAGE 3D may crash if you move points with Proportional Edit selected (Consider Neighbor option activated), and do an Undo.

6. **UV Texture coordinates for cylindrical mapping gives erroneous results.**

If you assign UV coordinates from a cylindrical mapping type, there

is a visual difference in the renderings of both the original and UV texture mapped models.

7. Bad updates of envelope with effect inverse.

If you use relational modelling and the **Effect → Inverse** command on an envelope, the envelope is inversed each time it is refreshed.

8. Skeleton → Group as Skeleton and Reset Actor commands in the Actor module do not always work.

The **Skeleton → Group as Skeleton** and **Reset Actor** commands do not work on some models with a “Skel” label (in the Schematic view), and it might even crash in some cases.

9. Problems with extrusion with local axis or not from origin.

Extruding along a path gives incorrect results if the path is not z-axis aligned and at the origin. Also, it is important to extrude an object on a local axis.

10. Cannot change directories from within the Save → VRML browser.

In the **Save → VRML** dialogue box, click Cancel to exit the browser. You must also type in the absolute path in the Filename text box to save the model to the desired location.

11. Inversed normals of shape-animated objects flicker with relational modelling active.

When you play back an animation, the normals of a model change after each frame is refreshed, if the model with relational modelling has its normals inversed, and then shape animation applied to it.

12. When duplicating and applying symmetry to an envelope/inverse kinematic chain assembly, the envelope is not updated.

If you make a copy of an envelope/inverse kinematic chain assembly by either choosing **Effect → Symmetry** or **Duplicate**, the envelope is not updated when the skeleton is moved. This is because the FlexEnv relation, which is set to allow envelope deactivation in the Dopesheet, is not created for a duplicated envelope. Therefore, translating the duplicate envelope's inverse kinematic chain effector or rotating its joints does not deform the duplicate envelope: the duplicate envelope remains static.

Workaround: You must save a scene containing the duplicate envelope/inverse kinematic chain assembly (or use **Save → Elements** for this hierarchy) and load it the number of times required. You

must translate the duplicate envelope's inverse kinematic chain effector or rotate any of its joints to deform the duplicate envelope while keeping its symmetrically weighted assignments.

Animations/Expressions

- 13. SOFTIMAGE 3D may crash when deleting a model animated by clusters and expressions.**

SOFTIMAGE 3D may crash after deleting a model that has been assigned a cluster that has been animated by an expression.

- 14. Fast Playback option of previously deleted scene kept active (*Windows NT only*).**

The **Fast Playback** option in a window's layout dialogue box of a previously deleted scene is kept active: when playing back an animated scene using dynamic simulation, the fast playback does not show the animation.

Channels

- 15. Channels not allowing cycles.**

You cannot use I/O drivers as filters to control one function curve (such as rotx) based on another function curve from the same model (such as etrn).

Viewing

- 16. Shade view reloads textures already loaded in memory.**

When activating the texture display in the Shade view setup, SOFTIMAGE 3D loads all the textures in the scene even if the textures are already loaded.

- 17. Hardware texture display is not restricted to selected models only.**

When selecting the **Enable Hardware Texture** → **Affected Models** → **Selected** option in the Shade view Setup dialogue box, the textures are incorrectly displayed on all objects, not just on the selected objects.

Paint and Colour Reduction

18. **SOFTIMAGE 3D crashes when flooding the Paint Colour Index with large images (*Windows NT only*).**

Using the **Paint** menu cell, this occurs only if you have a very big image (for example, 1500 x 1500) and a low virtual memory (for example, 125 Mb).

Workaround: Increase the virtual memory (to 200 Mb).

19. **Name of functions in the Colour Reduction palette is badly displayed (*Windows NT only*).**

The name of the functions of the Colour Reduction palette is badly displayed. Instead of Edit, there is Cut, Paste, etc.

20. **Tiling or Repeats do not show in paint box.**

In the Paint dialogue box, if you are using tiling functions or repeats in UV coordinates, the resulting tile or repeats do not show up in the paint box.

Shaders

21. **The OZ Atmosphere material shader produces some grid artifacts (*Windows NT only*).**

The OZ Atmosphere material shader produces some line artifacts along the y-axis: the clouds are broken into clear demarcations by lines running perpendicular to the camera.

Workaround: Animate the clouds with a constant value of 0.0 to eliminate the streaks in the sky.

22. **The following shader presets are incorrect for SOFTIMAGE 3D (*Windows NT only*):**

Output shaders:

fur_body.out
fur_tail.out
fur_tailtip.out
fur_udder.out
fur_warrior.out

Volume shaders:

mf_bang.vol
mf_blue_neon.vol

mf_fly_thru.vol
mf_fog.vol
mf_mist.vol
mf_red_neon.vol
mf_spotlights.vol
mf_sunbeams.vol
ms_gitane.vol
ms_smokey_room.vol

This problem is most obvious with shaders such as `mf_red_neon.vol` where nothing appears red.

For a list of the correct preset shaders, see *Appendix A, Correct Preset Shaders – Windows NT Only* at the back of this guide.

Rendering

23. Problems with rendering subregions.

If the subregion resolution is 0 in x and/or y, SOFTIMAGE 3D crashes when previewing. You get a SOFTIMAGE internal error when rendering.

When rendering a subregion, the subregion resolution is not calculated if you disabled the drawing of the red rectangle (set **Subregion** to ON, **Draw** always set to OFF). This renders the object, but not the correct size.

When using the standalone renderer (**soft -R**) and specifying the subregion on the command line (using the **-b** option), the rendered image will be 0 x 0 pixels.

Workaround: Save your scene with the required subregion area, then save your scene and render using **soft -R** (do not use the **-b** option).

24. Missing message with mental ray renderer.

The red string “Done. Press middle mouse button” is not displayed properly in the preview window when rendering with mental ray mode.

25. Problem with middle-click to exit preview (*IRIX only*).

There is a problem closing the preview window when clicking the middle mouse button or pressing the Esc key. You must often click several times before you can exit the preview.

26. Motion blur with static blur causes mental ray to hang.

The problem occurs when you render using mental ray, and static blur and motion blur are both active. This leads to memory corruption which can have any side effect (mental ray hangs, SOFTIMAGE 3D crashes, etc.).

