Chapter 2: User Interface

2.1 Spaces and Views

The purpose of trueSpace is to help you enter, navigate, edit, and create 3D **Spaces** (or scenes) and objects inside them. In order to make these spaces visible to human eyes, trueSpace provides a variety of 2D and 3D **Views** as well as 2D and 3D **Controls** to interact with the 3D objects you create.

2.1.1 Spaces

Spaces and the objects inside them exist independently of the user and his views. When you enter and edit objects in a space, you or other users visiting the same space later will see it permanently changed.

In most 3D authoring programs, 3D spaces can only be stored on your local hard drive and are accessible only to you. In trueSpace they can be stored anywhere on the web, allowing multiple participants to interact with them. These are called shared spaces. You can interact with the objects in a shared space in the same ways as those in local spaces. Even better, you can interact with other participants inside shared spaces, allowing you to collaborate with your partners or customers.

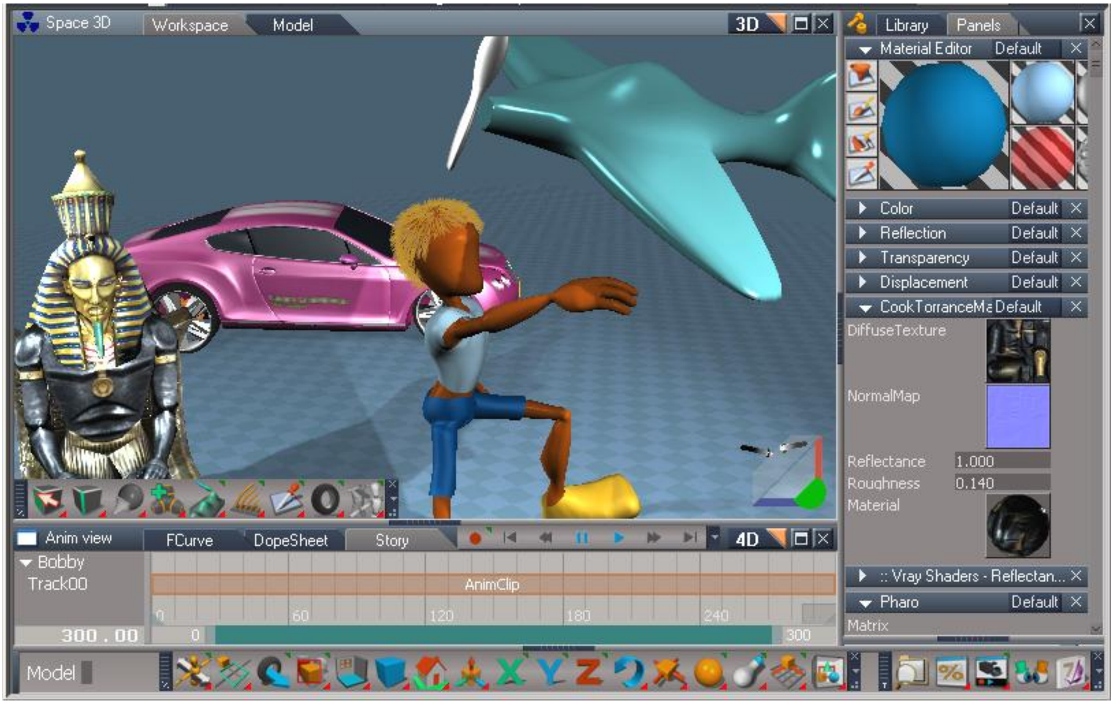
2.1.2 Views

Views make 3D spaces accessible to human senses, mostly to your eyes. Unlike spaces, which exist for many users, views belong just to you. They are considered local to your machine. When two users enter the same space, one could view it in wireframe form while others could view as solid if they desire. While the space is the same, individual/local views of it are not.

Views are your window on the objects you are working with. Just as Microsoft Word provides multiple views of your text data, so trueSpace provides multiple views of your scene and the objects in it. Using views, you can navigate through trueSpace as you work. In trueSpace, being in a “space” means literally entering an object where work will be done. In this way, you can expand your local or shared “universe” by creating new objects and entering them. Perhaps most are familiar with Stacking Dolls, where many dolls are each encapsulated inside a larger doll. trueSpace objects and spaces are composed with similar encapsulation of objects. Views allow you to navigate through your objects, allowing you to explore, learn and create.

2.1.3 Layouts

On your desktop you can open multiple views, each with its own toolbars and libraries, and arrange them into **layouts**. It is usually desirable to have more than one view open simultaneously. For example, you may want to have both a 3D view and a 2D view of the same space open for modeling and behavior modification, respectively. You may also want to have access to panels containing various tools and libraries alongside the views with which you are working.



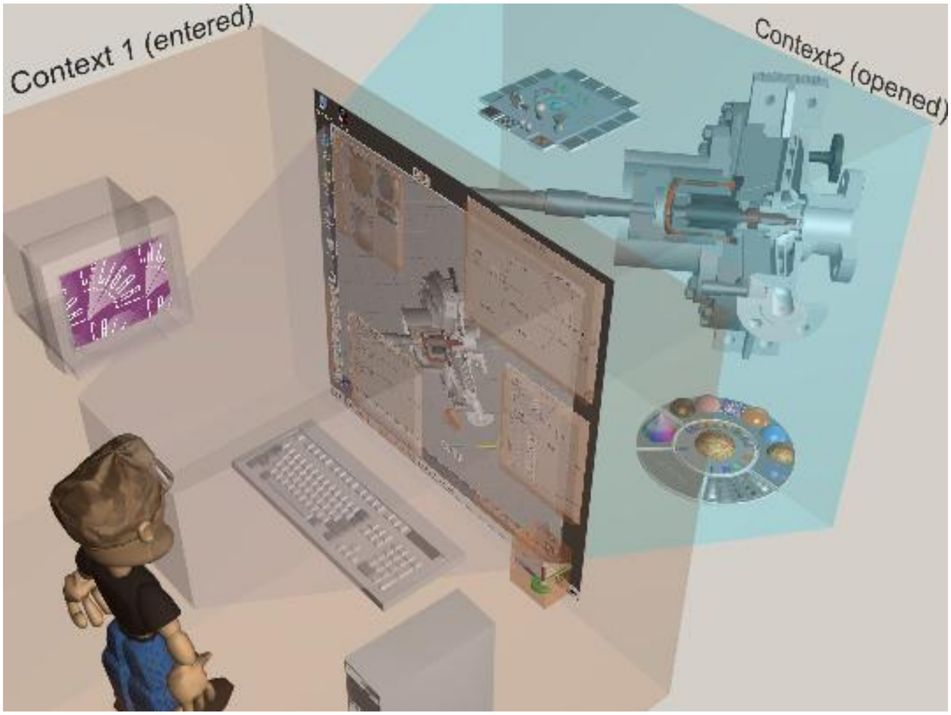
Reflectance 1.000
Roughness 0.140
300 . 00

Animation Layout with several Views opened

You can quickly change the layout of the desktop to best suit the way you work. To access a preset layout, just choose one from the Layouts library or from Tabs on title bar. You can also change the layout by opening additional panels and views such as 1D (Script Editor) views, 2D (Link Editor) views, or 3D Workspace or Model views. You can overlap windows or dock them, and easily save new layouts to the layout library and title bar tabs.

2.1.4 Contexts (Projects)

A **Context** is the state of a space in which you are interacting, together with your preferences and layouts. In addition to the space itself and your presence within it, a context includes the rules of behavior for objects and participants. The image below sums up some of these concepts:



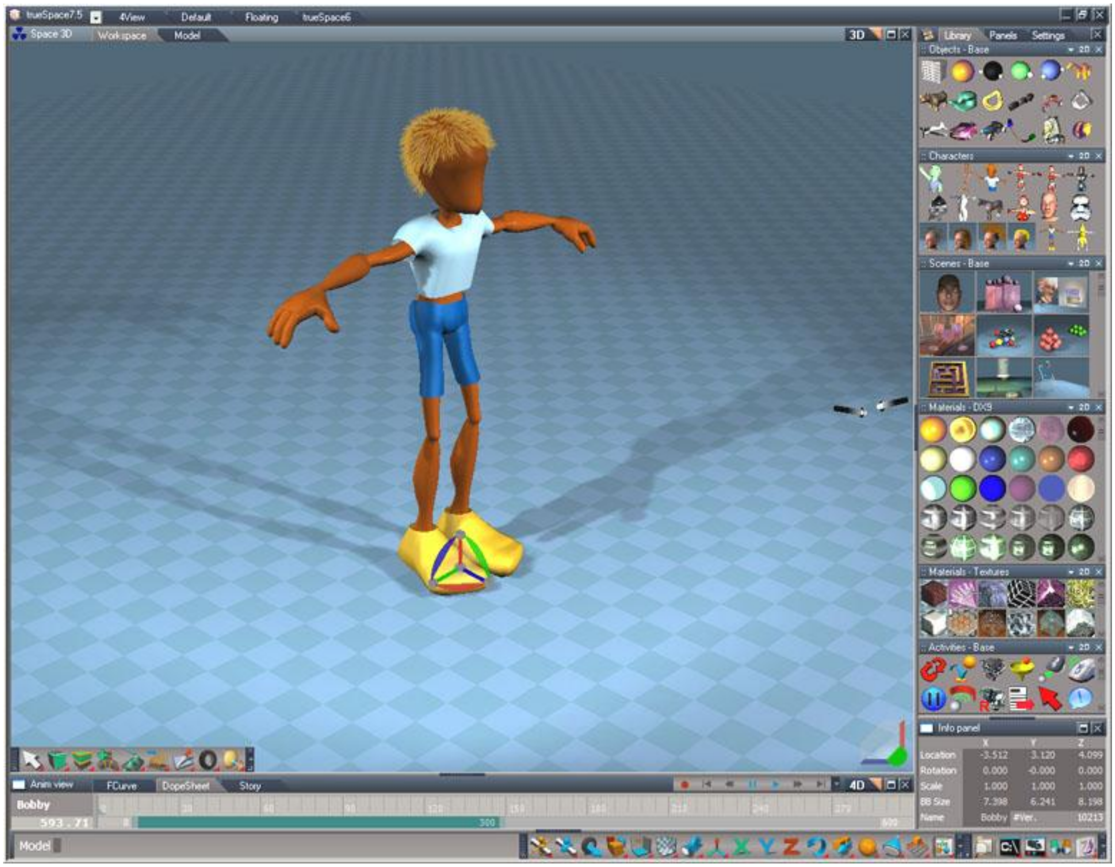
The relationship between contexts and participants

Everything in trueSpace is context dependent, and the meaning of any action may change as you move from context to context. For example, you can control a tricycle object in a street "play" context by rotating its pedals or rotating its handlebars to steer. By right-clicking on the tricycle, you can exit the "play" context and enter an "edit" context, where you can modify the tricycle by adding, for example, a bell to its handle bars.

You can create and edit individual spaces and save them as scenes, but sometimes it is useful to save a complete context, i.e. a space together with a layout containing your views and other tools.

2.1.5 Desktop

When you first open trueSpace you will see the desktop with a default layout. This layout contains several views and other panels with tools and libraries. It should look similar to the image below.



	X	Y	Z
Location	-3.512	3.120	4.099
Rotation	0.000	-0.000	0.000
Scale	1.000	1.000	1.000
BB Size	7.398	6.241	8.198
Name	Bobby	#Ver.	10213

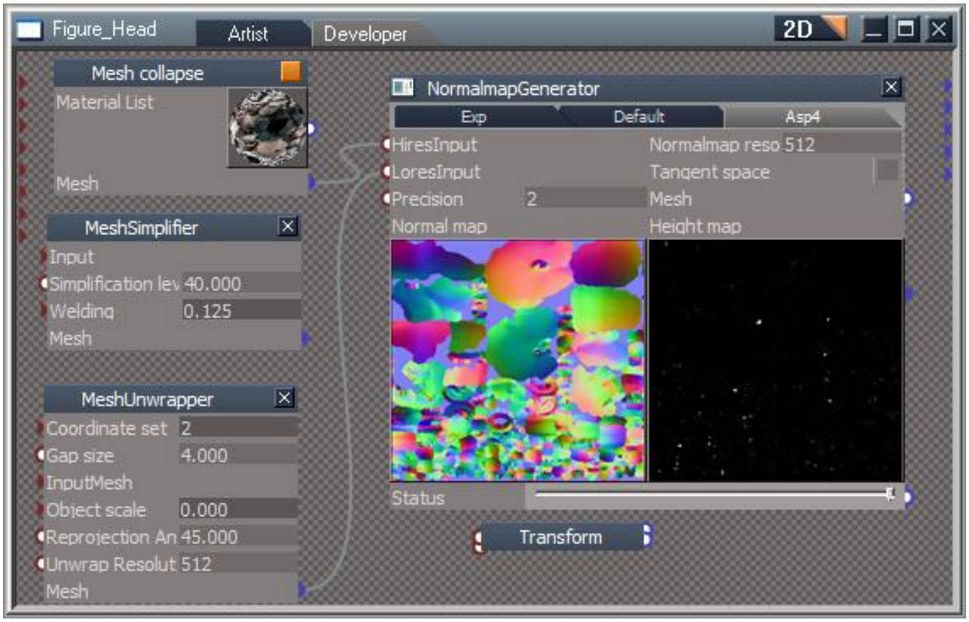
trueSpace Default layout in Desktop context.

The desktop context is the highest level context of the trueSpace interface. It is a collection of windows which together comprise your overall workplace. The trueSpace desktop is part of the Windows desktop.

Note: trueSpace's interface is optimized for high-resolution desktops, and a resolution of at least 1280x1024 is recommended.

2.2 Windows and Panels

Views are represented as Windows on the desktop. They are literally your windows into the space where you are working, which is really nothing other than an object that you have opened to edit. You can imagine that, by opening this view, you have entered the object and are inside of it. This is called your current context. You can navigate within the context of this 2D aspect by using the 2D navigation widget.



Mesh collapse
Material List
Mesh

MeshSimplifier
Input
Simplification lev 40.000
Welding 0.125
Mesh

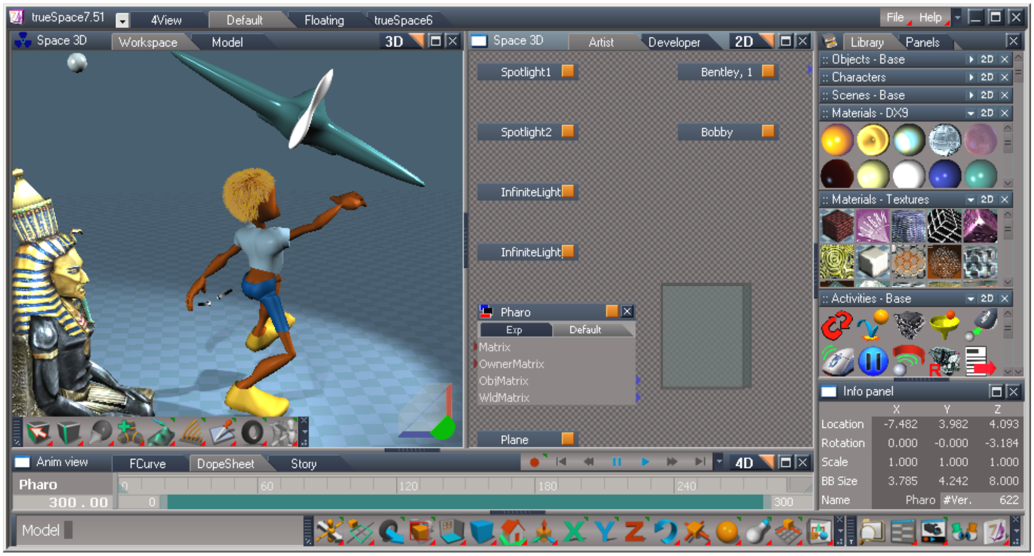
MeshUnwrapper
Coordinate set 2
Gap size 4.000
InputMesh
Object scale 0.000
Reprojection An 45.000
Unwrap Resolut 512
Mesh

NormalmapGenerator
HiresInput
LoresInput
Precision 2
Normal map
Normalmap reso 512
Tangent space
Mesh
Height map
Status

Link Editor View showing the Normal Map Generator

2.2.1 Active View Windows

The Active View Window is the window to which are all keyboard and mouse input are sent. The active view window is indicated by a lightened title bar. To activate a different view window simply click anywhere inside of it. Only one window may be active at a time.

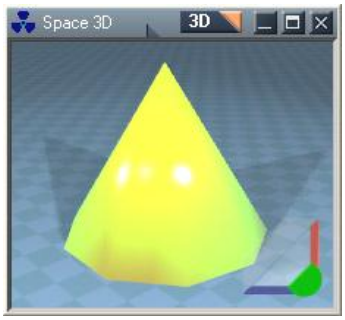


	X	Y	Z
Location	-7.482	3.982	4.093
Rotation	0.000	-0.000	-3.184
Scale	1.000	1.000	1.000
BB Size	3.785	4.242	8.000
Name	Pharo	#Ver.	622

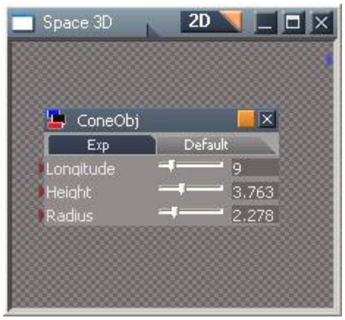
Link Editor is the active View in this layout

2.2.2 Object Aspects

Objects can be displayed in various windows in 3D, 2D or 1D aspects. Below you see the same object in different views and displaying different aspects. The main objective is to ensure you understand that with trueSpace, objects have evolved to multiple dimensions.

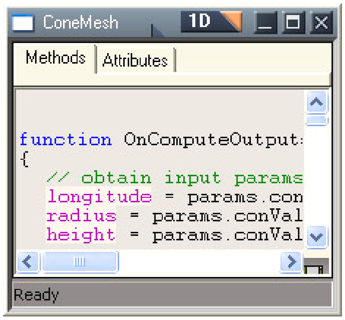


3D aspect (Workspace)



Longitude	9
Height	3.763
Radius	2.278

2D aspect (Link Editor)

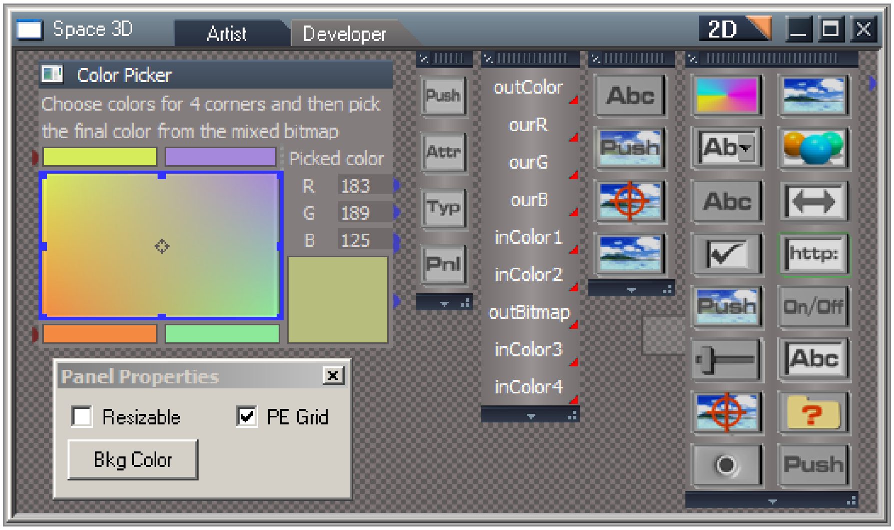


```
function OnComputeOutput
{
  // obtain input params
  longitude = params.con
  radius = params.conVal
  height = params.conVal
```

1D aspect (Script Editor)

Even inside of a single view, objects can still display multiple aspects. This is most useful in 2D, where aspects of an object are called **Panels.**

When selected in the Link Editor, Panels display tabs of all available aspects. You can edit these aspects and add new ones by right-clicking on panel’s title bar.



Choose colors for 4 corners and then pick the final color from the mixed bitmap

Picked color
R 183
G 189
B 125

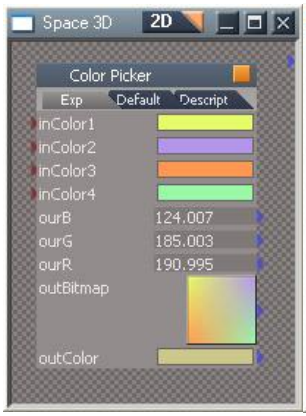
Panel Properties

☐ Resizable ☑ PE Grid

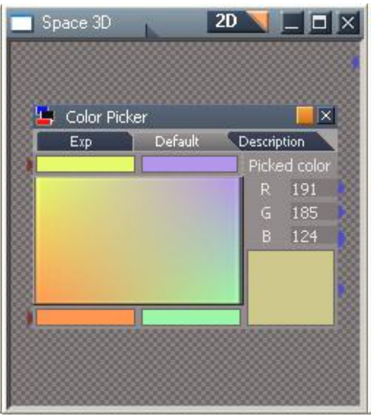
Bkg Color

Color Picker's Description aspect in Edit mode

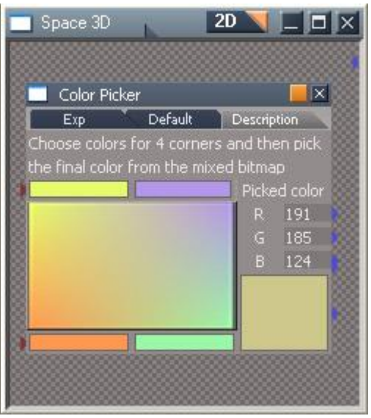
The expanded aspect below left was created automatically, but the Default and Description aspects were edited/created by a user.



inColor1
inColor2
inColor3
inColor4
ourB 124.007
ourG 185.003
ourR 190.995
outBitmap
outColor



Picked color
R 191
G 185
B 124



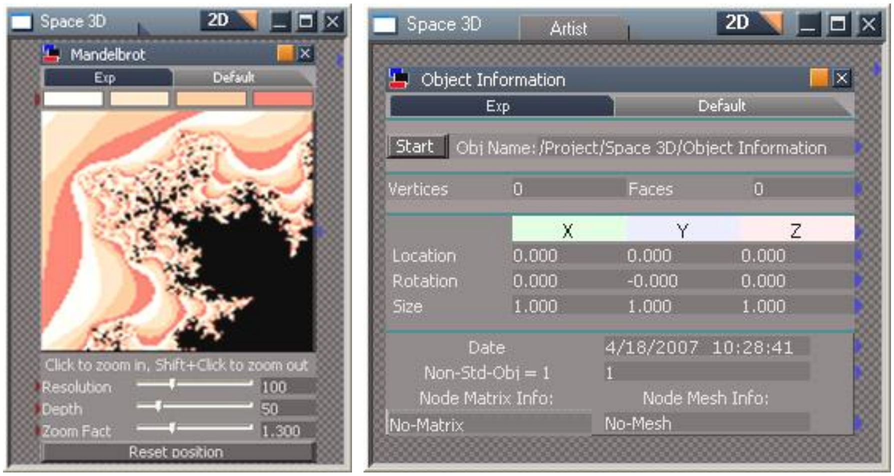
Choose colors for 4 corners and then pick the final color from the mixed bitmap

Picked color
R 191
G 185
B 124

Color Picker Exp View **Color Picker Default View** **Color Picker Description View**

Panels

A Panel is a 2D representation of an object. The object can be shown in 3D view as well. Panels are mostly used for storing and displaying trueSpace UI controls. Panels below were both user created.



Click to zoom in, Shift+Click to zoom out

Resolution 100

Depth 50

Zoom Fact 1.300

Obj Name: /Project/Space 3D/Object Information

Vertices 0 Faces 0

	X	Y	Z
Location	0.000	0.000	0.000
Rotation	0.000	-0.000	0.000
Size	1.000	1.000	1.000

Date 4/18/2007 10:28:41

Non-Std-Obj = 1 1

Node Matrix Info: No-Matrix

Node Mesh Info: No-Mesh

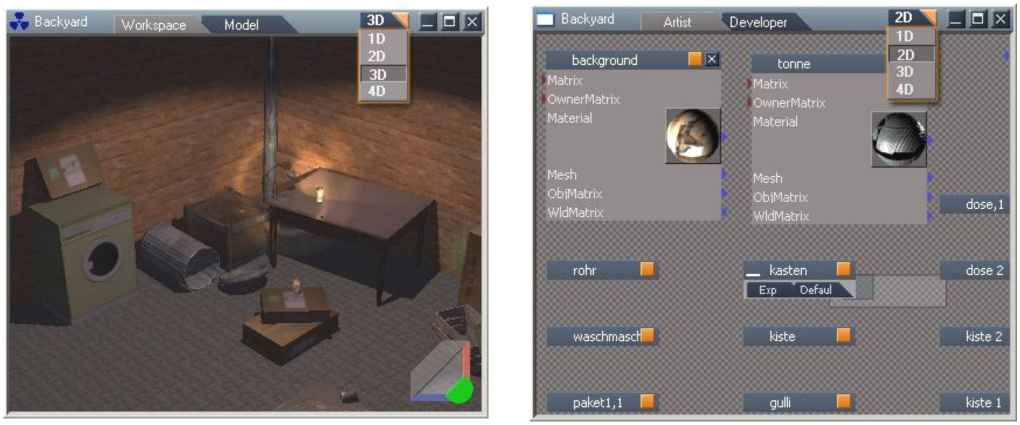
User created aspects for object panels in Link Editor

2.2.3 Windows Aspects

Every trueSpace object can display itself in one or more **Aspects** (the visible user interface) and windows are no different. The aspect of most trueSpace windows can be changed using the title bar popup menu. This popup menu is different for different types of windows.

View Aspects

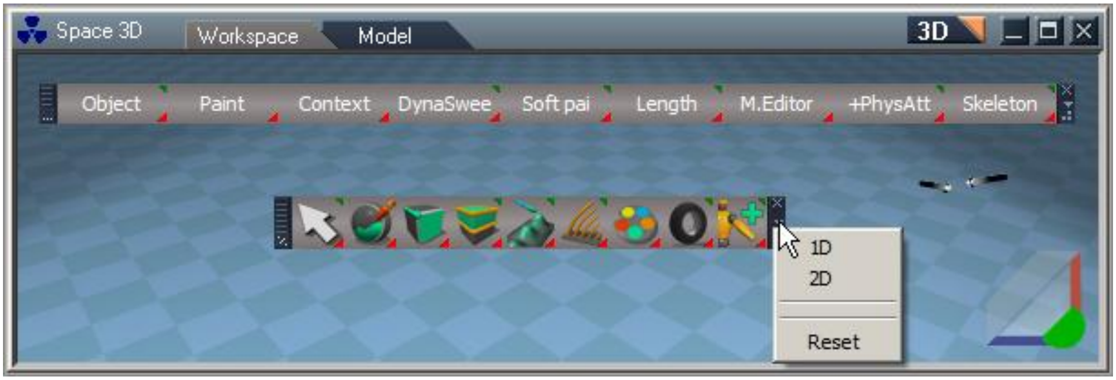
The most important aspect category of a view window is its **dimensional** aspect. Dimensional aspects are shown on a title bar in form of a popup. Below you can see the result of switching the dimensional aspect from 3D to 2D. Both views show same content but in different dimensions.



You can change the Workspace View to the Link Editor View simply by changing its dimensional aspect

Toolbar Aspects

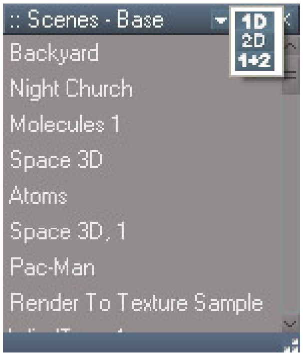
For toolbars, the aspect popup menu is available by clicking the black triangle on the bottom right-hand corner of the toolbar. Here you can select from 1D (text) or 2D (icons).



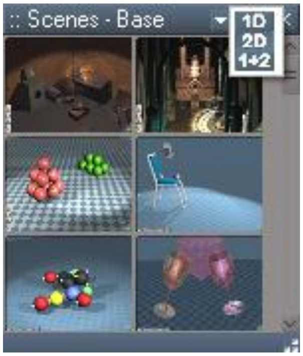
You can turn Icon toolbar into a text menu with one click

Library Aspects

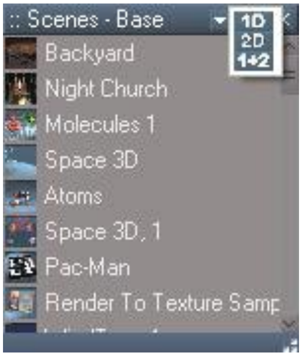
Libraries have three aspects to choose from. You can choose a dimensional aspect of either 1D (text), 2D (icons), or 1+2 (both text and icons) by clicking on the popup menu in the upper right-hand corner of the Library title bar.



1D Text View



2D Icon View

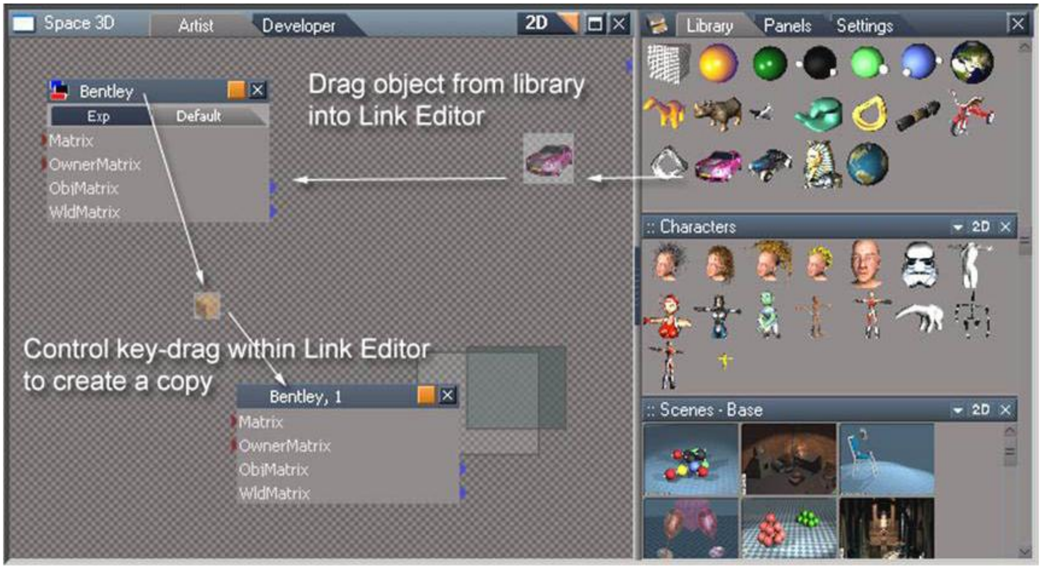


1+2 Text plus Icons

2.2.4 Drag and Drop Manipulation

Starting in the Library stack, you can drag and drop an item to another library, creating a copy in the other library. If you hold the Shift-key and drag an item, you actually move the item from the original library to the other library.

In the Link Editor, pressing the CTRL key while dragging an object creates a copy of that object.

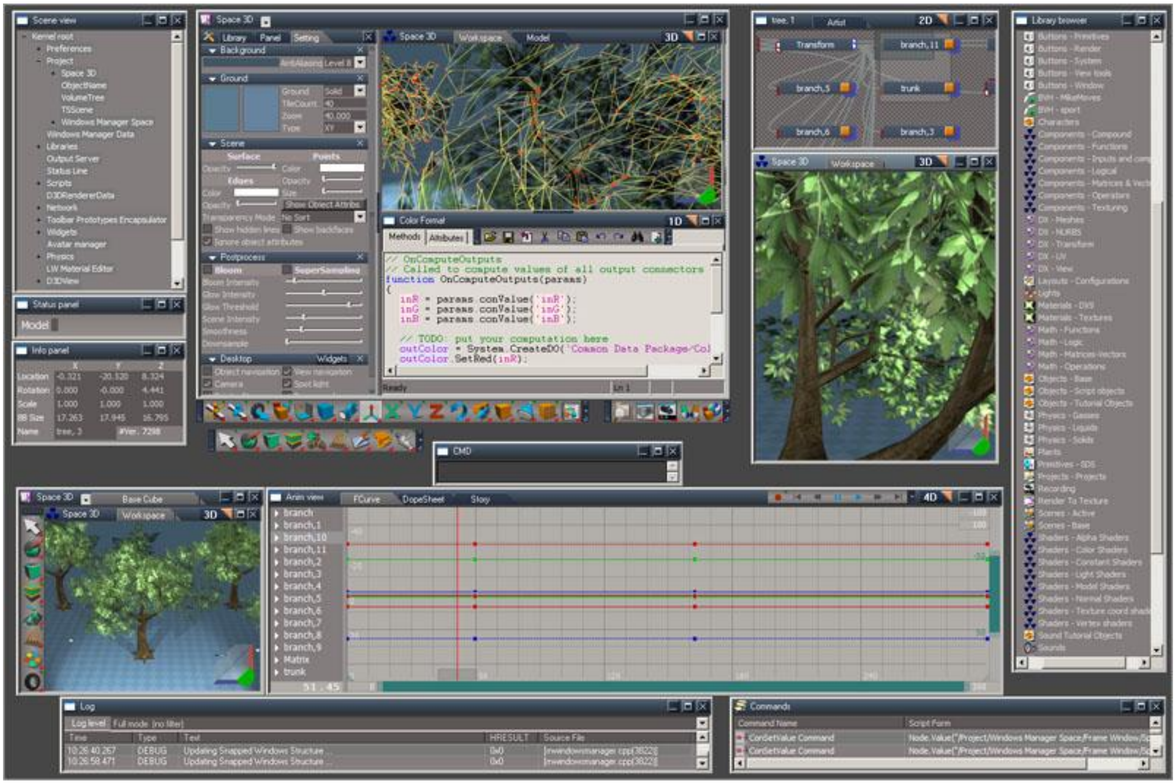

Drag object from library into Link Editor

Control key-drag within Link Editor to create a copy

Dragging object between and inside windows

Window Snapping

Toolbars, panels, and windows can be snapped together on the desktop simply by dragging them. Windows can be snapped edge-to-edge or corner-to-corner. Each window could have its own set of toolbars.



	X	Y	Z
Location	-0.321	-20.520	8.324
Rotation	0.000	-0.000	4.441
Scale	1.000	1.000	1.000
BB Size	17.263	17.945	16.795
Name	tree, 1	#Ver.: 7298	

```
// OnComputeOutputs
// Called to compute values of all output connectors
function OnComputeOutputs(params)
{
  inR = params.conValue('inR');
  inG = params.conValue('inG');
  inB = params.conValue('inB');

  // TODO: put your computation here
  outColor = System.CreateDO('Common Data Package/Col
  outColor.SetRed(inR);
```

```
Log level  Full mode (no filter)
Time          Type   Text                                  HRESULT  Source File
10:26:40.267  DEBUG  Updating Snapped Windows Structure ...  0x0      [rwindowsmanager.cpp(3822)]
10:26:58.471  DEBUG  Updating Snapped Windows Structure ...  0x0      [rwindowsmanager.cpp(3822)]
```

Command Name	Script Form
ConSetValue Command	Node.Value("/Project/Windows Manager Space/Frame Window/S
ConSetValue Command	Node.Value("/Project/Windows Manager Space/Frame Window/S

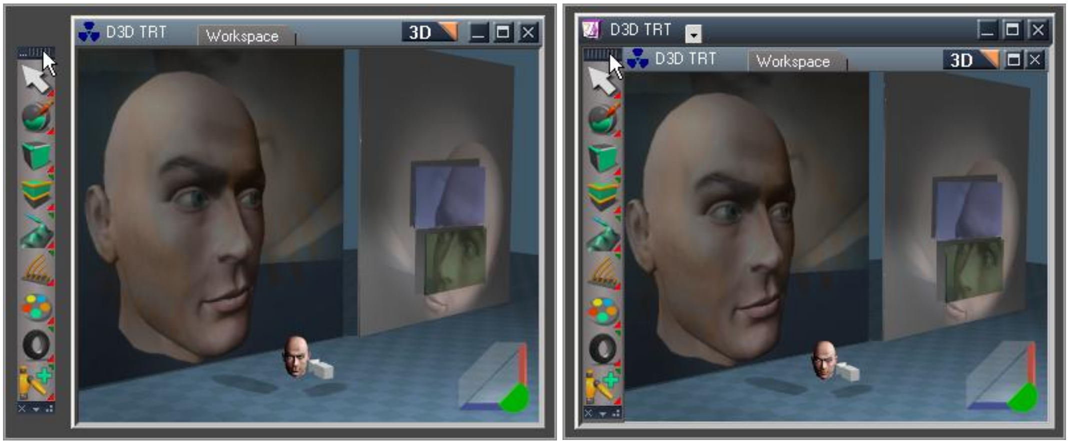
Complex Floating layout with some windows snapped

2.2.5 Windows Docking

You can dock any trueSpace window by CTRL+dragging into another open window. A new outer window will be created to hold two or more inner windows.

Docking Position

You can dock a window to the top, bottom, left, or right side of an open window. trueSpace shows you a preview of the docking position as transparent outline while you CTRL+drag the window to be docked.

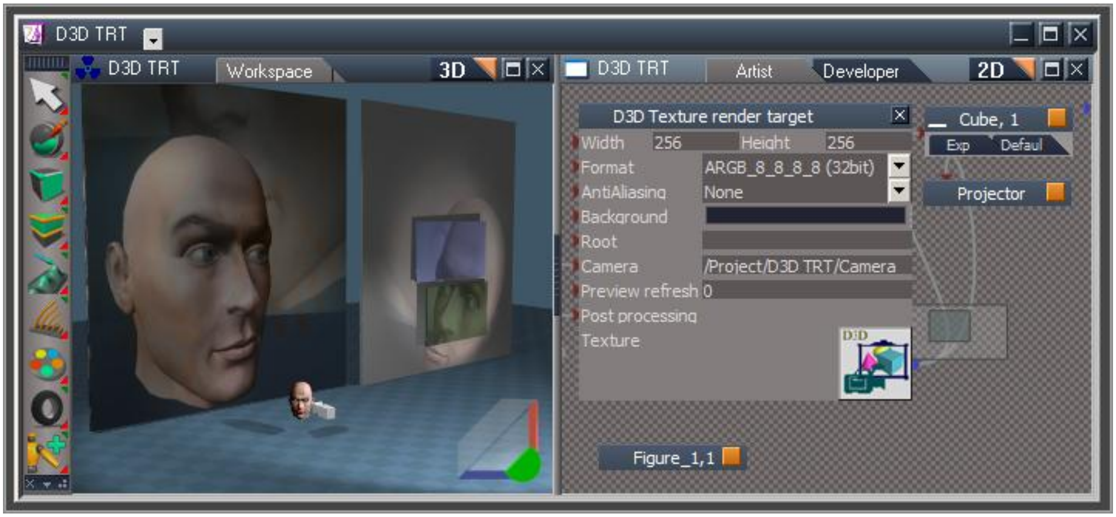


Control-key plus drag of toolbar on left, showing the toolbar now docked on the right

An outer docking position will be used if there is already a docked hierarchy present and the mouse is closer to the window border. This docking position will be shown interactively.

Docking More Windows

Here is an example of Workspace and Link Editor view windows docked inside trueSpace main window:



D3D Texture render target

Width 256 Height 256

Format ARGB_8_8_8_8 (32bit)

AntiAliasing None

Background

Root

Camera /Project/D3D TRT/Camera

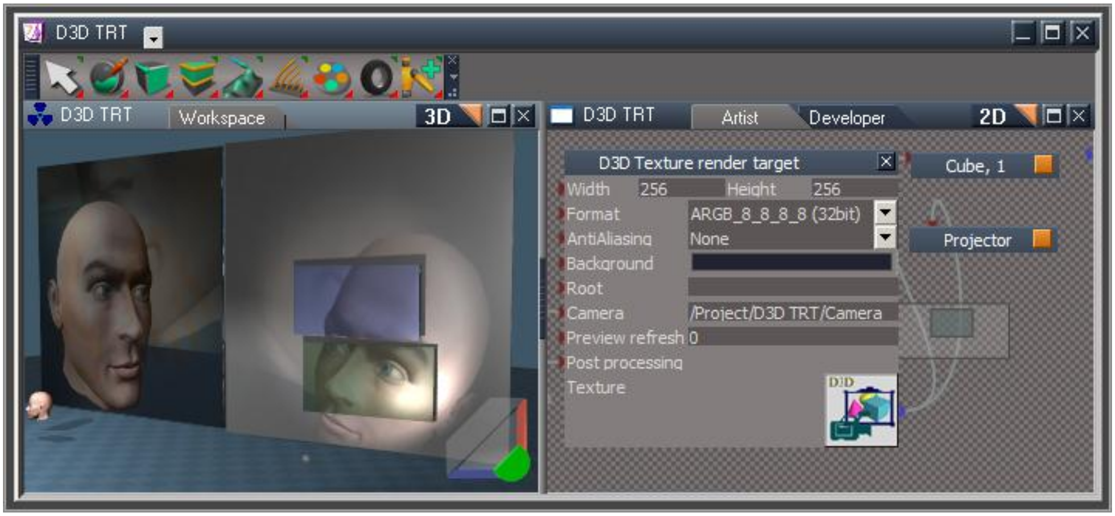
Preview refresh 0

Post processing

Texture

Docking windows in trueSpace

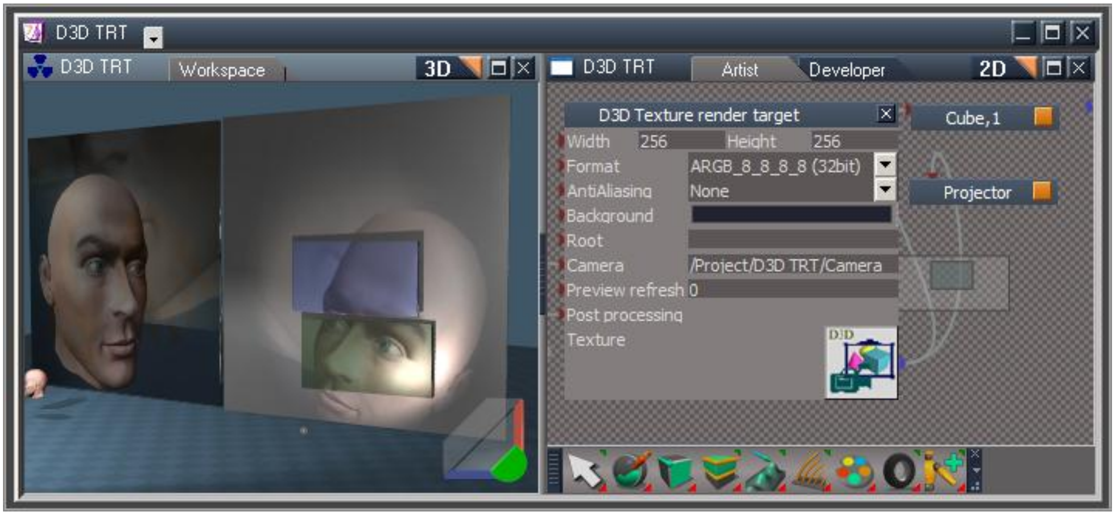
While CTRL+dragging a third window, you can dock it either on the same level of the hierarchy:



Width 256 Height 256
Format ARGB_8_8_8_8 (32bit)
AntiAliasing None
Background
Root
Camera /Project/D3D TRT/Camera
Preview refresh 0
Post processing
Texture

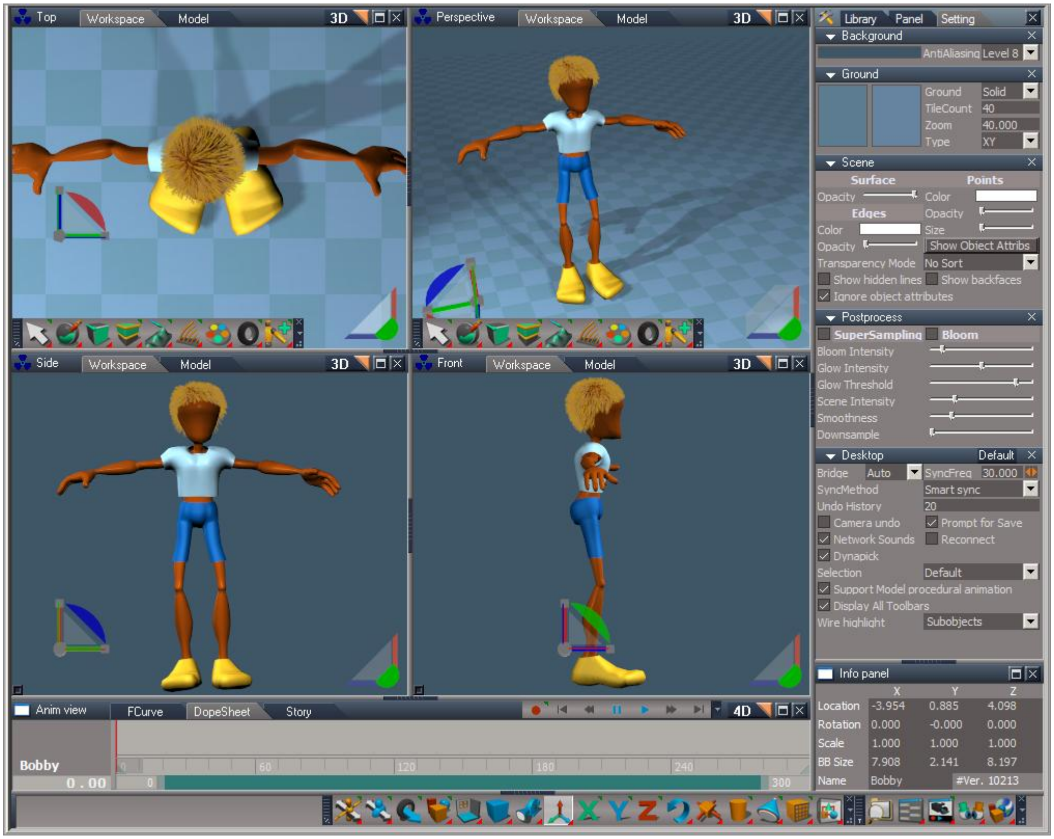
Three windows docked at the same hierarchy level

Or you can drag it to a docked window to create a two-level hierarchy:



Width 256 Height 256
Format ARGB_8_8_8_8 (32bit)
AntiAliasing None
Background
Root
Camera /Project/D3D TRT/Camera
Preview refresh 0
Post processing
Texture

Three windows docked hierarchically on two levels



Background
AntiAliasing Level 8

Ground
Ground Solid
TileCount 40
Zoom 40.000
Type XY

Scene
Surface
Opacity
Edges
Color
Opacity
Points
Color
Opacity
Size
Show Object Attribs
Transparency Mode No Sort
□ Show hidden lines □ Show backfaces
☑ Ignore object attributes

Postprocess
□ SuperSampling □ Bloom
Bloom Intensity
Glow Intensity
Glow Threshold
Scene Intensity
Smoothness
Downsample

Desktop Default
Bridge Auto SyncFreq 30.000
SyncMethod Smart sync
Undo History 20
□ Camera undo ☑ Prompt for Save
☑ Network Sounds □ Reconnect
☑ Dynapick
Selection Default
☑ Support Model procedural animation
☑ Display All Toolbars
Wire highlight Subobjects

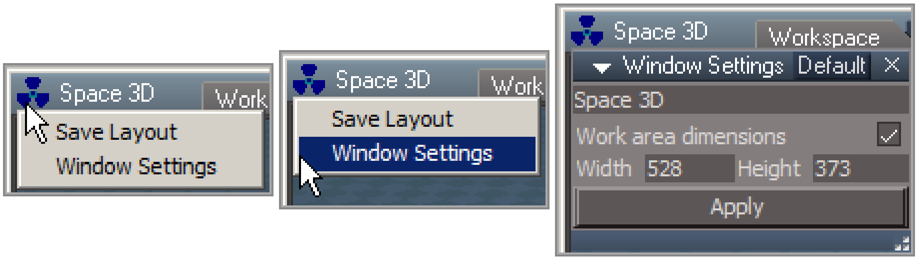
Info panel

	X	Y	Z
Location	-3.954	0.885	4.098
Rotation	0.000	-0.000	0.000
Scale	1.000	1.000	1.000
BB Size	7.908	2.141	8.197
Name	Bobby	#Ver. 10213	

More complex docking of preset 4-View layout

Window Settings

This can be used to set your windows to an appropriate size by entering values and clicking on Apply it is a common dialog available to all windows. The menu to show the settings dialog is shown by a left click on the top left part of any windows title bar.



Space 3D
Work area dimensions ☑
Width 528 Height 373

Windows Settings

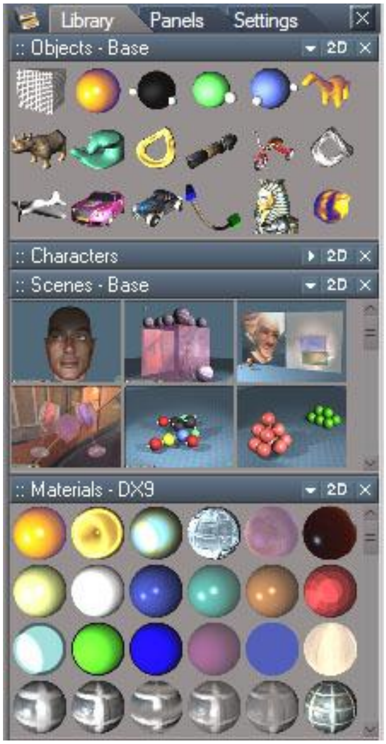
- Work area dimensions :
  If Checked the this places a border around the work area by adding pixels to the outer frame.
  For example: When checked if the width is set to 100 the working area inside the frame is set to 100 but the window size will have the added pixels added to its overall dimensions.
  When its unchecked if the width set to 100 then the whole window including the frame is set to 100.

A good example of use could be if you want to ensure your window matches the size settings for a Render To File preset, or if you want to use the size of your window in the Render To File dialog as a custom setting , then you can open the Window Settings dialog in the top left of the Workspace window and it will show you the Width and Height details you would need enter into the Render To File dialog.

- ➢ Further reference: Chapter 7 Light&Render 7.1.7 Real-time Render To File

Docking Panels

Panels are docked the same way as any window, but the result is a little different. When a panel is docked, a special area is created where only other panels and libraries can be vertically docked. If there is not enough space for all the docked panels, then a vertical scrollbar will appear, allowing you to scroll through these tall spaces.



Docked library panels in Stack View.

Undocking a Window

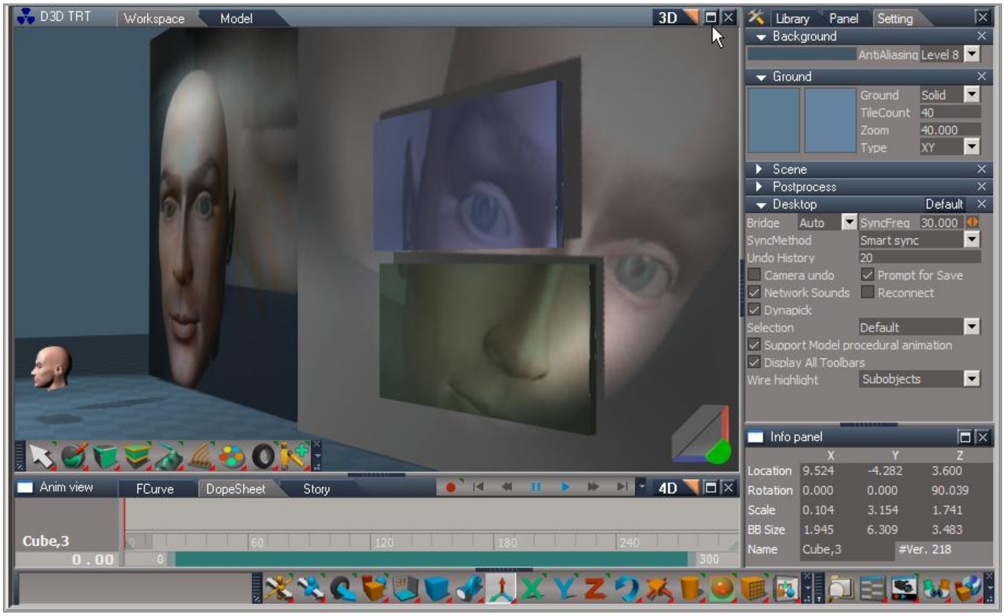
To un-dock a window from a docked hierarchy, CTRL+drag the window from its docked position, to a new location of your choice.

Docked Window Aspect

The aspect of each docked window is changed in the standard way. Changing the aspect of a docked window will not affect other windows.

2.2.6 Maximizing and Restoring Windows

When you want to temporarily maximize a docked window, you can accomplish this by clicking the **Maximize** icon on the title bar of the view you want to maximize. In this example, the Workspace window was maximized by using the Maximize option .



▾ Background

AntiAliasing Level 8

▾ Ground

Ground Solid
TileCount 40
Zoom 40.000
Type XY

▸ Scene

▸ Postprocess

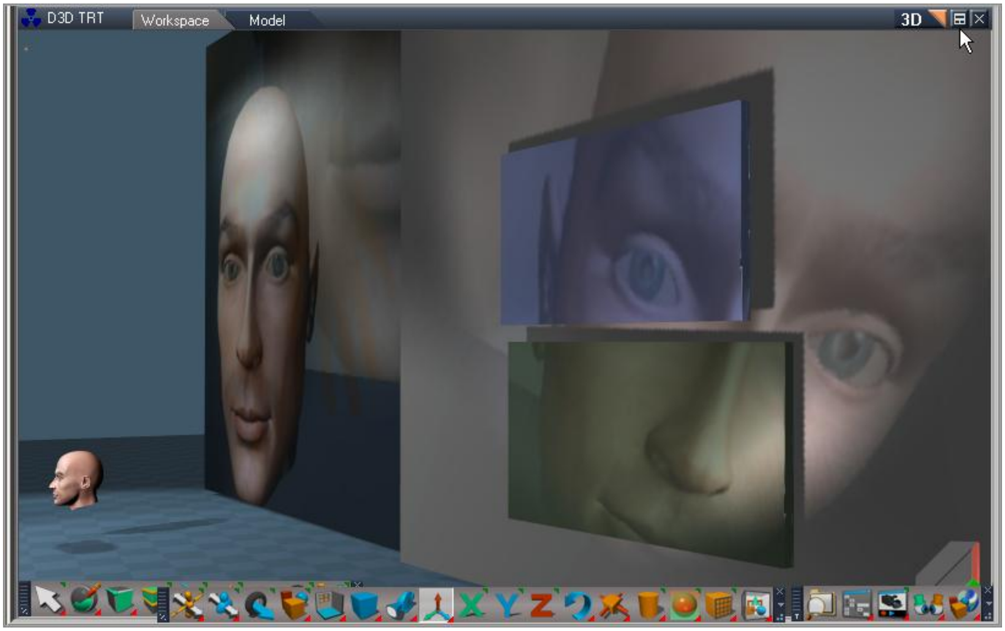
▾ Desktop Default

Bridge Auto SyncFreq 30.000
SyncMethod Smart sync
Undo History 20
□ Camera undo ☑ Prompt for Save
☑ Network Sounds □ Reconnect
☑ Dynapick
Selection Default
☑ Support Model procedural animation
☑ Display All Toolbars
Wire highlight Subobjects

	X	Y	Z
Location	9.524	-4.282	3.600
Rotation	0.000	0.000	90.039
Scale	0.104	3.154	1.741
BB Size	1.945	6.309	3.483
Name	Cube,3	#Ver. 218	

Cube,3
0.00
0 60 120 180 240 300

Default layout before Workspace view is maximized

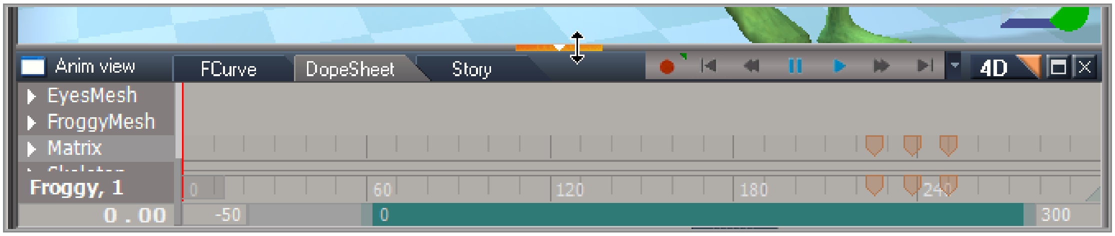


Workspace window in maximized state

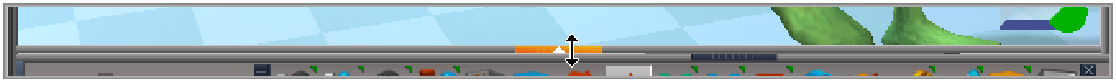
Note : the double border informs you this window is in its maximized state. To return to the original docking configuration click the **Restore** icon on the title bar.

Windows Folding

Every window has a dividing handle which can be dragged to resize the window the center handle highlights orange when mouse over, left clicking on the handle will fold and minimize or maximize a window according to the direction the triangle in it is pointing.
The direction of the folding can be reversed by a R-click over the white triangle in the handle



Anim view handle is left clicked

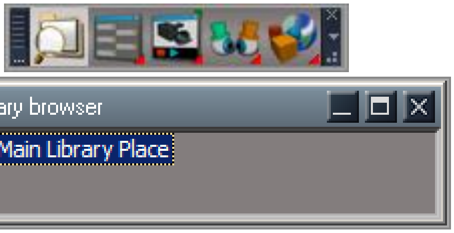


Anim view in folded state

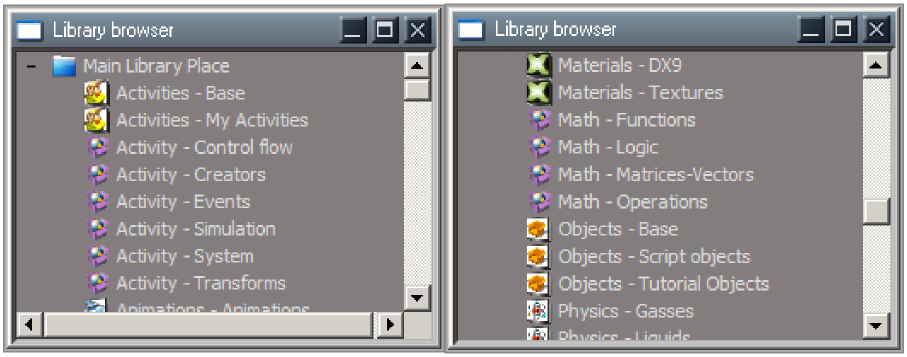
2.3 Library Browser – Libraries

2.3.1 Library Browser

Every library in trueSpace exists in the Library Browser. From the library browser, you can create and organize your libraries. By default, trueSpace comes with a wide variety of pre-existing libraries full of library items such as objects, materials, scripts, buttons and an assortment of other library types.

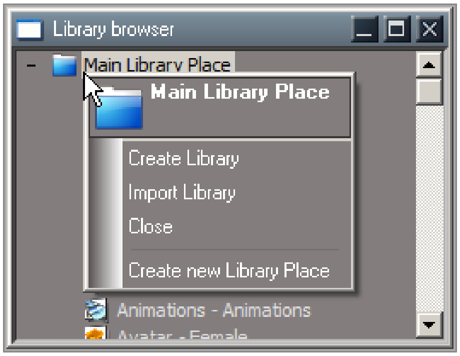


Library Browser's Main Library Place is minimized



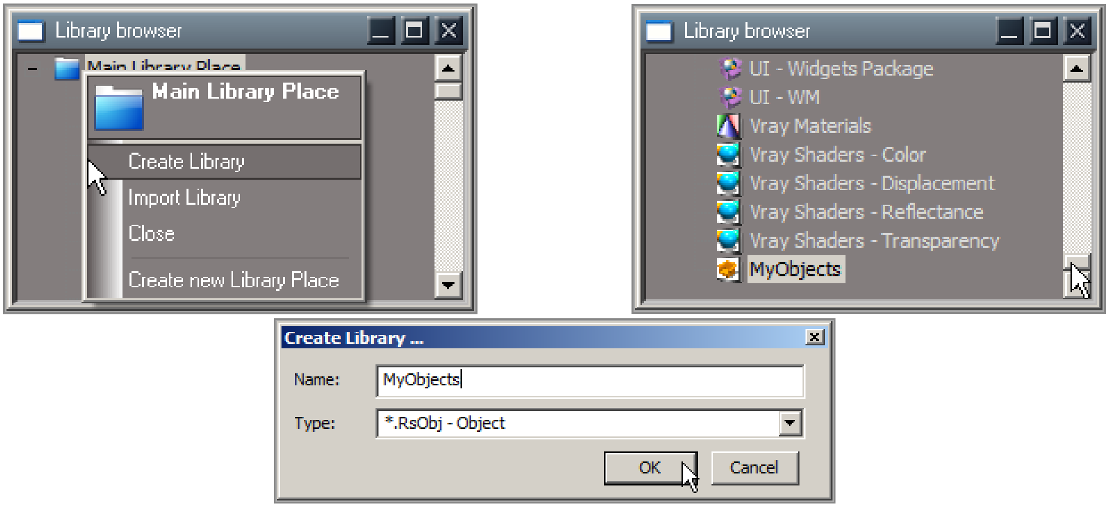
Library Browser expanded to show Main Library Place libraries

By default, the Main Library Place is expanded in the Library Browser. It contains all the libraries that come with trueSpace. It may also contain any libraries you wish to create or import. The menu shown below (right-click on Main Library Place) shows options available at this level of the Library Browser.



Main Library Place: right-click menu

Create Library: To create a custom library, choose the Create Library menu item. In the Create Library dialog enter a name for the new library and specify it's type. In this case, a new object library is created. Click "OK" on the dialog when you have named and set the type for the new library. It will show itself at the bottom of the Library Browser.

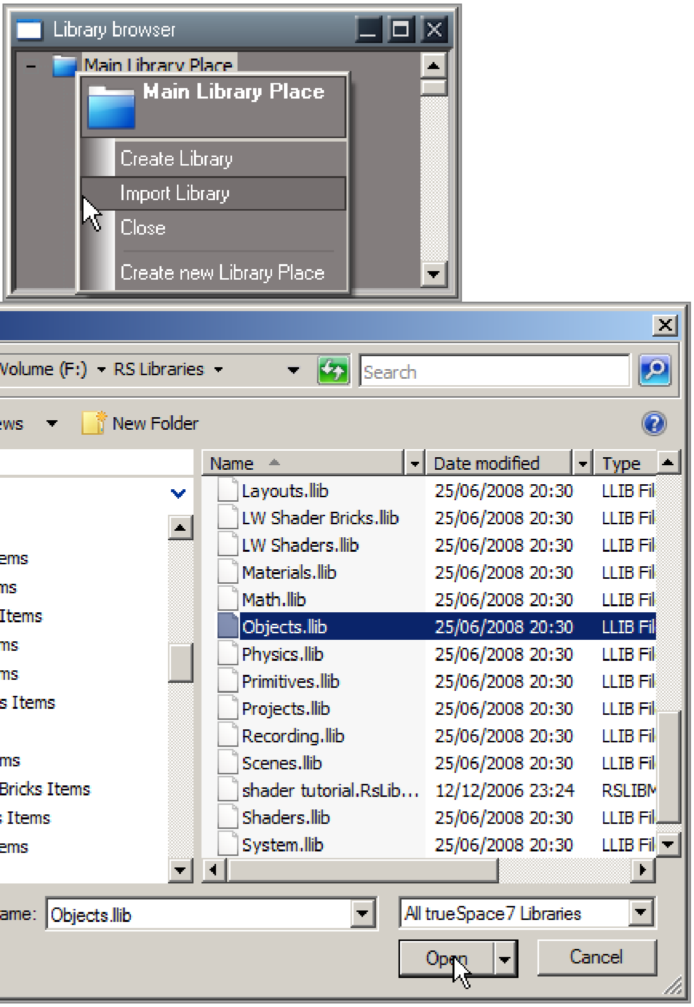


Name: MyObjects

Type: *.RsObj - Object

Create Library menu item, new library appended to bottom of Library Browser

Import Library: when you wish to import libraries from other sources, that exist on your hard drive, the Import Library menu item is used. This is actually a very powerful menu item as the following illustrations demonstrate.