27. Invalid path set for “real_camera” shader.

For the real_camera shader, the browser for the custom lens file returns a file path string that contains a backward slash after the drive letter. The backward slash(/) causes mental ray to reject the file path for the custom lens as an invalid path.

Workaround: This backward slash (/) must be replaced with a forward slash (\) manually after clicking OK in the file browser.

28. Problems with hidden objects in mental ray.

No matter which frames are selected to be rendered in the Render Setup dialogue box, the translator looks at the current frame for any visible models.

If there is no geometry detected within a frame, mental ray stops and produces a syntax error without rendering the image.

29. Validation of cropping values is not good in special cases.

Validation of cropping values is not good, if for example, Y1 is set to the maximum value; value Y2 is incremented by 1 which is not a valid value. The same problem occurs in X. In the case of invalid cropping values, the SOFTIMAGE renderer uses the whole scene.

Rendering Limitations

30. System limitation on Impacts related to performing interactive rendering or previewing with mental ray (IRIX only)

Note This limitation does not affect non-IGLOO platforms, the interactive SOFTIMAGE 3D preview/renderer, or the standalone renderers (both SOFTIMAGE and mental ray).

On IGLOO-based machines, such as Impacts and O2s, there is an operating system limitation in which the number of interactive mental ray previews/renderers that you can perform is limited by the size of a file descriptor table which IRIX maintains (this limit varies with each IGLOO-based platform). SOFTIMAGE 3D dynamically detects when this limit is about to be reached. When this happens, the interactive mental ray preview/render is disabled and this

message is displayed:

“System limit achieved. Save and restart soft.”

You will not be able to do any further interactive mental ray previewing or rendering. If you wish to do so, you should save the currently edited scene and then restart SOFTIMAGE 3D.

Workaround: The default detection and recovery should be enough to avoid disrupting the normal workflow. However, in intensive rendering and previewing sessions, the limit defined by the IRIX file descriptor table may not be enough. You can change the default limit by using the `SI_RLIMIT_NOFILE_MAX` environment variable. For example, the following command increases the limit of number of files in the file table to 1750:

```
setenv SI_RLIMIT_NOFILE_MAX 1750
```

The maximum number of files allowed by the system is 2500.

Standalones

31. dbtools and evalScene standalones look for incorrect resources path (*IRIX only*).

dbtools and evalScene look for the resources in the path:

```
/usr/softimage/Creative/rsrc/Relations.rsrc
```

instead of:

```
/usr/softimage/3D
```

which is the default path.

32. mb standalone does not handle nested instances.

The **mb** standalone does not handle instances that are inside a hierarchy (nested instances).

Particle

33. Node lattice animated objects as 3D source.

Path or surface-deformed objects, such as Obstacles, freezes Particle or makes it run slowly.

34. 3D source object with unfrozen centres don't emit particles correctly (*Windows NT only*).

3D objects with a centre that is not frozen emit particles with an abnormally fast speed.

35. **Copy and Paste functions in Curve Editor do not work properly (*Windows NT only*).**
36. **Deleting keyframes in the Parameter Editing area for an fcurve with less than two keyframes crashes Particle (*Windows NT only*).**
Workaround: Delete keyframes directly from within the Curve Editor instead.
37. **Product Version of particle.exe incorrect (*Windows NT only*).**
In the Properties for the Particles executable, it states that the Product Version as 1.5.0.1. This is incorrect and should be version 3.51.
38. **3D selection not active.**
3D selection is not active in the viewer.
Workaround: Perform selections from the different browsers.
39. **Status message box is not available with this version.**
40. **Metaclay doesn't appear when loaded in Particle.**
In File card → 3DScene, if you click on Load and select a 3D scene that contains a metaclay object, the object doesn't appear in the scene, but the name of the 3D scene appears in the 3D scene name text box.
41. **Minimize Supra key Alt -F9 is ineffective (*Windows NT only*).**

Supra key F9 is difficult to activate.

DXF and IGES Converters

dx2soft

The supported version of **dx2soft** is Release 11 or earlier. When creating a file to be imported into SOFTIMAGE 3D, use the proper export format.



dx2soft does not support Release 13.

The -G and -s options try to perform an optimization of the polygons being imported: there are no guarantees that these options will work.

Some DXF files produce as much as one model per triangle, which is

highly inefficient. This is usually due to the exporting software itself, which does not support the concept of models.

Workaround: Use the **-s** and **-G** options with the above limitation in mind.

Another workaround is to reorder the polygon inside SOFTIMAGE 3D by using the **Effect → Cleanup** and **Effect → Merge** commands in the Model module.

iges2soft

The supported version of **iges2soft** is Release 5.1 or earlier. When creating a file to be imported into SOFTIMAGE 3D, use the proper export format.

Parametric entities (141 - 144) don't import well.

Workaround: When exporting the scene from another software, specify that parametric surfaces should not be created, if possible.

Known Documentation Errors

Working with SOFTIMAGE 3D User's Guide

On page 92, the expression written in step 6:

```
( this.mdltyp == MOD ) ! ( this.etrnz <= near.etrnz ...
```

is missing the ampersand (&) before the exclamation mark (!).

The expression should read like this:

```
( this.mdltyp == MOD ) &! ( this.etrnz <= near.etrnz ...
```

Rendering User's Guide

- On page 4, it should be stated that you can render in *any* window with the mental ray renderer. Only the SOFTIMAGE renderer is limited to using the Perspective view of window B.
- On page 5, changing the angle of the camera lens has the same effect for either the **Automatic** or **Custom Angle** option: the angle is always automatically calculated. Also, if you zoom in to a scene, the angle is reduced, not increased.
- On page 19, the extensions of output files available for the mental ray renderer should also include the **.pic** extension.
- On page 21, this information needs to be added to the note about hardware rendering: "The hardware renderer also takes a snapshot

of the screen when you shift to another Windows NT application. This will ruin your rendering because some of the images will be replaced by the background of the application to which you shifted.”