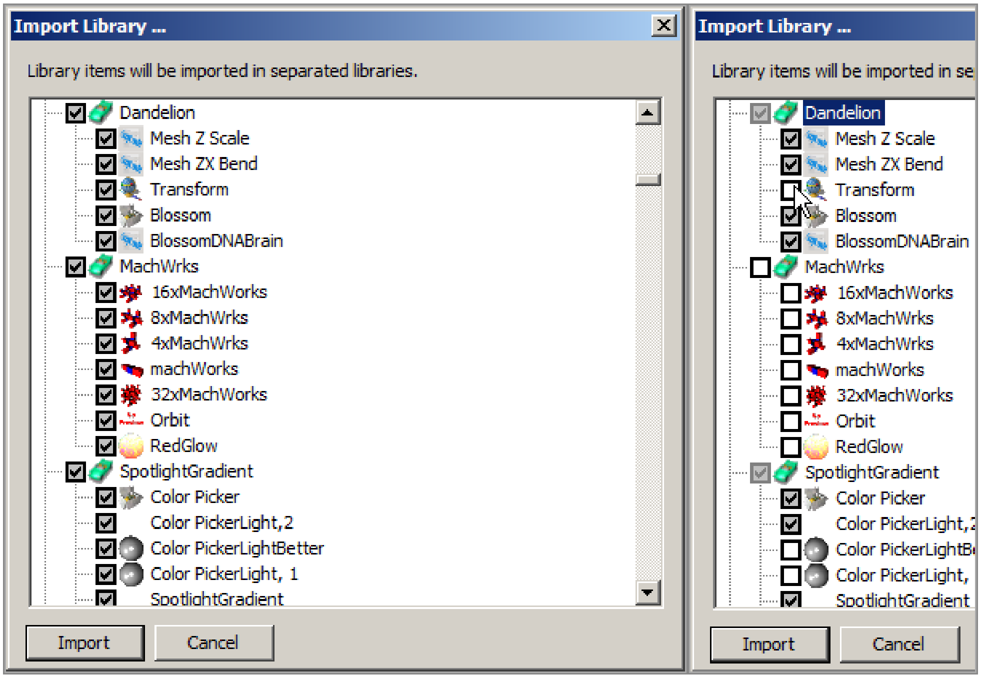
Name	Date modified	Type
Layouts.llib	25/06/2008 20:30	LLIB Fil
LW Shader Bricks.llib	25/06/2008 20:30	LLIB Fil
LW Shaders.llib	25/06/2008 20:30	LLIB Fil
Materials.llib	25/06/2008 20:30	LLIB Fil
Math.llib	25/06/2008 20:30	LLIB Fil
Objects.llib	25/06/2008 20:30	LLIB Fil
Physics.llib	25/06/2008 20:30	LLIB Fil
Primitives.llib	25/06/2008 20:30	LLIB Fil
Projects.llib	25/06/2008 20:30	LLIB Fil
Recording.llib	25/06/2008 20:30	LLIB Fil
Scenes.llib	25/06/2008 20:30	LLIB Fil
shader tutorial.RsLib...	12/12/2006 23:24	RSLIBM
Shaders.llib	25/06/2008 20:30	LLIB Fil
System.llib	25/06/2008 20:30	LLIB Fil

File name: Objects.llib

All trueSpace7 Libraries

Import Library menu item brings up the Import Library dialog

When the Import Library dialog shows up, you can navigate to your hard drive and any libraries which may be available for importing. Here in the illustration above, my trueSpace7 libraries are showing as available for import. I have some custom objects I wish to import. I simply select a library I wish to open and click the Open button on the dialog. This will cause trueSpace to load the entire library into the Library Browser’s selection dialog below.



Import Library ...

Library items will be imported in separated libraries.

- ☑ Dandelion
  - ☑ Mesh Z Scale
  - ☑ Mesh ZX Bend
  - ☑ Transform
  - ☑ Blossom
  - ☑ BlossomDNABrain
- ☑ MachWrks
  - ☑ 16xMachWorks
  - ☑ 8xMachWrks
  - ☑ 4xMachWrks
  - ☑ machWorks
  - ☑ 32xMachWorks
  - ☑ Orbit
  - ☑ RedGlow
- ☑ SpotlightGradient
  - ☑ Color Picker
  - ☑ Color PickerLight,2
  - ☑ Color PickerLightBetter
  - ☑ Color PickerLight, 1
  - ☑ SpotlightGradient

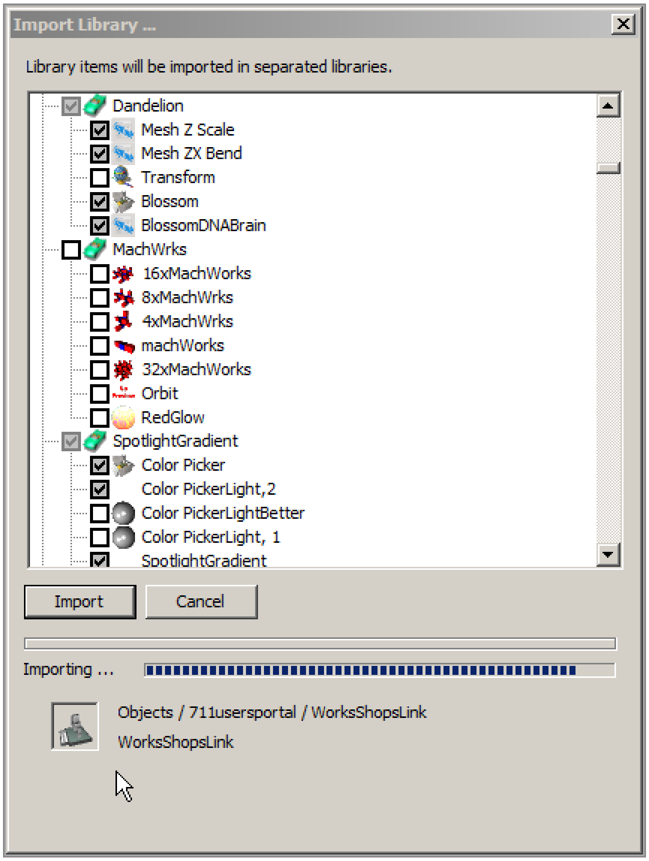
Import Library ...

Library items will be imported in se

- ☑ Dandelion
  - ☑ Mesh Z Scale
  - ☑ Mesh ZX Bend
  - ☐ Transform
  - ☑ Blossom
  - ☑ BlossomDNABrain
- ☐ MachWrks
  - ☐ 16xMachWorks
  - ☐ 8xMachWrks
  - ☐ 4xMachWrks
  - ☐ machWorks
  - ☐ 32xMachWorks
  - ☐ Orbit
  - ☐ RedGlow
- ☑ SpotlightGradient
  - ☑ Color Picker
  - ☑ Color PickerLight,2
  - ☐ Color PickerLightBe
  - ☐ Color PickerLight,
  - ☑ SpotlightGradient

Select all or some of the items contained in the library

By default all items in the library are checked for import. If you wish to only select a few items, uncheck the top-level checkbox. This will deselect all the items. You may then select only the items you wish to be imported. Special note that in the illustration above, The main Objects library contains sub-libraries, which in turn contain the actual items. When you select an item from any given sub-library, the sub-library is actually imported with the items you want included in the import process. Once you have made your decisions on what to import, you click the Import button to begin the process.



Import Library ...

Library items will be imported in separated libraries.

- ☑ Dandelion
  - ☑ Mesh Z Scale
  - ☑ Mesh ZX Bend
  - □ Transform
  - ☑ Blossom
  - ☑ BlossomDNABrain
- □ MachWrks
  - □ 16xMachWorks
  - □ 8xMachWrks
  - □ 4xMachWrks
  - □ machWorks
  - □ 32xMachWorks
  - □ Orbit
  - □ RedGlow
- ☑ SpotlightGradient
  - ☑ Color Picker
  - ☑ Color PickerLight,2
  - □ Color PickerLightBetter
  - □ Color PickerLight, 1
  - ☑ SpotlightGradient

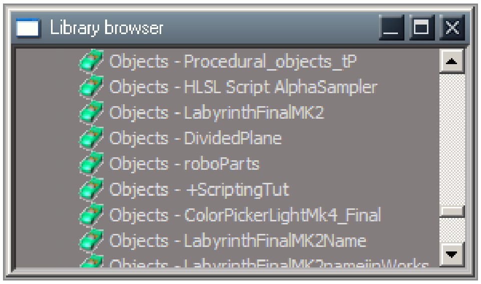
Importing ...

Objects / 711usersportal / WorksShopsLink

WorksShopsLink

Libraries and items being imported into the Library Browser. Progress of the operation is displayed.

Once the process completes importing, the new libraries will append to the bottom of the Library Browser.

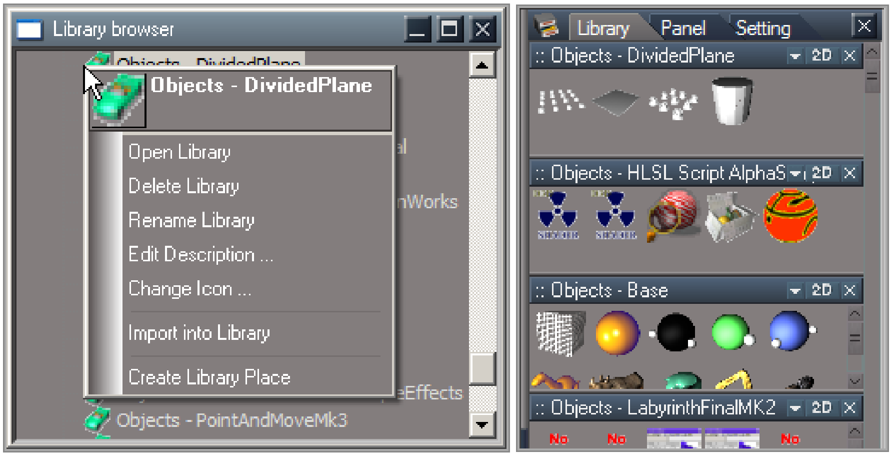


Library browser

- Objects - Procedural_objects_tP
- Objects - HLSL Script AlphaSampler
- Objects - LabyrinthFinalMK2
- Objects - DividedPlane
- Objects - roboParts
- Objects - +ScriptingTut
- Objects - ColorPickerLightMk4_Final
- Objects - LabyrinthFinalMK2Name
- Objects - LabyrinthFinalMK2nameijnWorks

Imported libraries show up at bottom of Library Browser

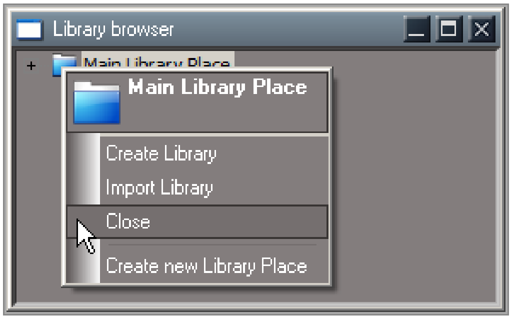
A double-click on the library in the Library Browser list will load that particular library to the top of the Library stack.



Loading a library from the Library Browser to Library stack area

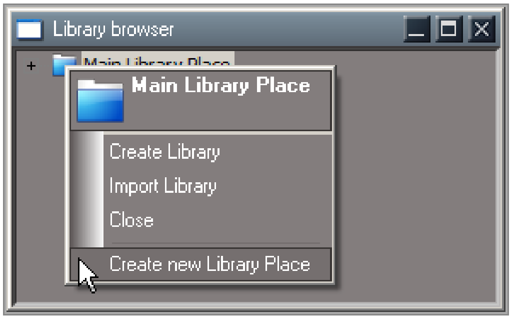
In addition to double-click method, a right-click on the library name in the Library Browser, displays a set of menu items. The Open Library menu item will also load the library in the Library stack.

Close: this menu item will close the current selected main library place.



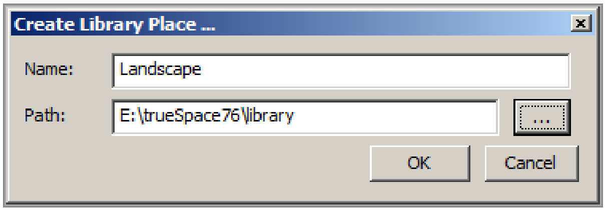
Close menu item used to close a Main Library Place in the Library Browser

Create New Library Place: this tool further enhances the library browser by providing you the ability to create top-level Main Library Places. As your ability and use of trueSpace grows, chances are so will your libraries. To help organize the Library Browser, you create top-level Places to store your libraries. You may have top-level Places such as Landscape, which could contain sub-libraries for Plants, grasses, trees, shrubbery and so forth. Perhaps yet another Main Library Place called Textures, with sub-libraries for Red, Green, Blue, Black, White and so forth, where you could organize some textures by color if desired. It would not be long before you are able to organize the Library Browser so it works well for you.



Create new Library Place menu item creates top-level library places for you

Once you select the Create new Library Place menu item, you are presented with the Create Library Place dialog.



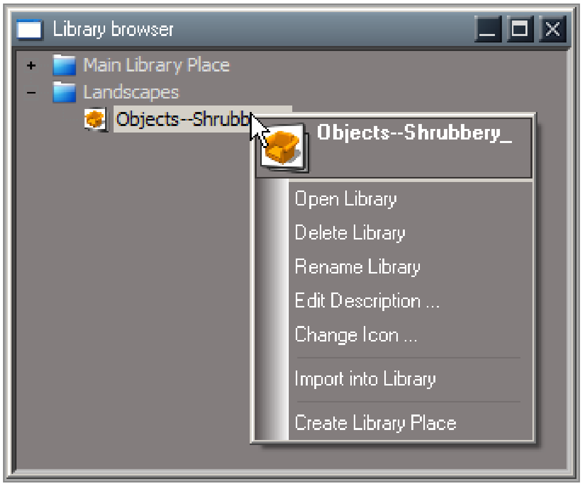
Name: Landscape

Path: E:\trueSpace76\library

Create Library Place dialog allows you to name and locate where the new Library Place will be on your hard drive.

- **Name:** enter the name of your new library place here.
- **Path:** use the browse button to right hand side and browse to a location on your hard drive. trueSpace will use this location to store sub-libraries to. If you wish to further refine your organization, create a New Folder in the path and point to it instead.

Now that we have covered Library Places, the next step is to cover libraries themselves. With an understanding of how to create and organize Main Library Places/Project Library Places, we can add specific libraries and items as desired. Yes we have covered importing libraries and items already into Main Library Places, so now we will explore the right-click menu items for a library.



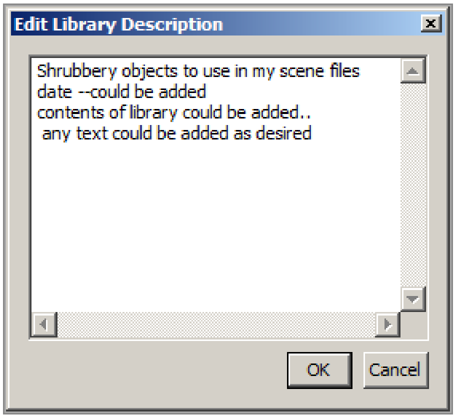
Right-click a library to reveal menu items

Open Library: opens the selected library to top of the Library Stack.

Delete Library: deletes the selected library from the library place. Use with caution as this also deletes the entire folder and all its contents from the hard drive and sends it to the recycle bin.

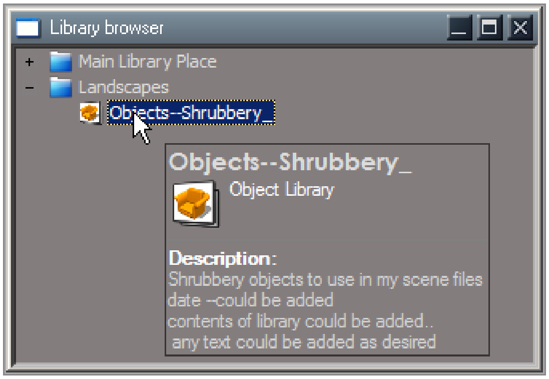
Rename Library: rename the selected library.

Edit Description: allows you to enter or edit information as you desire. This description you create appends to the library's tooltip during mouse-over the library.



Shrubbery objects to use in my scene files
date --could be added
contents of library could be added..
any text could be added as desired

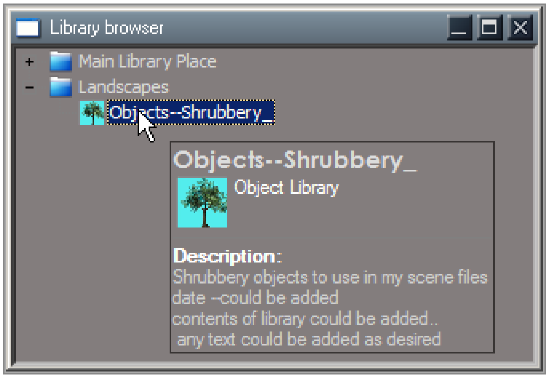
Edit Library Description as desired



Description:
Shrubbery objects to use in my scene files
date --could be added
contents of library could be added..
any text could be added as desired

Description appends to the library's tooltip during mouse-over

Change Icon: allows you to select an image on your hard drive to use as the icon for the object.



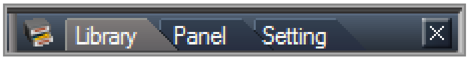
Description:
Shrubbery objects to use in my scene files
date --could be added
contents of library could be added..
any text could be added as desired

Using Change Icon to add custom icon for library

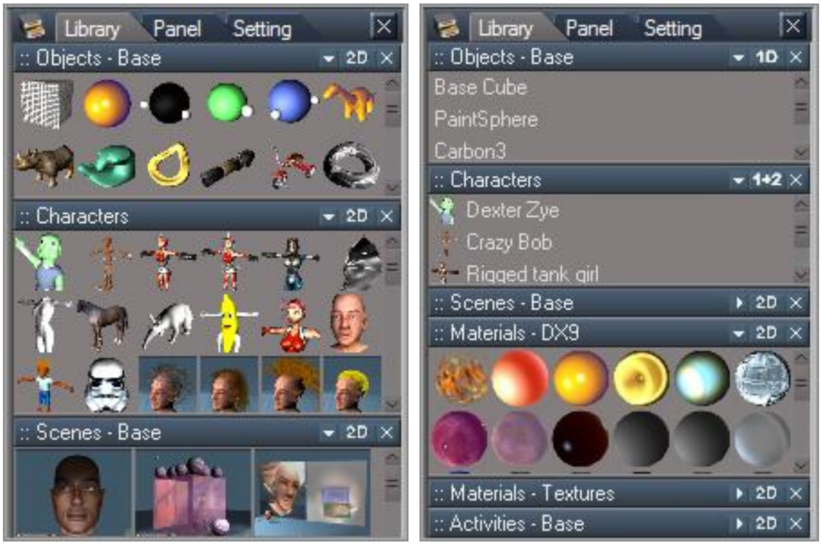
Import into Library: import items into the current selected library. This scenario is much the same as previously described for Main Library Place: Import Library. You are provided with dialogs to locate specific libraries and an import dialog to select/check items you wish to import into the current selected library.

Create Library Place: works the same as previously described for Main Library Place right-click menu.

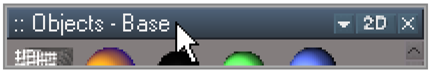
2.3.2 Library Stack



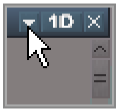
By default, the stack area of trueSpace is located along the right-hand side of the interface. The Library tab will change the Stack aspect to Library view.



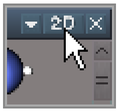
Library Stack view



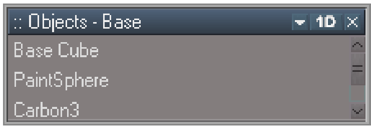
Each Library is represented by its own title-bar.



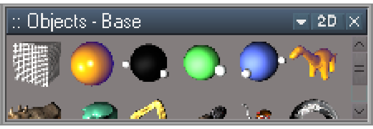
Minimize /maximize a library by clicking the white triangle, or double-click in the title-bar



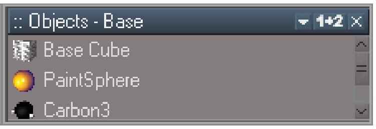
Each library can be in one of three aspects to provide a different view of the library.



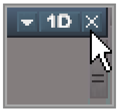
1D



2D

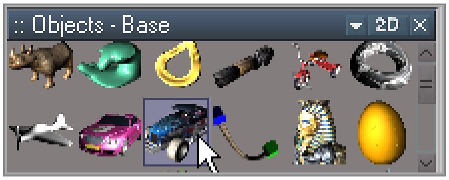


1+2.

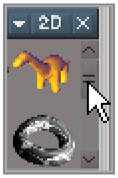


Close icon. This will close/remove the library from stack view. It doesn't delete a library from the system, the library can be re-opened again from the library browser.

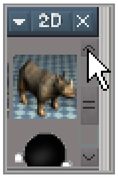
Scrolling Libraries.



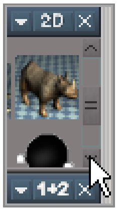
Middle Mouse wheel scroll when over an item.



Left Mouse pressed and drag or Middle Mouse wheel scroll when over the handle will scroll

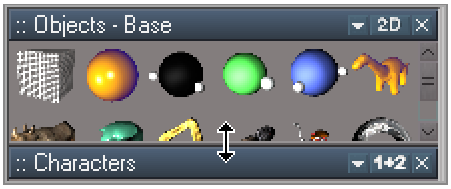


Left Mouse pressed when over the arrow will scroll continuously up , Left Mouse clicked scrolls in increments with each click.



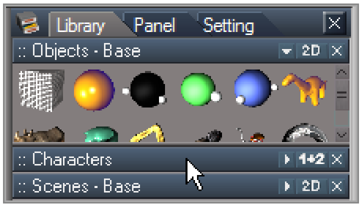
Left Mouse pressed when over the arrow will scroll continuously down , Left Mouse clicked scrolls in increments with each click.

Resizing.

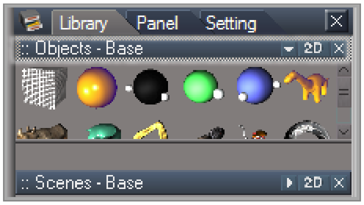


Resize Libraries by Dragging with the Left mouse on the divider between Libraries

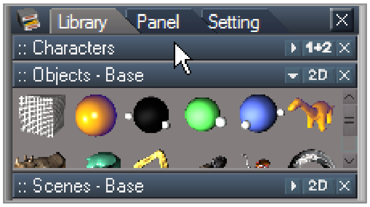
Rearranging - the library structure can be moved around by placing the cursor over the title -bar then hold the CTRL key and press the L-Mouse button to drag and drop between Libraries so the most used can be placed within easy reach. When you drag and drop you will see a target border around the drop area. Below the Characters library is dragged to the top of the stack.



Ctrl key + left-mouse drag



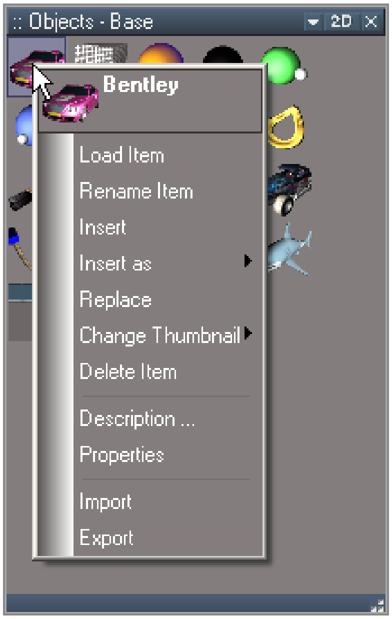
Border defines dock area



Release Mouse and drop the library

2.3.3 Library Items

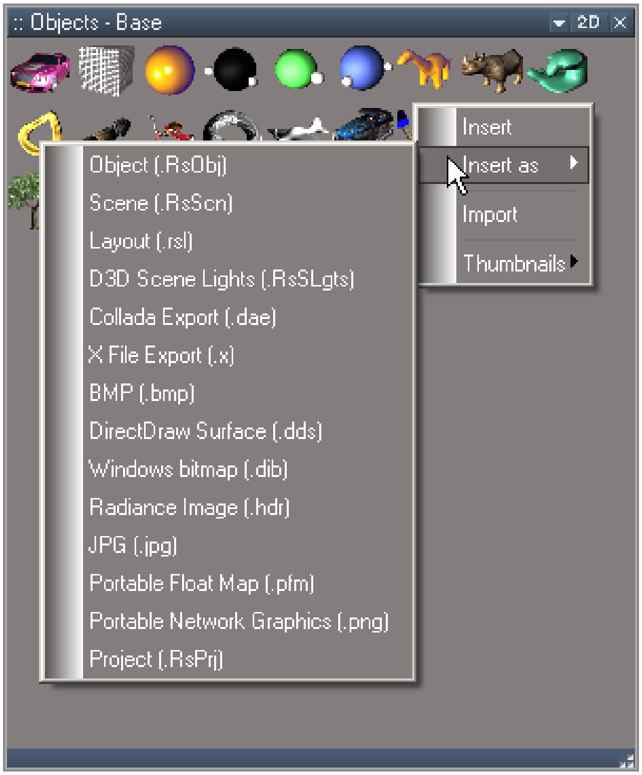
Items in libraries have a right-click menu.



Library Item right-click menu

- **Load Item**: loads the selected item into workspace.
- **Rename Item**: allows you to rename the current selected item in library.
- **Insert**: inserts the currently selected object in Link Editor/Workspace into the library. The new item will locate itself to the left or above the current selected item in library.
- **Insert as**: as we covered earlier with creation of libraries, different libraries of different types are allowed. To simply insert into a library, will insert the selected object/item as that type of

file. If however the currently selected object/item is not of the same type as the library itself, you can select the type of item to insert into the library. The illustration below indicates the available types.



Insert as extended menu items

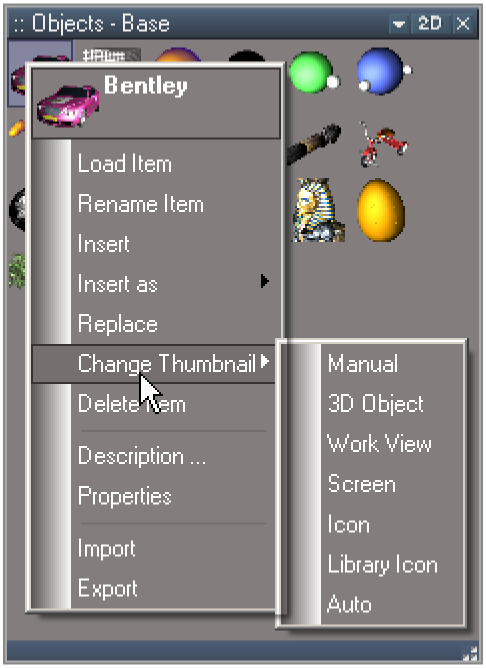
- **Object (.RsObj):** trueSpace object file format.
- **Scene (.RsScn):** trueSpace scene file format.
- **Layout (.rsl):** trueSpace layout format. most of the objects you will insert into libraries will be of this type.
- **D3D Material (.RsMat):** trueSpace D3D material file.
- **D3D Scene Lights (.RsSLgts):** trueSpace D3D scene lights.
- **Collada Export (.Dae):** Collada file format export.
- **X File Export (.x):** DirectX export format.
- **BMP (.bmp):** Standard .bmp image format.
- **DirectDraw Surface (.dds):** The DirectDraw Surface (.dds) file format is used for storing textures and cubic environment maps, both compressed and uncompressed. This makes the

format useful for Playstation 3 and Xbox 360.

- **Windows bitmap (.dib):** Windows and os/2 bitmap format.
- **Radiance Image (.hdr):** high dynamic range imaging.
- **JPG (.jpg):** jpg image file format.
- **Portable Float Map (.pfm):** PFM (Portable Float Map) is an unofficial extension to the pbm image format collection that supports HDR imaging.
- **Portable Network Graphics (.png):** PNG supports palette-based (palettes of 24-bit RGB colors) or greyscale or RGB images.
- **Project (.RsPrj):** trueSpace Project file format.
- **Skeleton (.RsSkel):** trueSpace skeleton format.

Replace: will replace the item selected in the library, with the currently selected item in the Link Editor/Workspace.

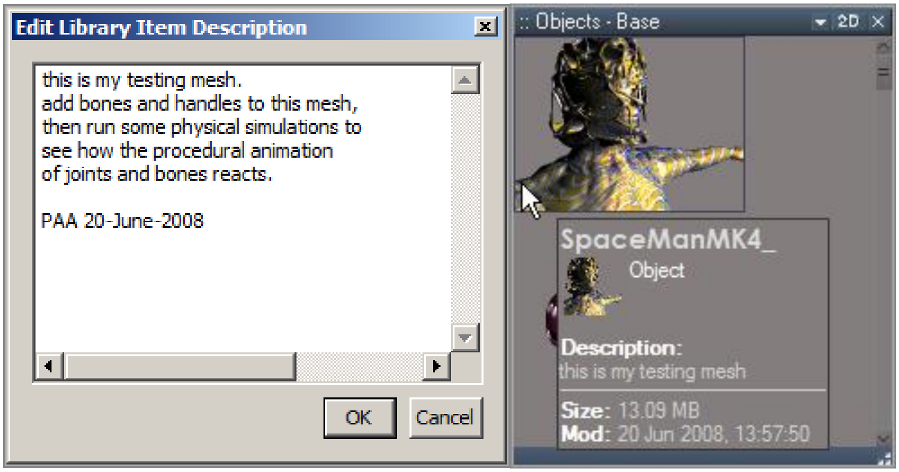
Change Thumbnail: here at the item level of libraries, the Change Thumbnail menu item has an expanded aspect to it. As illustrated below, there are a number of options for you to choose.



Change Thumbnail extended menu items

- **Manual:** select an image file manually from some location on your hard drive(s).
- **3D Object:** creates a 3D thumbnail representation of the selected object in Workspace.
- **Work View:** screen-grab of current selected window is used as thumbnail.
- **Screen:** full monitor screen-grab is used as thumbnail.
- **Icon:** default trueSpace generic type icon used as thumbnail.
- **Library Icon:** default trueSpace library icon is used.
- **Auto:**
    - If the object is visible in 3D view, the object is rendered. Furthermore, if a 3D window is open, its camera settings are used to create the thumbnail.
    - If the object is not render-able, but has 3D attributes, then only the View screen shot is used.
    - If neither of the two options above are successful/available, then node's icon is used. If the node's icon is missing as well, then the Library default icon is used.
- **Auto** for Scenes: when scenes are the selected item:
    - Active or first found 3D view is used to create thumbnail.
    - Screen shot is used if no 3D view is found.
- **Auto** for Projects, Layouts and System files:
    - Screen shot only is used.
- **Auto** for Materials, Physics and other file types.
    - Own rendering is used.
    - 3D Object.
    - Active or the first 3D view is used.
    - Object Icon.
    - Library Icon.

Delete Item: delete the currently selected library item.
Description: just as you can edit/append the description for a library or Library Place in the Library Browser, so can a library item have a description that you can edit/append.



this is my testing mesh.
add bones and handles to this mesh,
then run some physical simulations to
see how the procedural animation
of joints and bones reacts.

PAA 20-June-2008

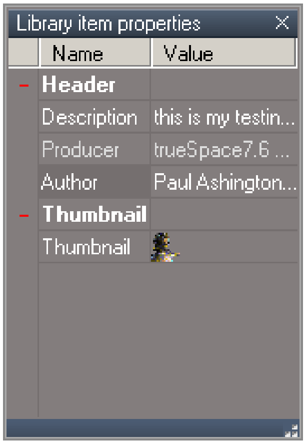
SpaceManMK4_
Object

Description:
this is my testing mesh

Size: 13.09 MB
Mod: 20 Jun 2008, 13:57:50

Editing the item description for the object

Properties: display and edit/append the property information as illustrated below. Changes made are saved when you close the panel. The default properties displayed may change from time to time as developers add more features. Basic edit is easy enough; double-click the value area of properties to edit.



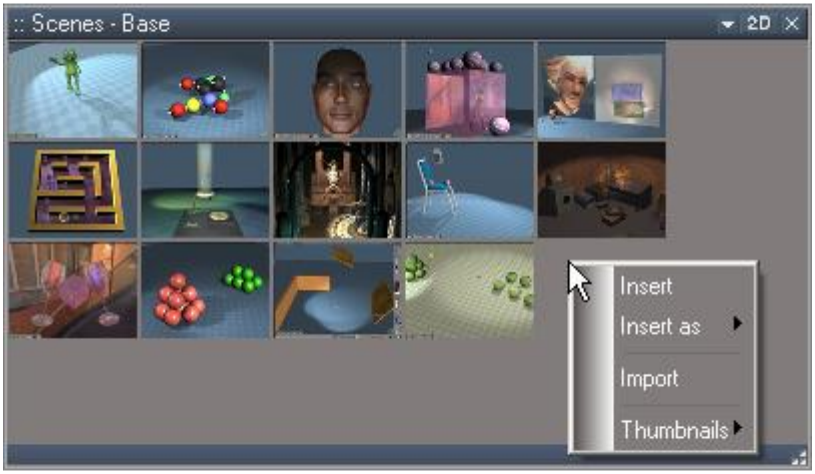
Name	Value
− **Header**	
Description	this is my testin...
Producer	trueSpace7.6 ...
Author	Paul Ashington...
− **Thumbnail**	
Thumbnail	

Library item properties being edited

Import: allows you to import items from libraries on your hard drive. Much the same as previously discussed for adding items to libraries in the Library Browser.

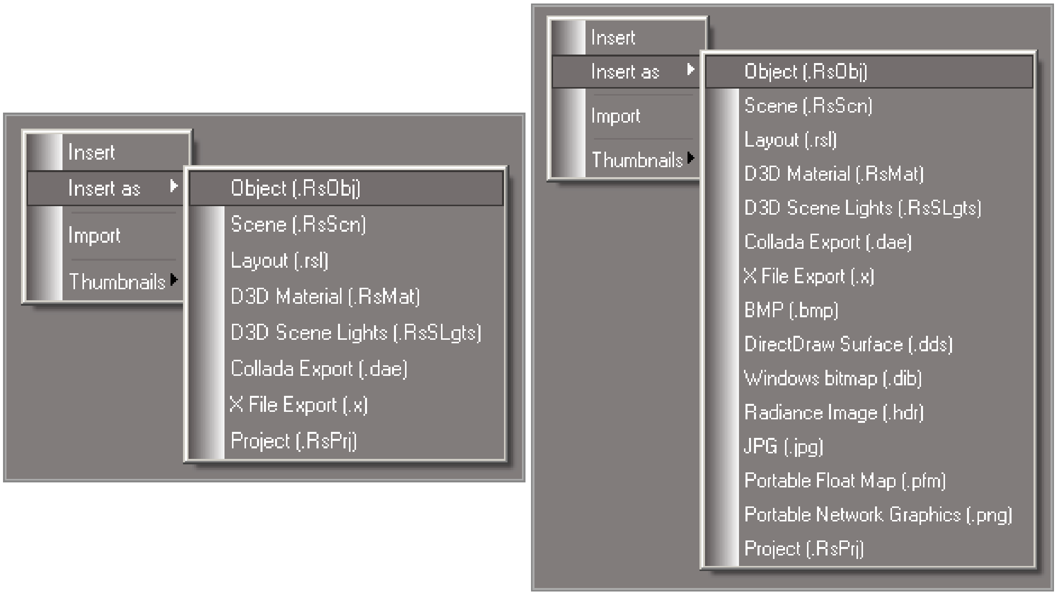
Export: exports the currently selected library item to hard drive location of your choice.

Within the libraries themselves is another menu, which you access by right-click in an open area of the library.



Right-click open/empty area of library to access menu items

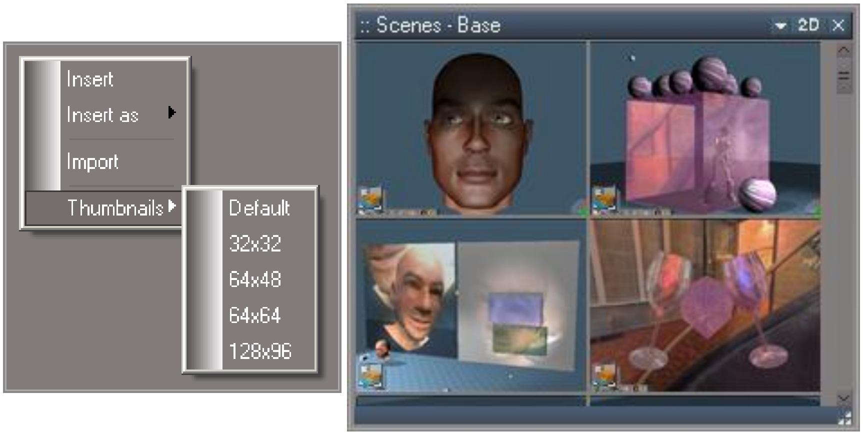
Insert: inserts currently selected object in Link Editor/Workspace into the library as library's type.
Insert as: depending what you have selected in the Link Editor/Workspace, the Insert as menu will populate with options.



Depending on what type of object is selected in LE or Workspace, menu items populate accordingly

Object (.RsObj): trueSpace object file format.
Scene (.RsScn): trueSpace scene file format.
Layout (.rsl): trueSpace layout format.
D3D Material (.RsMat): trueSpace D3D material file.
D3D Scene Lights (.RsSLgts): trueSpace D3D scene lights.
Collada Export (.Dae): Collada file format export.
X File Export (.x): DirectX export format.
BMP (.bmp): Standard .bmp image format.
DirectDraw Surface (.dds): The DirectDraw Surface (.dds) file format is used for storing textures and cubic environment maps, both compressed and uncompressed. This makes the format useful for Playstation 3 and Xbox 360.
Windows bitmap (.dib): Windows and os/2 bitmap format.
Radiance Image (.hdr): high dynamic range imaging.
JPG (.jpg): jpg image file format.
Portable Float Map (.pfm): PFM (Portable Float Map) is an unofficial extension to the pbm image format collection that supports HDR imaging.
Portable Network Graphics (.png): PNG supports palette-based (palettes of 24-bit RGB colors) or greyscale or RGB images.
Project (.RsPrj): trueSpace Project file format.
Skeleton (.RsSkel): trueSpace skeleton format.

Import: refers to library’s ability to import objects from other libraries or locations on your hard drive. Standard file selection dialog is presented for navigation and selection of library or items/objects.
Thumbnails: You can change the size of thumbnails in the 2D aspect of a library.



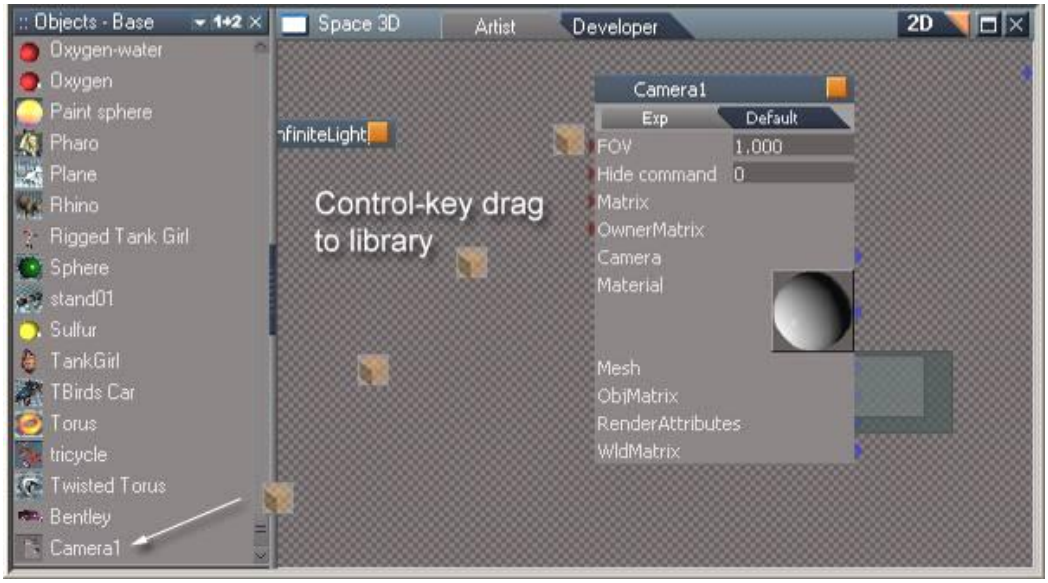
2D (aspect) library item's icon display size

2.3.4 Saving Items to a Library

You can drag and drop objects to a library from any window that will allow it. There are two useful sources:

From the Link Editor:

1. Hold CTRL.
2. Left-click on the object's title bar and hold the button down.
3. Drag the object into a library.
4. Release the mouse button.



FOV 1.000

Hide command 0

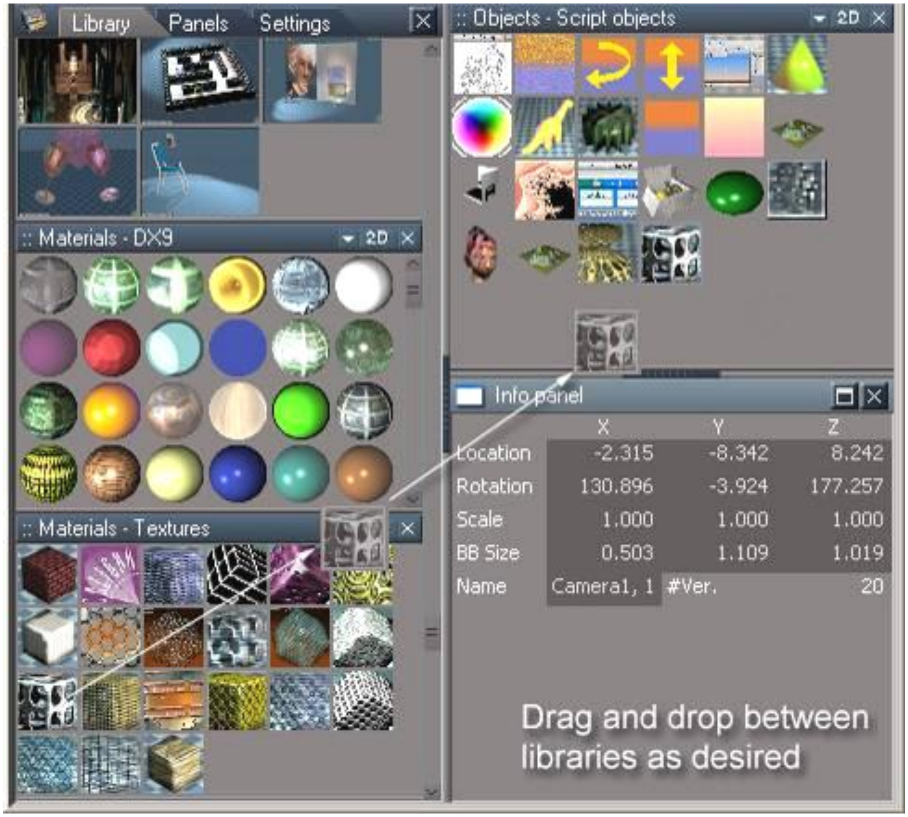
Control-key drag to library

Control-key + dragging objects from the Link Editor to a library

In the example above, a camera is being added to the Objects library.