- On pages 24, 25, and 30, it should be noted that when you use the standalone SOFTIMAGE renderer (**soft -R**), the **soft** alias is the standard one defined in your **.softimage** file. This means that the path to it is:
/usr/softimage/3D/bin/soft /usr/softimage/3D/rsrsc
- On page 43, it is not stated that shadows of moving objects are not blurred by the **mb** (motion blur) standalone.

Using Tools User's Guide

On page 59, the heading titled “SOFTIMAGE 3D to 3D Studio” for the **soft3ds** standalone is incorrect. Instead, this should be part of the section called “3DS to SOFTIMAGE 3D” on page 47 of the same guide. This is because the **soft3ds** standalone converts 3D Studio files to SOFTIMAGE 3D, not vice versa. See also *Get → 3DS2Soft Is Now Import → Objects → 3DStudio* on page 23 for more information on converting 3D Studio files to SOFTIMAGE 3D.

Particle User's Guide

On page 18 (for Windows NT only), the browser dialogue box has changed since the screen grab was taken. The browser can now access the chapters (Systems or Scene) directly, so the buttons shown are Open and Cancel instead of OK and Network.

C H A P T E R T W O

What's New – Windows NT® Only

The Interface

Using a Two-Button Mouse

If you are using a two-button mouse and you want to simulate the action of a three-button mouse, press the Ctrl key on the keyboard and click the right and left mouse buttons.

Scrolling in the Command Prompt Window

When using the standalones available in SOFTIMAGE 3D 3.51, you can read their command line options by typing the name of the standalone in the command prompt window on Windows NT.

However, the list may scroll past your view very quickly, so to read the options you have to set the command prompt options as follows:

1. From the Main group in the Program Manager, double-click the Command Prompt icon. The command prompt window opens.
2. Click in the upper-left corner to open the menu for the Command Prompt.
3. Click Properties and choose the Screen Size and Position tab.
4. Change the Height of the Screen Buffer Size to 900 and press Enter to accept that value.
5. Click the General Settings tab and select the QuickEdit Mode. Press Enter to accept.

This places a scroll bar on the side of the command prompt window that allows you to scroll up and down the window.

Note

For best results, you should not have any programs running in the “Always on Top” mode (such as the Microsoft Office tool bar located in the upper right corner of the screen). This seems to interfere with SOFTIMAGE 3D’s refresh ability.

Channels

Refer to *Using Remote Drivers on Windows NT* on page 16 for more information on this subject.

Tools and Standalones

Film_Out → Solitaire

The Solitaire image recorder is no longer supported. As a result, the **Film_Out → Solitaire** command in the Tools module has been removed from the interface.

Print

The **Letter** and **Legal** options for **Paper Format** have been removed from the Print dialogue box. The **Paper Format** option has been hard coded to **Letter**.

Video → Abekas → Record

The **Mask** and **True Black** options in the Record to Abekas dialogue box are not supported for this version.

AVI2soft

The AVI2soft standalone converts AVI picture files (.avi) into SOFTIMAGE 3D picture files.

Usage

AVI2soft <input file> <output file>

[-s <start frame> <end frame> <step>]

<input file> is the name of the AVI picture file to be converted.

<output file> is the name of the SOFTIMAGE 3D picture file to be created.

Options

-s Converts a sequence of frames.

Picture files in the AVI file format begin at 0 and end at <number of frames -1>.

dib2soft

The **dib2soft** standalone converts DIB picture files (**.dib**) into SOFTIMAGE 3D picture files (**.pic**).

Usage

```
dib2soft <input file> <output file>
```

```
[-s <start frame> <end frame> <step>] [-v]
```

<input file> is the name of the DIB picture file (**.dib**). If no extension is specified, **.dib** is assumed.

<output file> is the name you want to give to the SOFTIMAGE 3D picture file (**.pic**). If no extension is specified, the output file is given a **.pic** extension.

Options

-s Converts a sequence of frames.

-v Turns on the verbose mode.

imf_copy and imf_displ

The **imf_copy** and **imf_disp** mental ray standalones are not available on SOFTIMAGE 3D 3.51 Windows NT.

imginfo

The **imginfo** standalone on Windows NT does not recognize input images without signatures or file extensions. As a result, the **-format <name>** option has been added to explicitly state the image format.

soft2AVI

The **soft2AVI** standalone converts SOFTIMAGE 3D picture files (**.pic**) into AVI picture files (**.avi**).

Usage

```
soft2AVI <input file> <output file>
```

```
[-r <frame rate>] [-s <start frame> <end frame> <step>]  
[-f <sequence width>]
```

<input file> is the name of the SOFTIMAGE 3D picture file (**.pic**).

<output file> is the name you want to give to the AVI picture file (**.avi**).

Options

- r Specifies the number of frames per second. The default number is 30.
- s Converts a sequence of frames.
- f Defines the number of digits contained in each frame number. The default number is 1.

Examples

The following example shows the conversion of a SOFTIMAGE 3D picture file (foo.0001.pic) to AVI format.

```
soft2AVI foo -s 1 10 1 -f 4
```

soft2dib

The **soft2dib** standalone converts SOFTIMAGE 3D picture files (.pic) into DIB picture files (.dib).

Usage

```
soft2dib <input file> <output file>
```

```
[-s <start frame> <end frame> <step>] [-v]
```

<input file> is the name of the SOFTIMAGE 3D picture file (.pic). If no extension is specified, .pic is assumed.

<output file> is the name you want to give to the DIB picture file (.dib). If no extension is specified, the output file is given a .dib extension.

Options

- s Converts a sequence of frames.
- v Turns on the verbose mode.

Watch Standalone Custom Effect

The Watch standalone custom effect is not supported for SOFTIMAGE 3D 3.51 Windows NT.

Custom Effects

The DDSound, Splound, and Watch custom effects are not supported for SOFTIMAGE 3D 3.51 for Windows NT.

C H A P T E R T H R E E

What's New – IRIX Only

Channels

Sidewinder Driver

The sidewinder driver outputs degrees of freedom corresponding to the movement of the joystick and buttons on the Sidewinder.

To activate the sidewinder channel driver, choose the **Channel** → **Channel Setup** command in the Motion module. The Channel Setup dialogue box appears.

To add the sidewinder driver to the Driver list, click **New** and select the driver from the browser.

Active	Type	Driver	RunMode	Machine
on	I	mouse		Local

CHANNELS:

Active	Type	Channel	Offset	Scaling
on	I	mouse.xpos	0.000	1.000
on	I	mouse.ypos	0.000	1.000

CONNECTIONS:

Active	Source	Destination
--------	--------	-------------

The driver has also ramping capabilities that you can control with three floating point values. The **Global attack time** value is the time, in seconds, in which the degree of freedom passes from 0.0 (base value) to 1.0 (maximum value). When the key is released, the **Global delay time** specifies how long the degree of freedom retains its value. Then the degree of freedom value is ramped down to 0.0 in the time

specified by the **Global decay time**.

Degrees of Freedom

joyx
joyy
rudder
slider
button_1
button_2
button_3
button_4
hat_up
hat_down
hat_left
hat_right
blink

Custom Shaders

All user-developed SOFTIMAGE 3D version 3.0 (IRIX) shaders need to be recompiled with the mental ray toolkit version 3.51 to be able to run in SOFTIMAGE 3D version 3.51.

New Sounds

There are two new environment variables for SOFTIMAGE 3D 3.51. You can now edit the sounds your SGI makes on those few occasions when it crashes. Set the environment variables to point to your favorite .aiff, .aifc, .wav, or Next/Sun sound files.

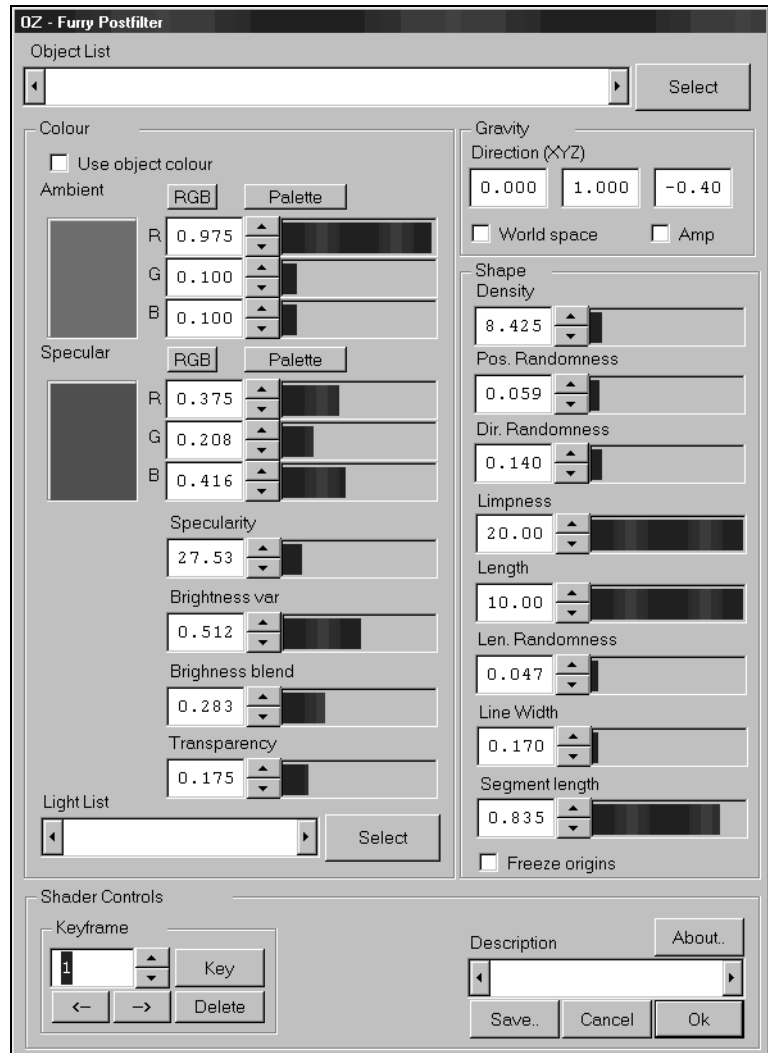
Environment Variable	Description
SI_SOUND_CRASH_BLAMMO	The sound the machine makes when it crashes.
SI_SOUND_CRASH_SAVE_OK	The sound the machine makes when emergency save is enabled.

A P P E N D I X A

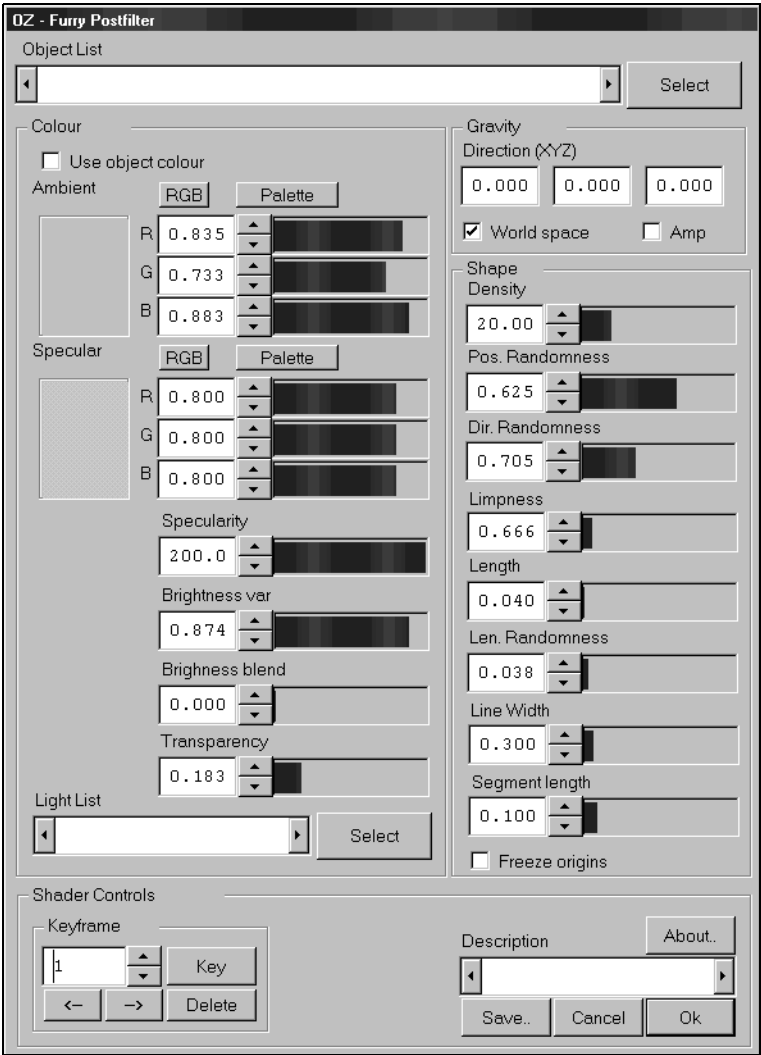
Correct Preset Shaders – Windows NT® Only