From other libraries:

1. Left-click on the item you want to copy and hold the button down.
2. Drag this object into the destination library.
3. Release the mouse button.



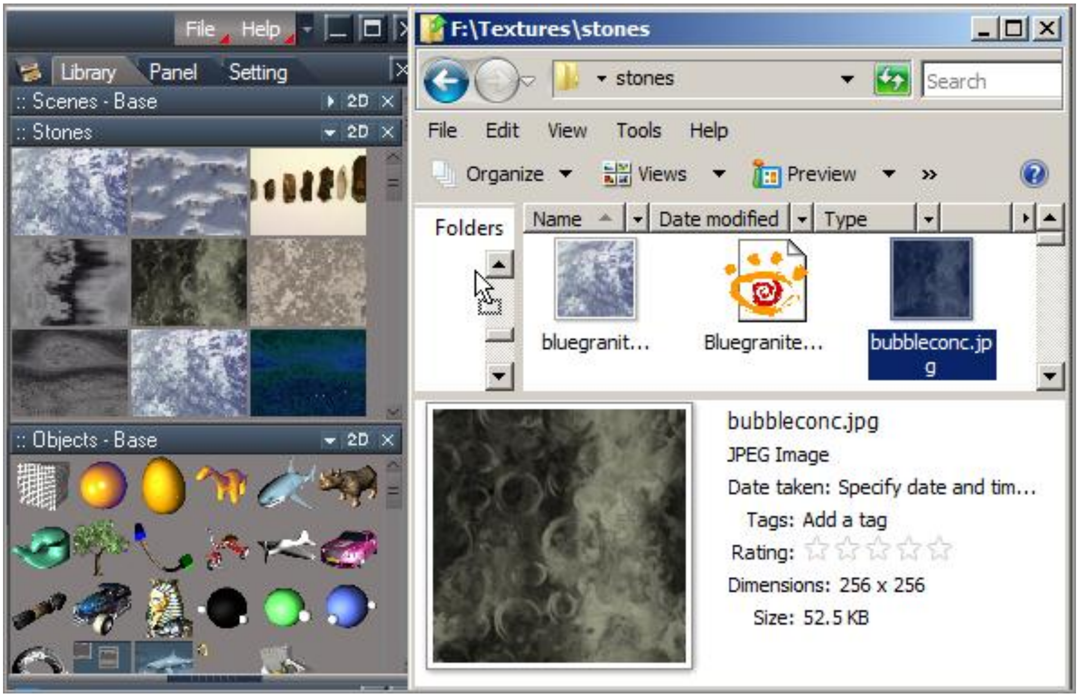
	X	Y	Z
Location	-2.315	-8.342	8.242
Rotation	130.896	-3.924	177.257
Scale	1.000	1.000	1.000
BB Size	0.503	1.109	1.019
Name	Camera1, 1	#Ver.	20

Drag and drop between libraries as desired

Drag and drop between libraries

Note that in the example above we added a material to a 3D object library. This is perfectly legal in trueSpace, where object categories are entirely up to you and your intentions.

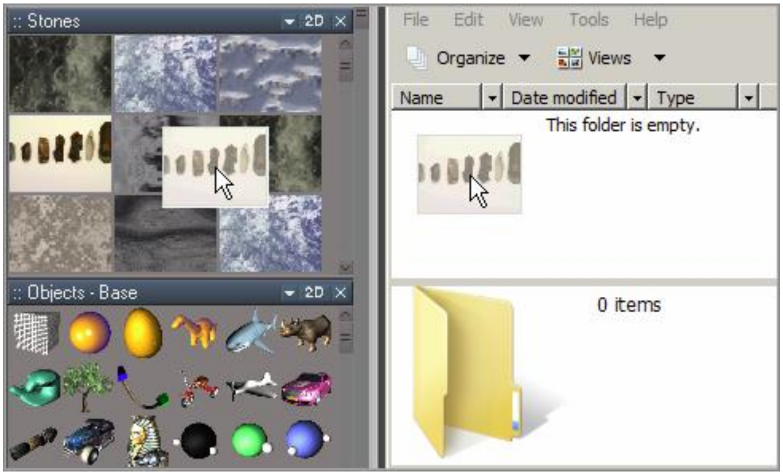
Libraries can be populated by dragging items from your hard drive directly from a Windows Explorer valid items can also be dragged and dropped into the Link Editor or the Workspace.



bubbleconc.jpg
JPEG Image
Date taken: Specify date and tim...
Tags: Add a tag
Rating: ☆☆☆☆☆
Dimensions: 256 x 256
Size: 52.5 KB

Drag and drop items from your hard drive to a library

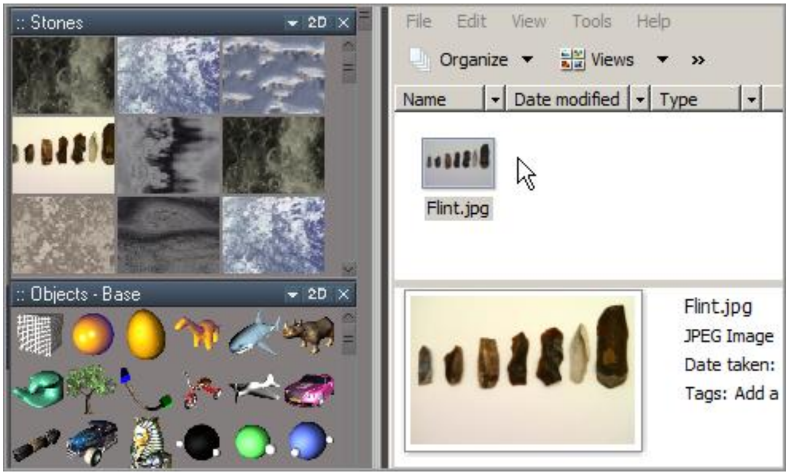
It is also possible to drag and drop items from within a trueSpace library, into your Windows Explorer.
1. Select an item then Use Left mouse button and Drag the item to a folder.



This folder is empty.
0 items

Drag from library to a folder

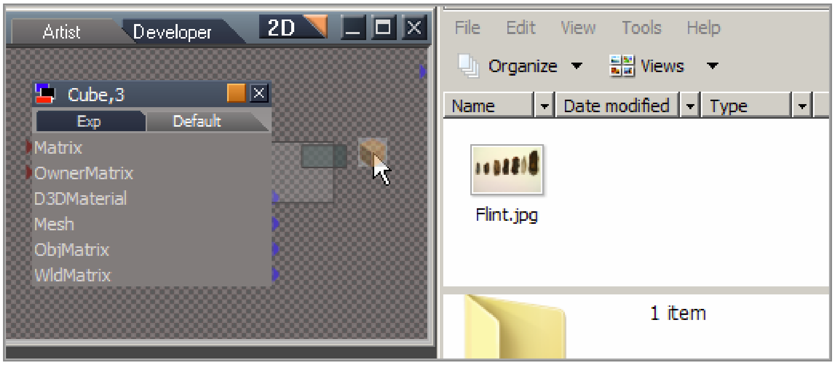
2. Release the mouse in the target destination then use a single left click to drop.
The Item is created on Left mouse click in target folder .



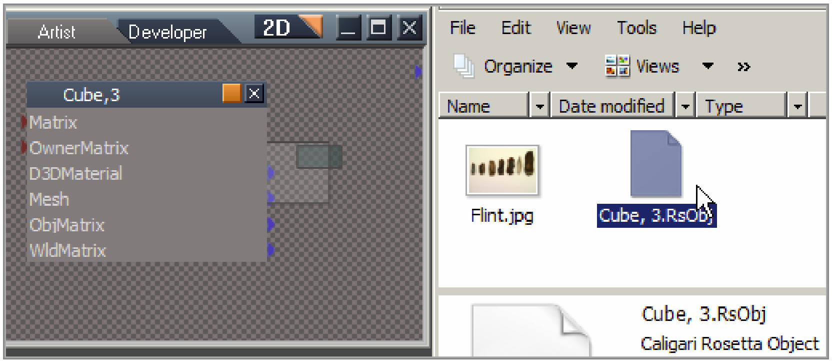
Item created on drop and Left - Mouse click

Objects can be dragged from the Link-editor and the Workspace to Windows Explorer folders.

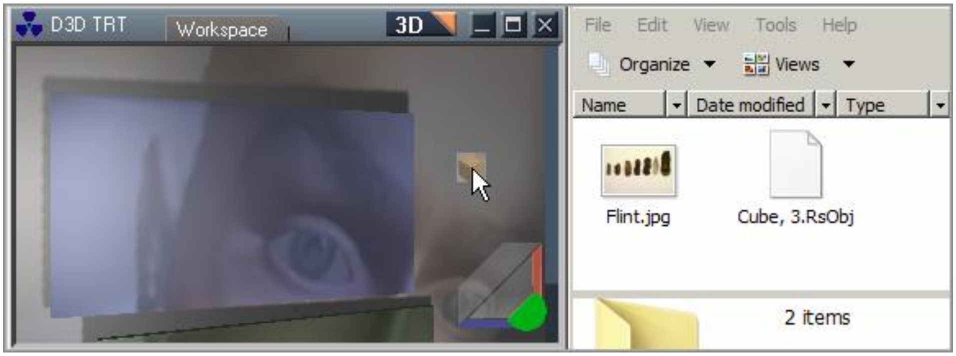
1. Select an item then Use CTRL key + Left mouse button and Drag the item to a folder.
2. Release the mouse in the target destination then use a single left click to drop.
The Item is created on Left mouse click in target folder .



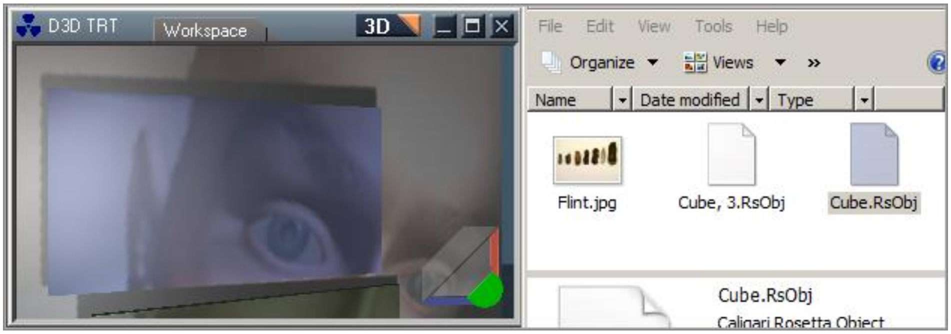
CTRL key + Drag from the Link Editor to a folder



Item created on drop and Left - Mouse click



CTRL key + Drag from the Workspace to a folder



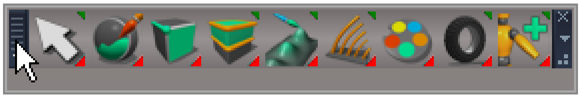
Item created on drop and Left - Mouse click

2.4 Toolbars

A **toolbar** is a simple form of a library. It is a one-level, 2D container for storing unlinked icons that typically activate some tool. All icons are of the same type, and they “stick” to each other along their edges. A toolbar container always “snaps” tight around the icons inside of it.

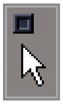
Toolbar icons are normally darkened or de-saturated in color by default De-saturated icons represent a “not applicable” state of the tool. They will show full-color when in an “applicable” state. Icons will highlight when you mouse-over them.

You can minimize and maximize a toolbar by double-clicking its handle.
Below is a simple toolbar whose items are shown in their 2D aspect as Icons.

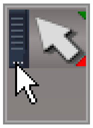


Maximized toolbar .

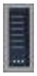
Minimize by a Double Click on the Handle



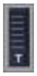
Minimized toolbar. Maximize by Double Click on the Dot .



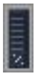
Fix a toolbar’s position by pinning it. The toolbar’s Handle will indicate what state it's in.



Floating: toolbar floats on the desktop.

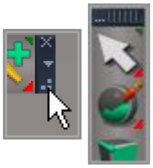


Pinned: the toolbar is pinned in location/relation to trueSpace.

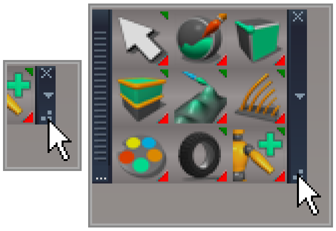


Snap to edge: when in this state, the toolbar can be snapped to trueSpace window edges. If the toolbar approaches a corner of trueSpace, it will snap to both edges and place itself nicely into a corner.

- When actually snapped to edge, Floating state is not available for the toolbar. Move the toolbar away from edges to access Floating state for toolbar.



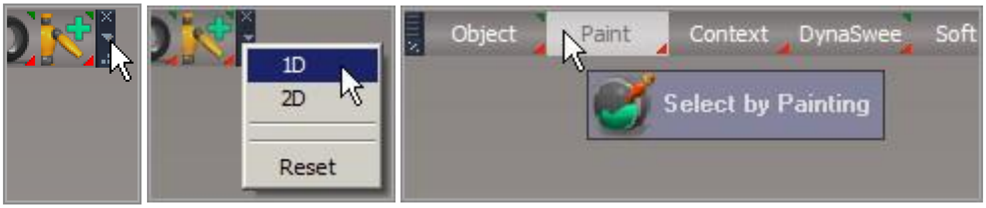
Rotate 90 degrees by left-click on the bottom right corner.



Resize by dragging the bottom right corner.

Set the toolbar to 1D text or 2D icon . Use the white triangle to show the toolbar's menu. to return the original toolbar to default state/settings, select the Reset option.

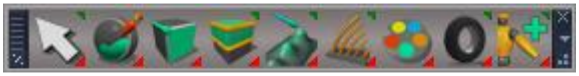
Note- The Reset option is only available on native trueSpace toolbars. Custom toolbars do not display this option.



Toolbars can also be reconfigured from 2D icon to text state by choosing 1D

2.4.1 Pop-Up Toolbars

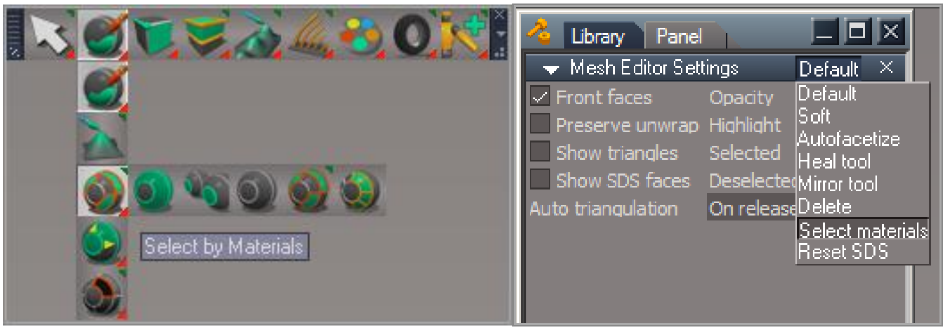
When you press and hold a mouse button on an icon that has a red triangle in the corner, its pop-up toolbar is displayed. The placement of the inner toolbar depends on the current position of the outer toolbar.



Toolbar with active icons

The inner toolbars of a horizontal toolbar are expanded upwards, and the inner toolbars of a vertical toolbar are expanded to the right. If there is not enough space above or to the right of the toolbar, then a downward or leftward direction is used. This is similar to how pop-up menus work in Microsoft Windows. You must expand the inner toolbar and select the desired icon to activate the command. The button representing a sub-toolbar changes to the last-used button of that toolbar for easier access to frequently-used functions.

Some buttons have a small green triangle in upper right corner, this indicates that you can open an attribute panel for this tool in Stack. You can also use Right mouse Drag on such buttons to open a pop-up toolbar without actually activating the tool.



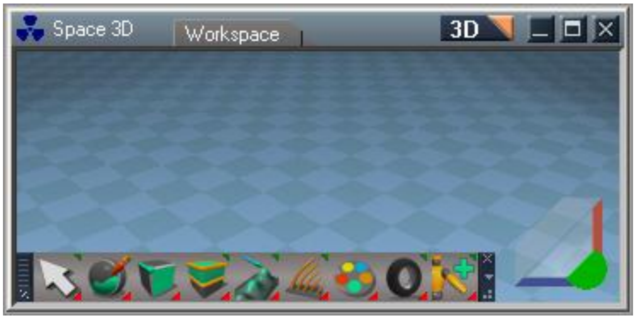
Mesh Editor Settings — Default

☑ Front faces — Opacity
□ Preserve unwrap — Highlight
□ Show triangles — Selected
□ Show SDS faces — Deselected
Auto triangulation — On release

Red triangle indicates an available pop-up toolbar, green triangle shows a panel on r-click

2.4.2 Snapping Toolbars

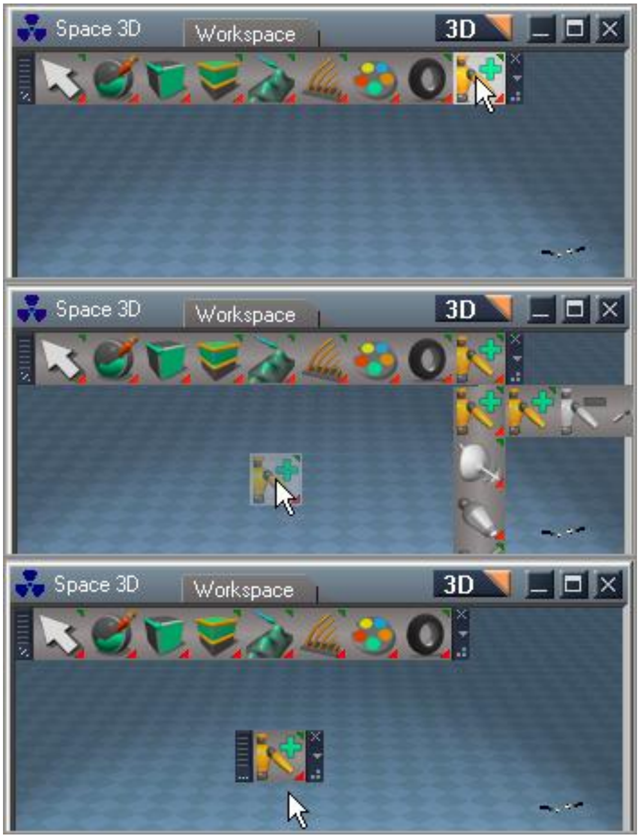
When a toolbar is dragged, it snaps to the nearest edge or corner of a nearby window. When you do snap a toolbar, it will retain its position as you resize the main trueSpace window.



Toolbar snapped to corner of window

2.4.3 Toolbar Drag and Drop

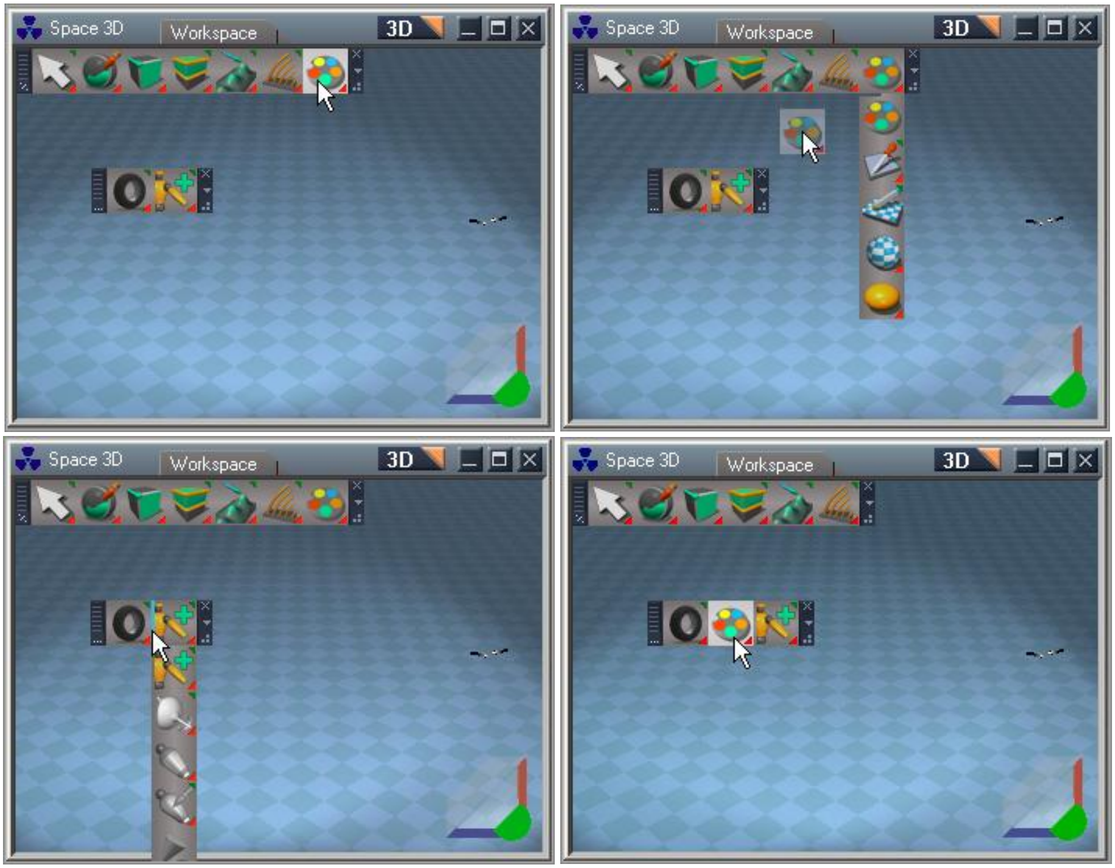
To create new toolbars, Control-drag the button outside any toolbar or library and drop it into the Workspace. The original toolbar will be missing the icon/tool afterwards. To create a copy Control-Shift-drag the icon into the workspace which leaves the original toolbar intact.



Control-drag toolbar button to create a new toolbar

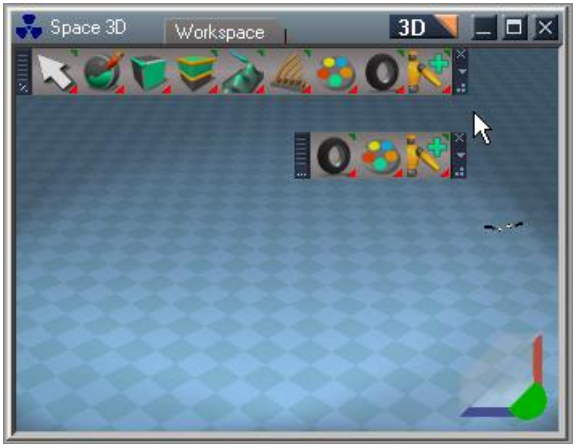
New toolbars have an “x” in upper corner used to close them.

Re-arrange toolbar buttons by dragging and dropping.
Left-click and drag a button while holding the CTRL key. The drag target’s position is indicated by a blue line.



Rearranging toolbar icons

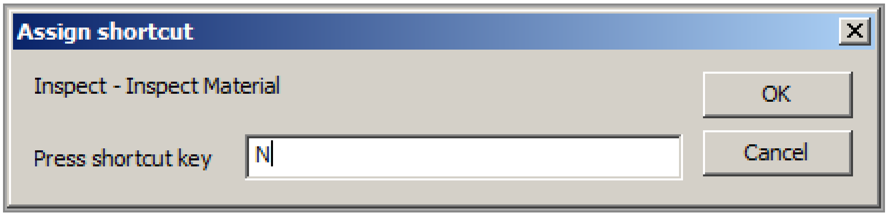
By holding the CTRL key while dragging, the button will be moved and the source button deleted. To make a duplicate copy of the source button, hold SHIFT+CTRL instead.



Use SHIFT+CTRL and drag to create a copy of the tool

2.4.4 Shortcut customization

Assign a shortcut to a given toolbar button by pressing CTRL and right-clicking on the button. A dialog opens to change or set the shortcut.


Assign shortcut

Inspect - Inspect Material

Press shortcut key: N

Assign shortcut for Material Pick tool

The dialog shows information about the currently assigned shortcut for the button (if any), as well as an edit field where you can enter a new shortcut by pressing the key combination. If this shortcut is already assigned, then the current assignment is shown in the edit field as well.

Advanced shortcut editing is possible outside of trueSpace by editing the Window.Shortcuts.xml file located in the Scripts directory. We recommend this approach only to experienced users. A list of all possible names for shortcut keys is in the file shortcut_description.txt, and a backup of the default shortcut list is stored in the file Window.Shortcuts.default.

2.4.5 Preset Shortcuts

trueSpace comes with a set of predefined shortcuts for Workspace, Link Editor, and some that work in every view. The following table shows the list of these predefined shortcuts.

Workspace shortcuts	
Key	**Description**
DELETE	Delete selected object
CTRL K	Delete selected object
CTRL M	Start the material pick tool
CTRL R	Render to File
Z	Object move tool
X	Object rotate tool
C	Object scale tool
A	Camera move Tool
S	Camera Rotate Tool
D	Camera Field of view Tool
Q	Set object coordinate system
W	Set world coordinate system
E	Set screen coordinate system
CTRL ~	Set the selected camera as active
CTRL 1	Switch to Perspective Eye view
CTRL 2	Switch to top orthogonal view
CTRL 3	Switch to bottom orthogonal view
CTRL 4	Switch to front orthogonal view
CTRL 5	Switch to back orthogonal view
CTRL 6	Switch to left orthogonal view
CTRL 7	Switch to right orthogonal view
CTRL 8	Switch to Nearest orthogonal view
CTRL 9	Reset View
CTRL Z	Undo
CTRL SHIFT Z	Redo
CTRL C	Copy object
ARROW KEYS	Navigate Hierarchy
SHIFT X	Toggles X axis lock on and off
SHIFT Y	Toggles Y axis lock on and off
SHIFT Z	Toggles Z axis lock on and off

Workspace Point Edit shortcuts	
Key	**Description**
ESC	Quits Tool and returns to selection mode
SHIFT DEL	Collapse Loop
CTRL F	Form Face
TAB	Toggles PE Widget visibility
V	Select Vertices
F	Select Faces
E	Select Edges
L	Select Face Loops
T	Select Context
SHIFT 4	Set Tangent coordinate system
NUM +	Grow Selection
NUM -	Shrink Selection
SHIFT Q	Select Connected
SHIFT C	Convert Selection
SHIFT S	Smooth Selection
SHIFT O	Outline Selection
CTRL I	Invert Selection
CTRL S	Add SS
CTRL SHIFT S	Subtract SS
1	Pick and Move selection
2	Paint Selection
3	Lasso Selection
4	Rectangle Selection
5	Soft Selection
6	Select by Material

Link Editor shortcuts	
Key	**Description**
DELETE	Delete selected object
CTRL Z	Undo
CTRL SHIFT Z	Redo
CTRL C	Copy object
ARROW KEYS	Navigate Hierarchy

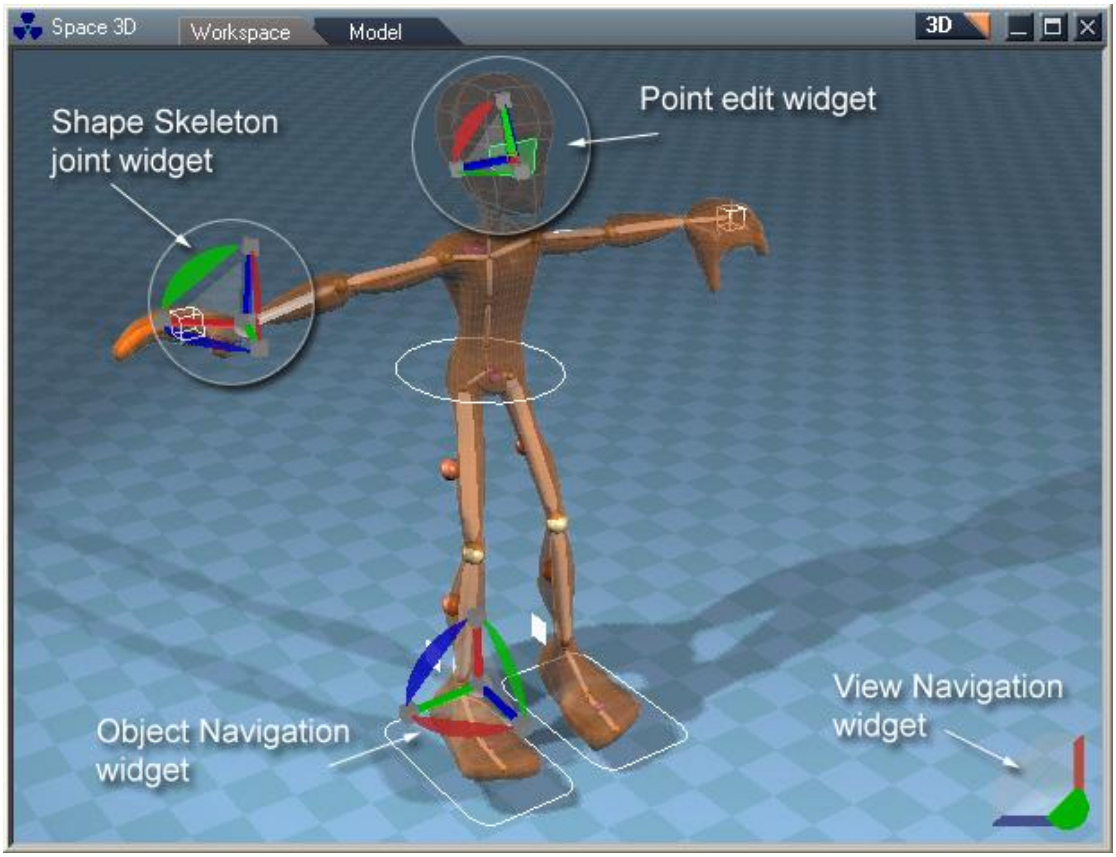
Anim View shortcuts	
Key	**Description**
LEFT ARROW	Previous keyframe
RIGHT ARROW	Next keyframe
CTRL K	Set keyframe
CTRL C	Copy Object
CTRL X	Delete Object
CTRL V	Paste Object
CTRL A	Select All
CTRL Z	Undo
CTRL SHIFT Z	Redo

Global shortcuts	
Key	**Description**
SHIFT CTRL F1	Open 3D window
SHIFT CTRL F2	New LE window
SHIFT CTRL F3	New Output console window
SHIFT CTRL F4	Open Progress panel
SHIFT CTRL F5	Open History Window
SHIFT CTRL F6	Open Stack View
SHIFT CTRL F7	Command prompt
SHIFT CTRL F10	Open Network connection
SHIFT CTRL R	Reset to default context
SHIFT CTRL N	Generate new space

UV editor shortcuts	
Key	**Description**
CTRL Z	Undo
CTRL SHIFT Z	Redo

2.5 Widgets

trueSpace provides visible 3D control objects, called **widgets**, that allow you to manipulate objects, views, and other aspects of your 3D space. Widgets respond to your mouse clicks and movements and translate that input into movement, rotation, and other actions. The image below shows some of the many controllers in the 3D Workspace view. The Object Navigation and View Navigation widgets are two of the most commonly-used widgets in trueSpace.



trueSpace offers many different widgets

trueSpace widgets are easy to use and can be easily customized to suit your needs. You can change the widget's shape, color, control surfaces, location, inputs and actions, and more. You can even create a new widget from scratch, without a single line of programming or scripting. See **DEVELOPER'S GUIDE**

CHAPTER 2: WIDGETS for more information on customizing and building widgets.

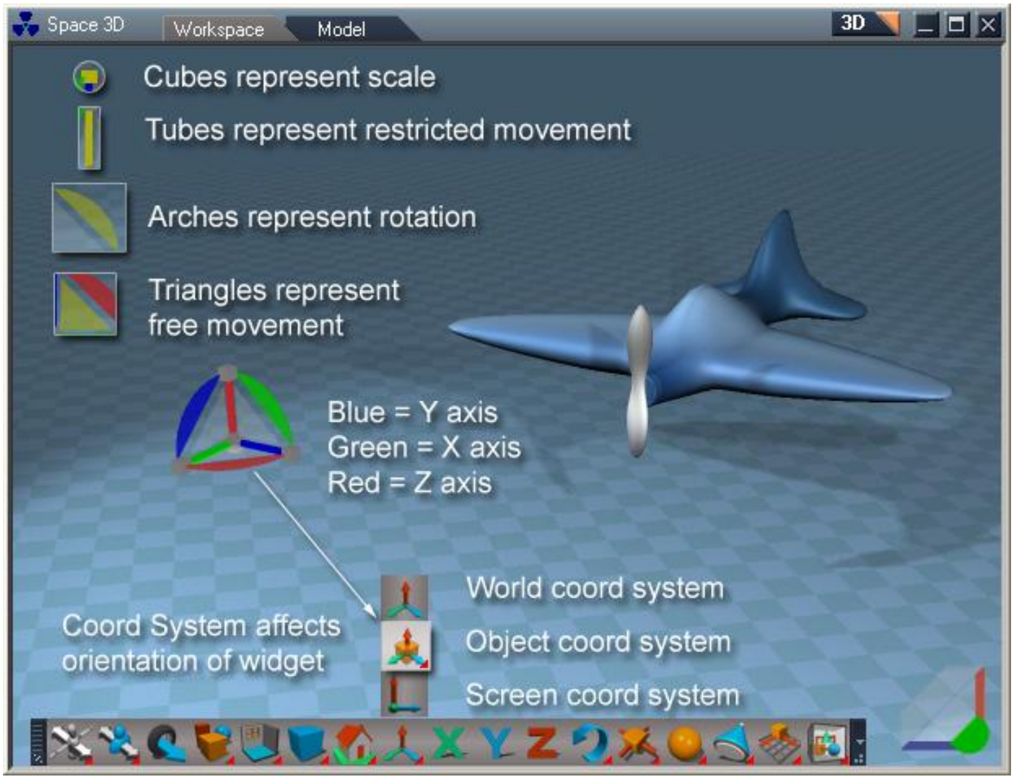
2.5.1 Using Widgets

To use a widget, simply left-click and drag on the part of the widget that controls the action you want to perform. Additional actions may be available by holding the CTRL or SHIFT keys while you click and drag. Actions controlled by the widget depend on the particular widget you are using.

Usable control surfaces will highlight in yellow as you mouse over them. This is important feedback, allowing you to see which action you are about to perform.

The Object Navigation Widget

The object navigation widget was designed to help you manipulate the position, orientation, and scale of an object. The image below shows which control surfaces control which actions.

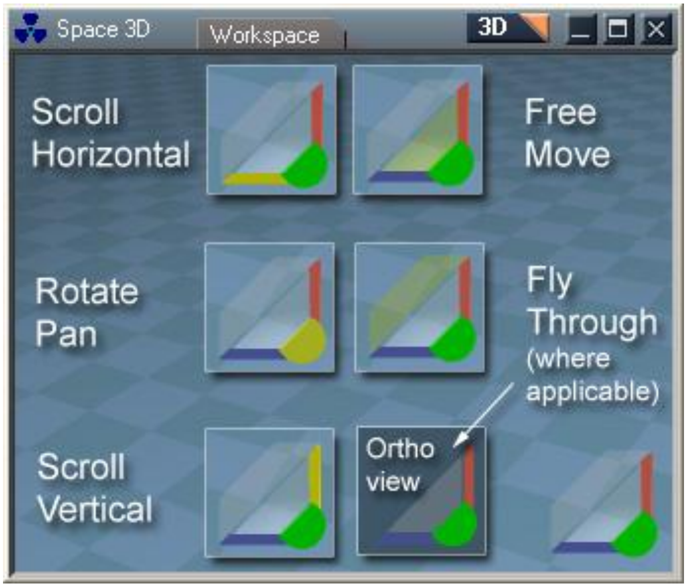


When you click and drag on any of the object navigation widget's control surfaces (move, scale, etc.) the other surfaces will be hidden from view until you release the mouse button. This gives you a clear visual indicator of which action you are currently controlling. The Object Navigation widget's orientation depends on which Coordinate System is currently selected. Take a few moments to select different Coord System options, to grasp how the widget orients itself.

Triangle surfaces, although described as free-movement surfaces, will actually restrict movement to the axis whose arch surface is directly beside the triangle. Right-click-drag to move in a restricted mode. The cube surfaces represent scale. Scale is restricted by default to the axis the cube is attached to. You can however scale uniformly in all axis directions by simultaneously holding down both left and right mouse buttons and dragging your mouse.

The View Navigation Widget

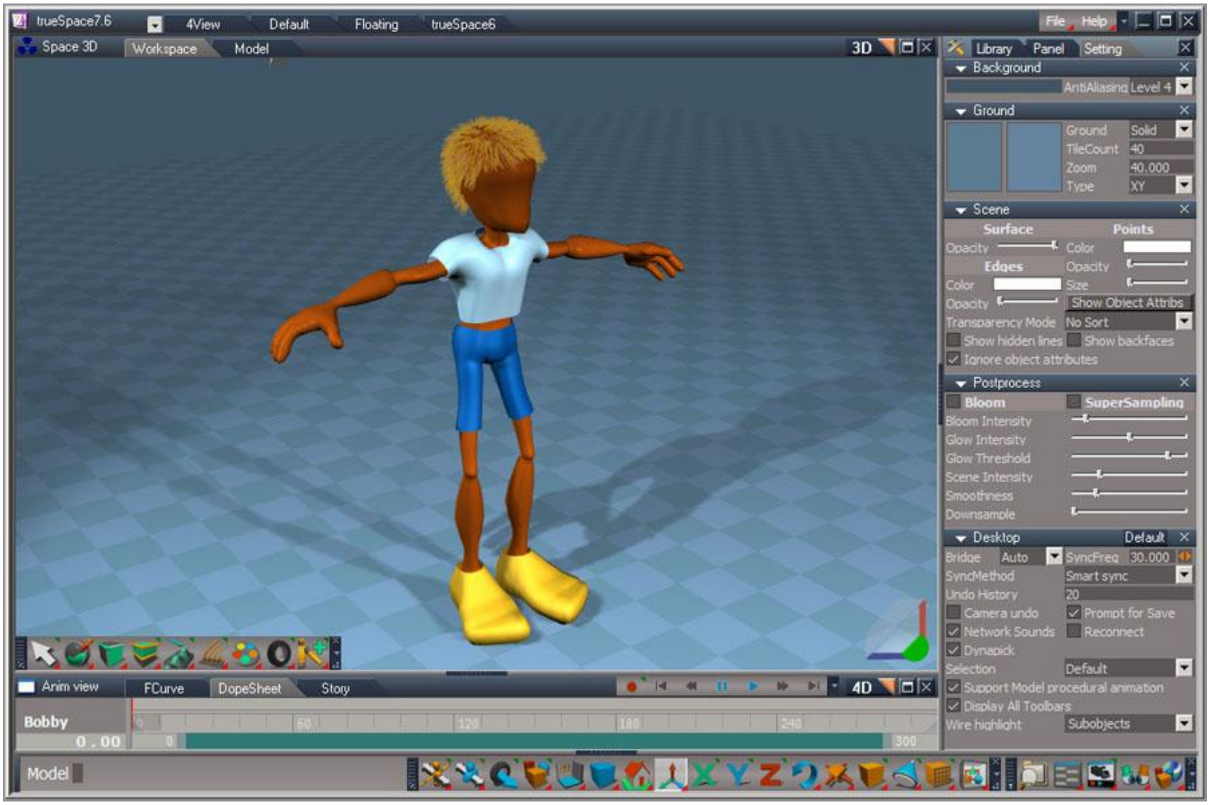
The view navigation widget is used for moving your own view through 3D space. With this widget you can move, pan, and rotate the view, or make the view circle around the selected object. The image below shows which control surfaces control which actions for the View Widget.



Control surfaces of the View Navigation Widget

2.6 Stack View

Stack view is a large vertical window shown in Default view docked on the right side of trueSpace. Its purpose is to contain and display your libraries, object information and important settings. Stack has three aspects: **Library** aspect, **Panels** aspect and **Settings** aspect. Switching between aspects is accomplished through the buttons in the title bar of the Stack.



Background
AntiAliasing Level 4

Ground
Ground Solid
TileCount 40
Zoom 40.000
Type XY

Scene
Surface
Opacity
Edges
Color
Opacity
Points
Color
Opacity
Size
Show Object Attribs
Transparency Mode No Sort
□ Show hidden lines □ Show backfaces
☑ Ignore object attributes

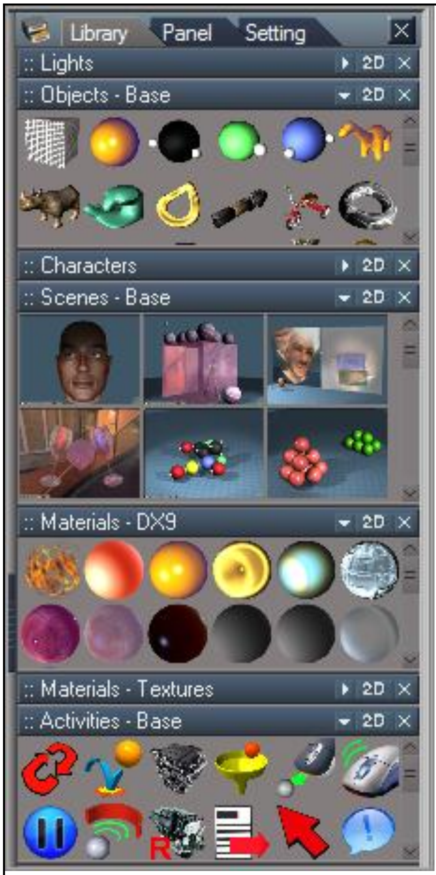
Postprocess
□ Bloom □ SuperSampling
Bloom Intensity
Glow Intensity
Glow Threshold
Scene Intensity
Smoothness
Downsample

Desktop Default
Bridge Auto SyncFreq 30.000
SyncMethod Smart sync
Undo History 20
□ Camera undo ☑ Prompt for Save
☑ Network Sounds □ Reconnect
☑ Dynapick
Selection Default
☑ Support Model procedural animation
☑ Display All Toolbars
Wire highlight Subobjects

Stack View on right-hand side of trueSpace

You can use also this area as a standard docking zone for panels and libraries, but be aware that your panel or library will be moved down when new panels are added. Also, if there are more than 20 panels or libraries, then bottom panels and libraries will be closed.