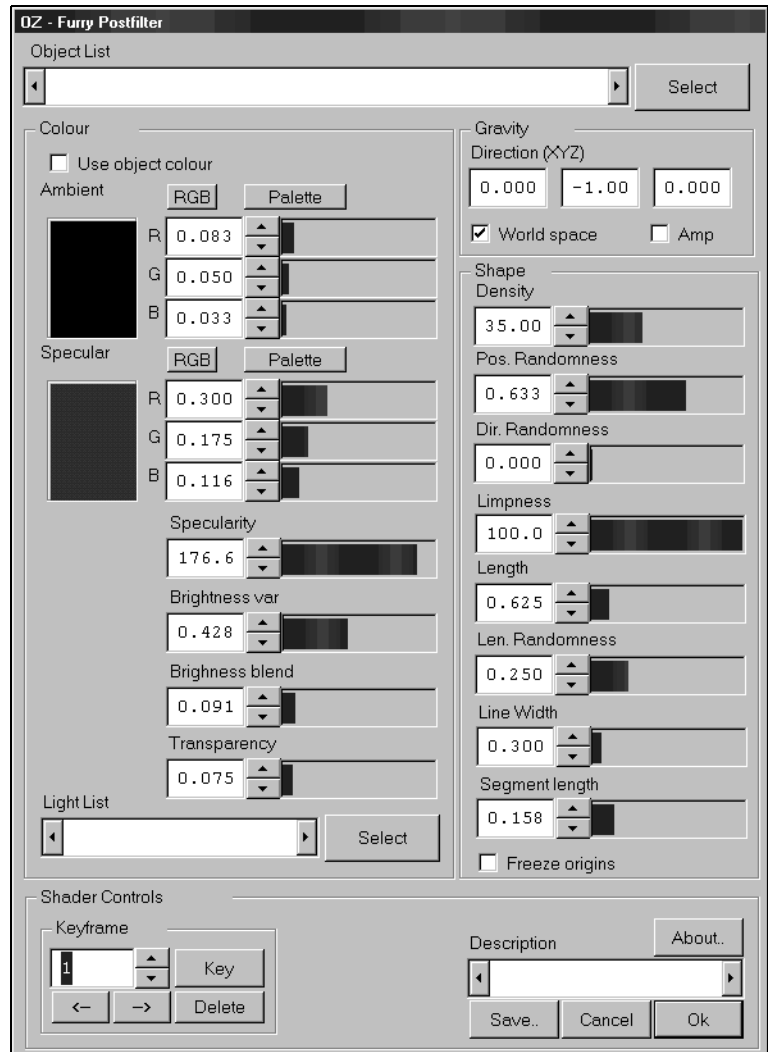
Output Shaders

Fur_warrior

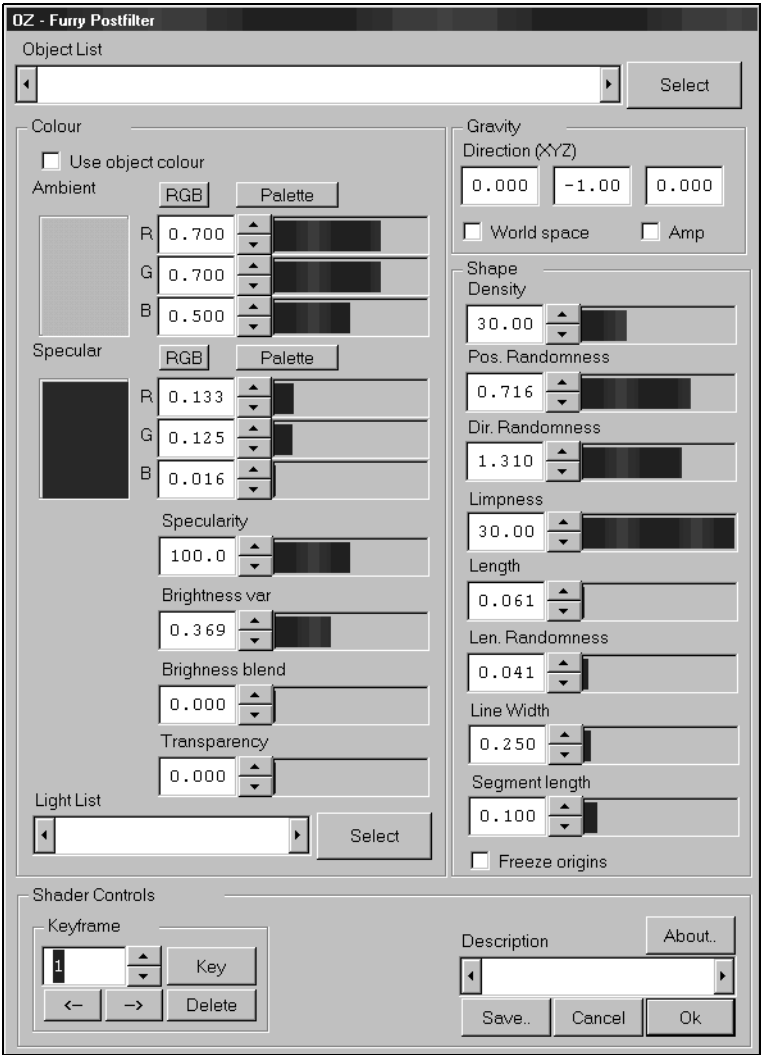


Fur_udder

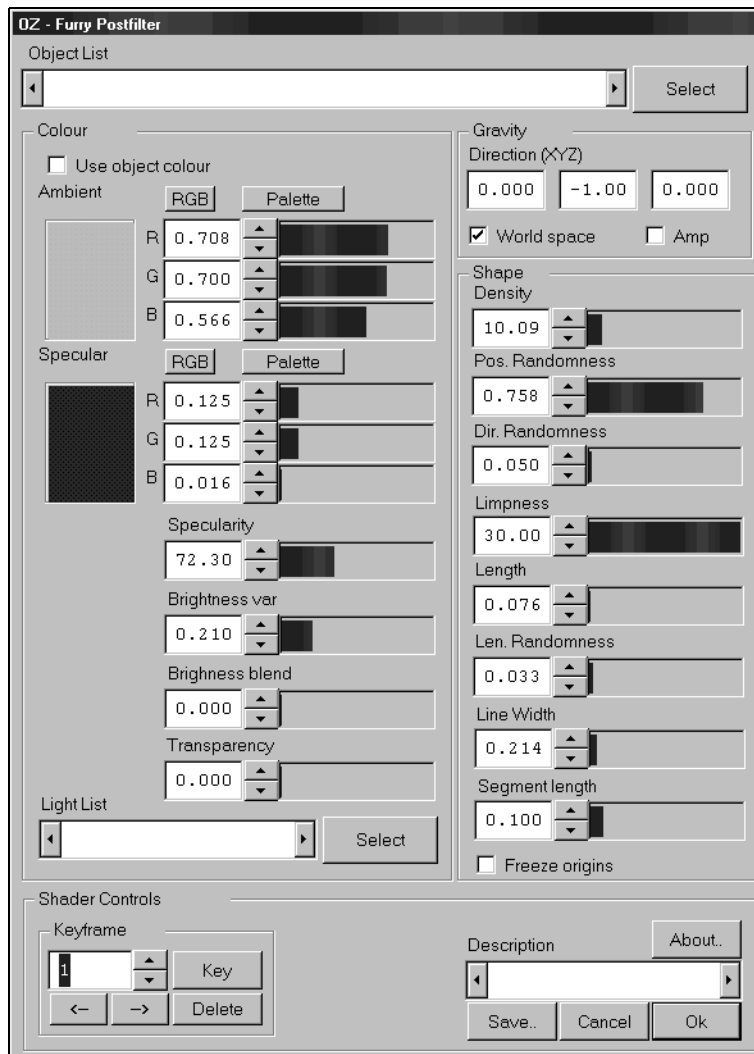


Fur_tailtip

Fur_tail



Fur_body



Volume Shaders

MS_smoky_room

02 - Raymarcher - Smoke Column

Marching

0.1

Step Size

0.20

Maximum Difference

3

Maximum Subdivisions

0.00

Jitter

Scattering

Lights

Select

Shadowed Lights

Select

0.50

Asymmetry

☐ Uniform

Ambience

RGB

Palette

R

1.000

G

0.895

B

0.671

Absorption

RGB

Palette

R

0.000

G

0.000

B

0.000

☐ Volumic Density

0.04

Density Scale

☐ Transparent

Transformation Model

Select

☒ Move With Geometry

Bounding Box

MIN

MAX

X

-20.000

20.0000

Y

0.00000

20.0000

Z

-20.000

20.0000

☐ Always Honour

Proximity

☐ On

20.

Intensity Scale

30.

Cone Angle

Fractal Parameters

0

Noise Type

0.100

Minimum Level

☐ Absolute

0.707

Decay

1

Maximum Iterations

2.000

Frequency Multiplier

Turbulence

Amount

7.024

1.000

Scale

10.00

Altitude

0.200

Bias

Column Shape

15.00

End

0.100

Bias

1.000

Rise Speed

10.00

Radius

12.00

Time Offset

2.000

Thickness

1.000

Start Length

2.000

Texture Amplitude

Swirl

Amount

0.741

1.000

Change Rate

1.000

Scale

Wind

0.200

Speed

10.00

Altitude

Colour

RGB

Palette

R

0.316

G

0.232

B

0.135

Shaderball

Preview

☐ Auto Refresh

Keyframe

1

Insert

Prev

Next

Remove

User Comment

Help

Save

Cancel

Ok

MS_gitane

OZ - Raymarcher - Smoke Column

Marching

0.1 Step Size 0.10 Maximum Difference

4 Maximum Subdivisions 0.70 Jitter

Scattering

Lights Select

Shadowed Lights Select

0.50 Asymmetry ☐ Uniform

Ambience RGB Palette

R 0.753 G 0.888 B 1.000

Absorption RGB Palette

R 0.000 G 0.000 B 0.000

☐ Volumic Density 2.00 Density Scale

☐ Transparent

Transformation Model

Select

☒ Move With Geometry

Proximity

☐ On

20. Intensity Scale

30. Cone Angle

Bounding Box

	MIN	MAX
X	-2.0000	2.00000
Y	0.00000	20.0000
Z	-2.0000	2.00000

☐ Always Honour

Fractal Parameters

1 Noise Type 0.100 Minimum Level

☐ Absolute 0.707 Decay

1 Maximum Iterations 2.000 Frequency Multiplier

Turbulence

Amount 1.900 Scale 1.000

10.00 Altitude 0.200 Bias

Column Shape

15.00 End 0.100 Bias

1.000 Rise Speed

12.00 Time Offset

0.500 Radius

0.300 Thickness

15.00 Start Length

0.050 Texture Amplitude

Swirl

Amount 0.205

1.000 Change Rate

1.000 Scale

Wind

0.300 Speed

15.00 Altitude

Colour RGB Palette

R 0.116 G 0.406 B 0.929

Shaderball

Preview

☐ Auto Refresh

Keyframe

1 Insert

Prev Next Remove

User Comment

Help Save Cancel Ok

MF_sunbeams

0.3

Step Size

0.20

Maximum Difference

0

Maximum Subdivisions

0.50

Jitter

Lights

Select

Shadowed Lights

Select

0.50

Asymmetry☐ Uniform

Ambience

RGB

Palette

R

0.000

G

0.000

B

0.000

Absorption

RGB

Palette

R

0.000

G

0.000

B

0.000

☐ Volumic Density

0.50

Density Scale

☐ Transparent

Transformation Model

Select

☒ Move With Geometry

Bounding Box

MIN

MAX

X

-15.000

15.0000

Y

0.00000

15.0000

Z

-15.000

15.0000

☐ Always Honour

Proximity

☐ On

20.

Intensity Scale

30.

Cone Angle

Fractal parameters

1

Noise Type

0.807

Minimum Level

☐ Absolute

0.707

Decay

10

Maximum Iterations

2.000

Frequency Multiplier

3.000

Fractal Size

0.000

Threshold

0.700

Amplitude

0.000

Diffusion

0.300

Offset

Falloff

☒ None

☐ Y

☐ Radius

1.000

Distance

0.500

Bias

Animation

☒ Use Time

1.000

Time Scaling

Colour 0

RGB

Palette

R

0.000

G

0.000

B

0.000

Colour 1

RGB

Palette

R

0.707

G

0.535

B

0.285

Shaderball

Preview

☐ Auto Refresh

Keyframe

1

Insert

Prev

Next

Remove

User Comments

Help

Save

Cancel

Ok

MF_spotlights

OZ - Raymarcher - Fractal

Marching

Step Size: 0.3 Maximum Difference: 0.20

Maximum Subdivisions: 4 Jitter: 0.50

Scattering

Lights: [] Select

Shadowed Lights: [] Select

Asymmetry: 0.50 ☐ Uniform

Ambience RGB Palette

R: 0.000 G: 0.000 B: 0.000

Absorption RGB Palette

R: 0.000 G: 0.000 B: 0.000

☐ Volumic Density Density Scale: 0.40

☐ Transparent

Transformation Model

[] Select

☒ Move With Geometry

Proximity

☒ On

Intensity Scale: 30.

Cone Angle: 30.

Bounding Box

	MIN	MAX
X	-10.000	10.0000
Y	0.00000	20.0000
Z	-10.000	10.0000

☐ Always Honour

Fractal parameters

Noise Type: 0 Minimum Level: 0.100

☐ Absolute Decay: 0.707

Maximum Iterations: 1 Frequency Multiplier: 2.000

Fractal Size: 1.000 Threshold: 0.000

Amplitude: 0.000 Diffusion: 0.000

Offset: 0.500

Animation

☐ Use Time

Time Scaling: 1.000

Falloff

☒ None

☐ Y

☐ Radius

Distance: 1.000

Bias: 0.500

Colour 0 RGB Palette

R: 0.000 G: 0.000 B: 0.000

Colour 1 RGB Palette

R: 0.957 G: 0.964 B: 0.957

Shaderball

Preview

☐ Auto Refresh

Keyframe

1 Insert

Prev Next Remove

User Comments

[]

Help Save Cancel Ok

MF_red_neon

0Z - Raymarcher - Fractal

Marching

4.0

Step Size

0.20

Maximum Difference

3

Maximum Subdivisions

0.00

Jitter

Scattering

LightsSelect

Shadowed LightsSelect

0.50

Asymmetry

☐ Uniform

Ambience

RGB

Palette

R

0.000

G

0.000

B

0.000

Absorption

RGB

Palette

R

0.000

G

0.000

B

0.000

☐ Volumic Density

0.50

Density Scale

☒ Transparent

Transformation ModelSelect

☒ Move With Geometry

Proximity

☒ On

20.

Intensity Scale

30.

Cone Angle

Bounding Box

MIN

MAX

X

-50.000

50.0000

Y

-50.000

50.0000

Z

-50.000

50.0000

☐ Always Honour

Fractal parameters

0

Noise Type

0.100

Minimum Level

☐ Absolute

0.707

Decay

1

Maximum Iterations

2.000

Frequency Multiplier

1.000

Fractal Size

0.000

Threshold

0.000

Amplitude

0.000

Diffusion

0.500

Offset

Falloff

☒ None

☐ Y

☐ Radius

1.000

Distance

0.500

Bias

Animation

☐ Use Time

1.000

Time Scaling

Colour 0

RGB

Palette

R

0.106

G

0.113

B

0.000

Colour 1

RGB

Palette

R

1.000

G

0.035

B

0.000

Shaderball

Preview

☐ Auto Refresh

Keyframe

1

Insert

Prev

Next

Remove

User Comments

Help

Save

Cancel

Ok

A-12

Release Notes

SOFTIMAGE 3D

MF_mist

OZ - Raymarcher - Fractal

Marching

Step Size: 0.5 Maximum Difference: 0.20

Maximum Subdivisions: 3 Jitter: 0.00

Scattering

Lights: [] Select

Shadowed Lights: [] Select

Asymmetry: 0.50 ☐ Uniform

Ambience RGB Palette

R: 0.000 G: 0.000 B: 0.000

Absorption RGB Palette

R: 0.000 G: 0.000 B: 0.000

☐ Volumic Density Density Scale: 0.70

☐ Transparent

Transformation Model

[] Select

☒ Move With Geometry

Bounding Box

	MIN	MAX
X	-20.000	20.0000
Y	0.00000	6.00000
Z	-20.000	20.0000

☒ Always Honour

Proximity

☐ On

Intensity Scale: 20.

Cone Angle: 30.

Fractal parameters

Noise Type: 1 Minimum Level: 0.100

☐ Absolute Decay: 0.707

Maximum Iterations: 3 Frequency Multiplier: 2.000

Fractal Size: 1.000 Threshold: 0.000

Amplitude: 2.000 Diffusion: 0.000

Offset: 0.500

Animation

☒ Use Time

Time Scaling: 1.000

Falloff:
☐ None
☐ Y
☒ Radius
Distance: 15.00
Bias: 0.500

Colour 0 RGB Palette

R: 0.000 G: 0.000 B: 0.000

Colour 1 RGB Palette

R: 0.211 G: 0.556 B: 0.558

Shaderball

Preview

☐ Auto Refresh

Keyframe

1 Insert Prev Next Remove

User Comments

[]

Help Save Cancel Ok

MF_fog

0.5

Step Size

0.20

Maximum Difference

3

Maximum Subdivisions

0.00

Jitter

Lights

Select

Shadowed Lights

Select

0.50

Asymmetry

☐ Uniform

Ambience

RGB

Palette

R

0.055

G

0.074

B

0.044

Absorption

RGB

Palette

R

0.000

G

0.000

B

0.000

☐ Volumic Density

0.02

Density Scale

☐ Transparent

Transformation Model

Select

☒ Move With Geometry

Proximity

☐ On

20.

Intensity Scale

30.

Cone Angle

Bounding Box

MIN

MAX

X

-30.000

30.0000

Y

-1.0000

30.0000

Z

-30.000

30.0000

☒ Always Honour

Fractal parameters

0

Noise Type

0.100

Minimum Level

☐ Absolute

0.707

Decay

2

Maximum Iterations

2.000

Frequency Multiplier

1.000

Fractal Size

0.000

Threshold

1.000

Amplitude

0.000

Diffusion

0.500

Offset

Falloff

☒ None

☐ Y

☐ Radius

1.000

Distance

0.500

Bias

Animation

☒ Use Time

2.000

Time Scaling

Colour 0

RGB

Palette

R

0.000

G

0.000

B

0.000

Colour 1

RGB

Palette

R

0.100

G

0.216

B

0.227

Shaderball

Preview

☐ Auto Refresh

Keyframe

1

Insert

Prev

Next

Remove

User Comments

Help

Save

Cancel

Ok

MF_fly_thru

OZ - Raymarcher - Fractal

Marching

Step Size: 0.7 Maximum Difference: 0.20

Maximum Subdivisions: 3 Jitter: 0.08

Scattering

Lights: [] Select

Shadowed Lights: [] Select

Asymmetry: 0.50 ☐ Uniform

Ambience RGB Palette

R: 0.477 G: 0.537 B: 0.522

Absorption RGB Palette

R: 0.000 G: 0.000 B: 0.000

☐ Volumic Density 0.03 Density Scale

☐ Transparent

Transformation Model

[] Select

☒ Move With Geometry

Bounding Box

	MIN	MAX
X	-50.000	50.0000
Y	-1.0000	50.0000
Z	-50.000	50.0000

☐ Always Honour

Proximity

☐ On

Intensity Scale: 20.

Cone Angle: 30.

Fractal parameters

Noise Type: 1 Minimum Level: 0.100

☐ Absolute Decay: 0.707

Maximum Iterations: 3 Frequency Multiplier: 2.000

Fractal Size: 8.000 Threshold: 0.000

Amplitude: 2.000 Diffusion: 0.000

Offset: 0.500

Animation

☐ Use Time

Time Scaling: 1.000

Falloff

☐ None

☐ Y

☒ Radius Distance: 50.00 Bias: 0.500

Colour 0 RGB Palette

R: -1.00 G: -1.00 B: -1.00

Colour 1 RGB Palette

R: 0.757 G: 0.807 B: 0.835

Shaderball

Preview

☐ Auto Refresh

Keyframe

1 [] Insert

Prev Next Remove

User Comments

[]

Help Save Cancel Ok

MF_blue_neon

0.5

Step Size

0.20

Maximum Difference

3

Maximum Subdivisions

0.00

Jitter

Lights

Select

Shadowed Lights

Select

0.50

Asymmetry☐ Uniform

Ambience

RGB

Palette

R

0.000

G

0.000

B

0.000

Absorption

RGB

Palette

R

0.000

G

0.000

B

0.000

☐ Volumic Density

0.50

Density Scale

☐ Transparent

Transformation Model

Select

☒ Move With Geometry

Bounding Box

MIN

MAX

X

-50.000

50.0000

Y

-50.000

50.0000

Z

-50.000

50.0000

☐ Always Honour

Proximity

☐ On

20.

Intensity Scale

30.

Cone Angle

Fractal parameters

0

Noise Type

0.100

Minimum Level

☐ Absolute

0.707

Decay

1

Maximum Iterations

2.000

Frequency Multiplier

1.000

Fractal Size

0.000

Threshold

1.000

Amplitude

0.000

Diffusion

0.500

Offset

Falloff

☒ None

☐ Y

☐ Radius

1.000

Distance

0.500

Bias

Animation

☐ Use Time

1.000

Time Scaling

Colour 0

RGB

Palette

R

0.000

G

0.000

B

0.000

Colour 1

RGB

Palette

R

1.000

G

0.900

B

0.800

Shaderball

Preview

☐ Auto Refresh

Keyframe

1

Insert

Prev

Next

Remove

User Comments

Help

Save

Cancel

Ok

MF_bang

OZ - Raymarcher - Fractal

Marching

1.0 Step Size 0.20 Maximum Difference

3 Maximum Subdivisions 6.90 Jitter

Scattering

Lights Select

Shadowed Lights Select

0.50 Asymmetry ☐ Uniform

Ambience RGB Palette

R 0.000

G 0.000

B 0.000

Absorption RGB Palette

R 0.000

G 0.000

B 0.000

☒ Volumic Density 0.51 Density Scale

☐ Transparent

Transformation Model

Select

☒ Move With Geometry

Bounding Box

	MIN	MAX
X	-50.000	50.0000
Y	-50.000	50.0000
Z	-50.000	50.0000

☐ Always Honour

Proximity

☐ On

20. Intensity Scale

30. Cone Angle

Fractal parameters

1 Noise Type 0.100 Minimum Level

☐ Absolute 0.707 Decay

3 Maximum Iterations 2.000 Frequency Multiplier

1.000 Fractal Size 0.000 Threshold

1.000 Amplitude 0.000 Diffusion

0.500 Offset

Animation

☒ Use Time

1.460 Time Scaling

Falloff

☐ None

☐ Y

☒ Radius

11.90 Distance

0.626 Bias

Colour 0 RGB Palette

R 0.286

G 0.129

B 0.000

Colour 1 RGB Palette

R 0.608

G 0.034

B 0.000

Shaderball

Preview

☐ Auto Refresh

Keyframe

1 Insert

Prev Next Remove

User Comments

Help Save Cancel Ok