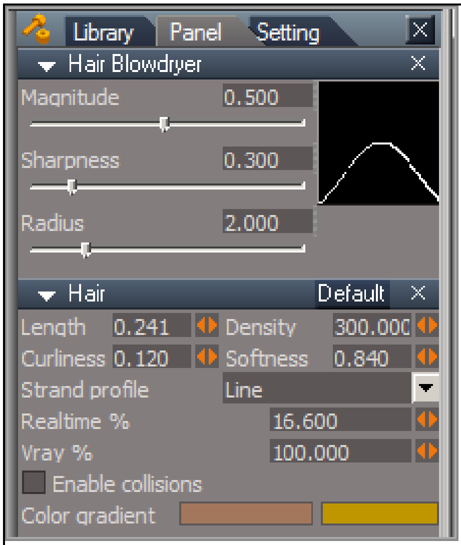
Library Aspect



The Library aspect houses various libraries, which can be loaded via the Library Browser. When you do load a library, it will appear at the top of the Library aspect. If at first glance you do not see the newly loaded library, use the scroll bar at right-hand side of the Stack. It will allow you to scroll as required to see all libraries currently loaded in the stack.

The image on left shows some of the libraries loaded by default. If the individual libraries are not fully expanded, you will see inner scroll bars, which are used to scroll the individual library as required to see all the items contained in the library.

Panels Aspect



Hair Blowdryer

Magnitude 0.500

Sharpness 0.300

Radius 2.000

Hair Default

Length 0.241 Density 300.00C

Curliness 0.120 Softness 0.840

Strand profile Line

Realtime % 16.600

Vray % 100.000

□ Enable collisions

Color gradient

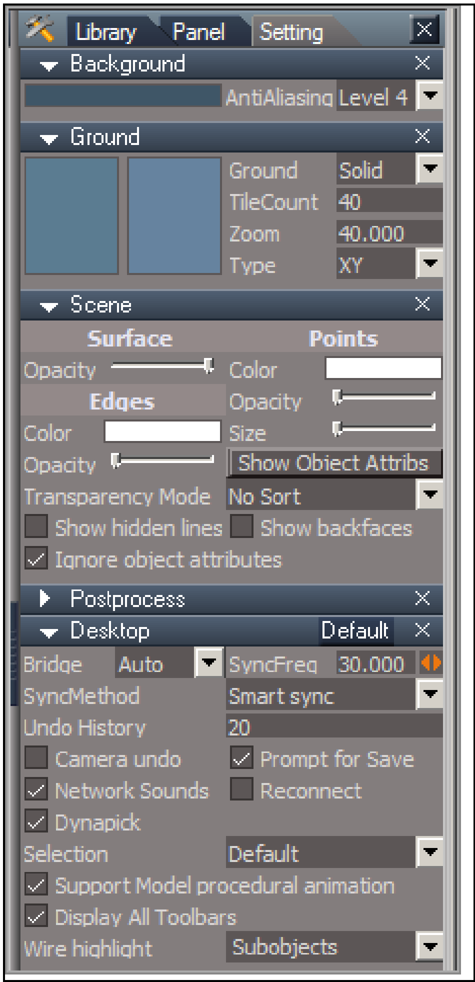
The Panels aspect displays panels of currently selected tool and currently selected object.

Each panel in turn can have multiple aspects itself which are selectable from the pop-up in upper right corner.

Each aspect presents you with a convenient set of UI controls with which you can change the attributes of an object or tool.

Aspects are created by author of each object. This is easy to do and you too could create new aspects using Panel Editor.

Settings Aspect



Library Panel Setting

Background

AntiAliasing Level 4

Ground

Ground Solid

TileCount 40

Zoom 40.000

Type XY

Scene

Surface

Opacity

Edges

Color

Opacity

Points

Color

Opacity

Size

Show Object Attribs

Transparency Mode No Sort

□ Show hidden lines □ Show backfaces

☑ Ignore object attributes

Postprocess

Desktop Default

Bridge Auto SyncFreq 30.000

SyncMethod Smart sync

Undo History 20

□ Camera undo ☑ Prompt for Save

☑ Network Sounds □ Reconnect

☑ Dynapick

Selection Default

☑ Support Model procedural animation

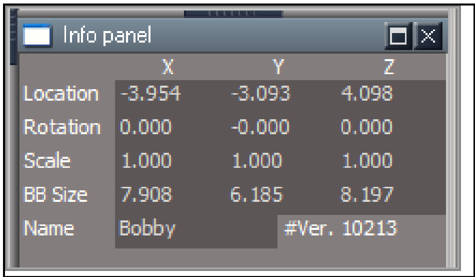
☑ Display All Toolbars

Wire highlight Subobjects

The Settings aspect of the stack houses attributes for the currently selected window in trueSpace.

The illustration to the left is displaying settings for a 3D Window in trueSpace. The Settings aspect is considered Context Sensitive in that it will display information for whichever window you have selected.

Info Panel



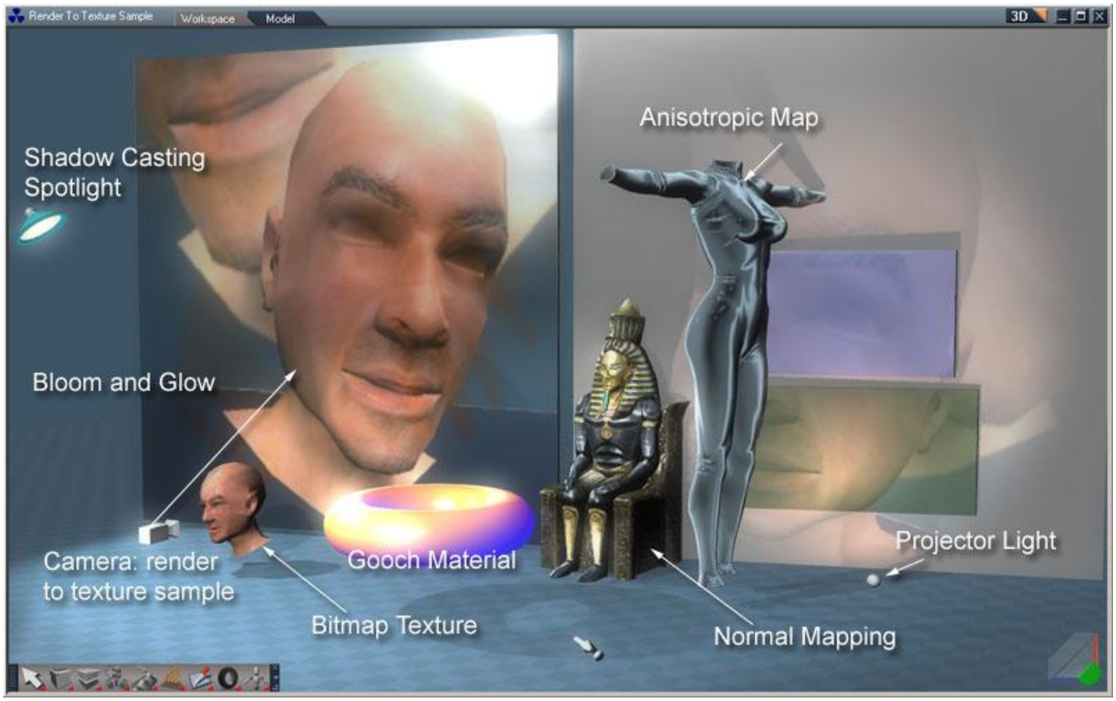
Info panel

	X	Y	Z
Location	-3.954	-3.093	4.098
Rotation	0.000	-0.000	0.000
Scale	1.000	1.000	1.000
BB Size	7.908	6.185	8.197
Name	Bobby	#Ver. 10213	

The Info panel will by default, locate itself under the Stack View. The current selected object’s information is contained here in regards to Location, Rotation, Scale, Bounding Box Size, Name and number of Vertices in the object. If you close the Info panel, you can easily access it by right-click on the Object tool in Workspace.

2.7 3D View – Workspace

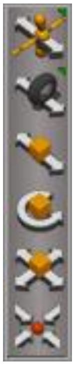
The trueSpace 3D Workspace view supports full Microsoft DirectX9 functionality for the real-time display of **transparency**, **normal maps**, **shading**, **pixel and vertex shaders**, and other effects. Complex lighting models, such as **real-time shadows**, and post-processing effects such as **glow** and **bloom**, allow you to pre-visualize your scene with complex materials, lighting, etc., before you render a final image.



The Workspace view shows realistic materials and lighting with real-time feedback

2.7.1 Object Navigation Tools

You can either use the object navigation widget to manipulate your objects, or you can use the navigation icons (**Move**, **Rotate**, and **Scale**), which are accessible from the navigation toolbar shown below. Each Workspace window includes a navigation toolbar by default.



Dynamic Pose

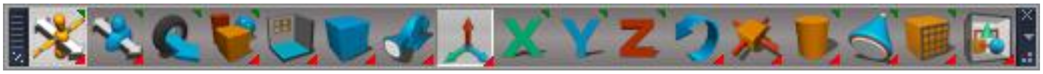
Phys Move

Move

Rotate

Scale

Look At



Dynamic Pose The Dynamic Pose tool allows you to manipulate characters in real-time. The Dynamic Pose tool is covered in more detail in Chapter 8: Characters.

Physics Move

You can use the **Physics Move** tool during a simulation to set the **speed** of objects with physical attributes. To interact with physical objects during a simulation, you must first select the Physics Move Tool from the navigation toolbar. (The tool remains active until another tool is selected.) Then, click and drag on any physical object in the Workspace view. If the object has physical attributes and a simulation is running, then the object will change speed according to the speed and direction of the mouse movements. Ref: Chapter 10 Physics

Object Move Press the **Move** button to activate the tool. The Move tool remains active until another tool is selected. Left-click and drag to move the selected object along the X and Y axes. Right-click and drag to move the object along the Z axis. Movement occurs in the current coordinate system (**World/Object/Screen**), which you can set in the Workspace View title bar.

Object Rotate Press the **Rotate** button to activate the tool. Left-click and drag to rotate the selected object in the Workspace about X and Y axes, Right-click and drag to rotate it around Z axis. Rotation is performed in the current coordinate system.

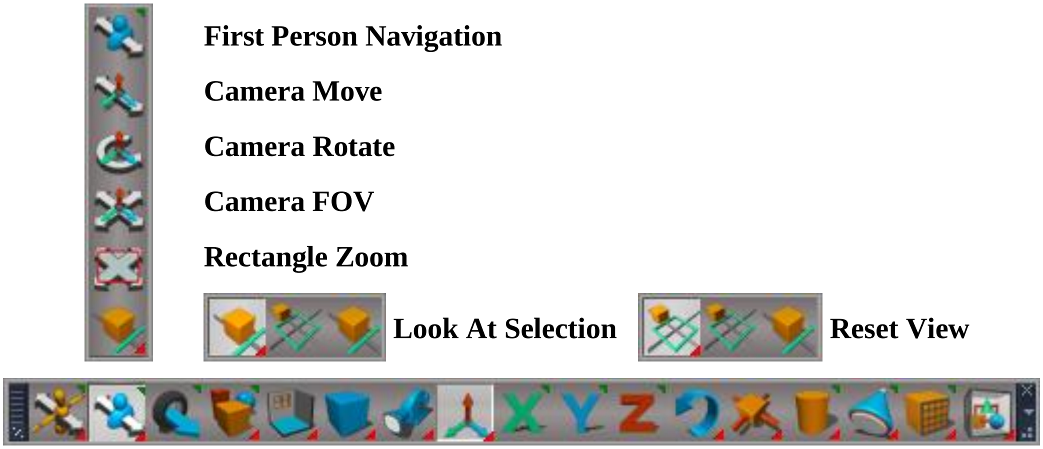
Object Scale Press the **Scale** button to activate the tool. Left-click and drag to scale the selected object along the X and Y axes . Right-click and drag to scale the object along the Z axis. Scaling is performed in the current coordinate system.

Object Look At

This command is used to make an object, camera, or light continually realign itself to remain pointed at another object. The program accomplishes this by automatically pointing the original object's Z axis at the second object's axis location (see the Axes tool). To use **Look At**, select the object that is to be constrained, select **Look At**, then select the target object. When an object is under **Look At**'s influence, it is constrained and cannot be rotated manually.

2.7.2 View Navigation Tools

The Workspace view camera can be manipulated using either the view navigation widget or the **Move**, **Encircle**, and **Zoom** view navigation tool icons on the navigation toolbar.



First Person Navigation

Camera Move

Camera Rotate

Camera FOV

Rectangle Zoom

Look At Selection

Reset View

First Person Navigation Press the button to activate a first person navigation mode, use the predefined WASD keyboard keys to navigate the Workspace. A Right-click on the tool displays a panel in the stack where settings can be adjusted for using a different set of Keys to use for the navigation movements, speed and mouse sensitivity. When the tool is active a Right -click or ESC key will toggle the mode on and off to allow for selecting objects to edit or manipulate. To quit the tool use the ESC key twice or Left click the Object arrow to quit FPN mode.

Camera Move Press the Camera Move button to activate the Camera/Eye Move tool. Left-click and drag to move the camera along the X and Y axes. Right-click and drag to move the camera along the Z axis

Camera Rotate Press the Camera Rotate button to activate Camera/Eye Rotate tool. Left-click and drag to rotate the camera around the selected object in X and Y axes. Right-click and drag to rotate the camera around the Z axis in the Screen coordinate system.

Camera FOV Press the Camera FOV button to activate the tool. Left-click and drag to change the field of view of the camera.

Rectangle Zoom Press the Rectangle Zoom button to activate the tool.

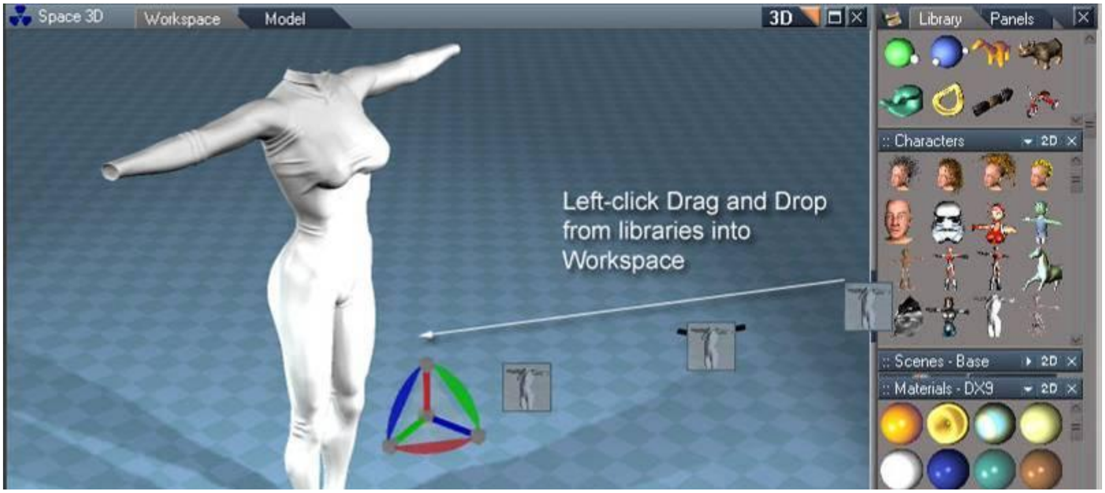
Zoom In : Hold left mouse down and Dragging the mouse left to right zooms in to the area specified and is marked by a rectangle framing the selected area.
Zoom Out : Hold left mouse down and Dragging right to left zooms out. when zooming out, the tool zooms out to the specified area where the current view is to be placed and marked by a rectangle framing the selected area. Exit the tool by a r-click in an empty part of the Workspace window.

Look At Selection Left-Click on the tool places the active Workspace view so the current selection is in the center of the viewing area.

Reset View Left-Click on the tool places the Workspace view so the current selection is in the center of the viewing area.

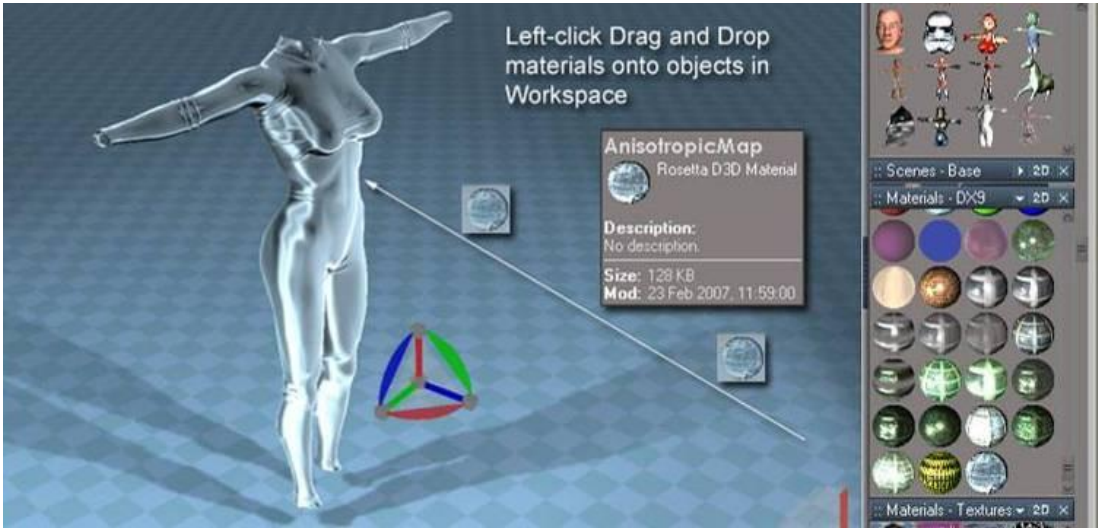
2.7.3 Workspace View Drag-and-Drop

You can add new objects to a space by dragging them from a Library directly into the Workspace view. An example of this is shown in the image below.



Dragging an object from a library into the Workspace view

You can also drag and drop materials onto objects in the Workspace view. Simply drag the material from a Library and drop it onto the target object in the Workspace view. This will automatically apply the material, replacing any existing material.



Left-click Drag and Drop materials onto objects in Workspace

AnisotropicMap
Rosetta D3D Material

Description:
No description.

Size: 128 KB
Mod: 23 Feb 2007, 11:59:00

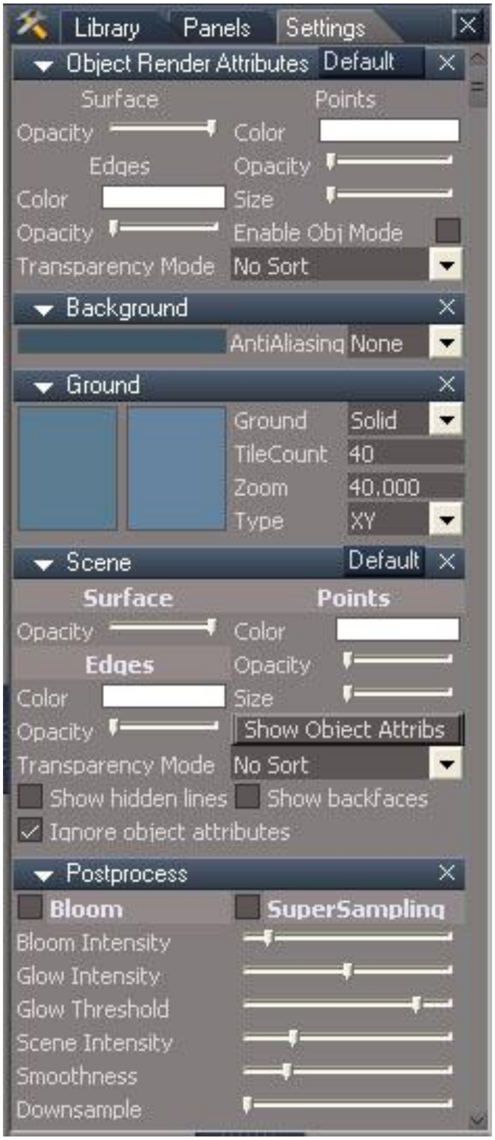
Dragging Anisotropic Map Material onto an object in the Workspace view

2.7.4 Render Preferences

Change the appearance of objects in the Workspace view by altering the Workspace render attributes (Scene at right). Change how the current surface, edges, vertices/points, and other features are rendered in the Workspace. The transparency/opacity of each component is controlled by a slider. The Scene setting is global for workspace.

- Surface Opacity
- Edges
    - Color: dbl-click for color
    - Opacity
- Points
    - Color
    - Opacity
    - Size
- Show Object Attributes
- Transparency Mode
- Show hidden lines
- Show backfaces
- Ignore object attributes

Change rendering attributes for the whole view or for each individual object. The settings for individual objects are much the same as for the global/scene settings. When changing appearance on an object-by-object basis, be sure to check *Enable Obj Mode* near the bottom-right of the Object Render Attributes panel (top=right).



Object Render Attributes: Default
Surface: Opacity
Edges: Color, Opacity
Points: Color, Opacity, Size
Enable Obj Mode □
Transparency Mode: No Sort

Background
AntiAliasing: None

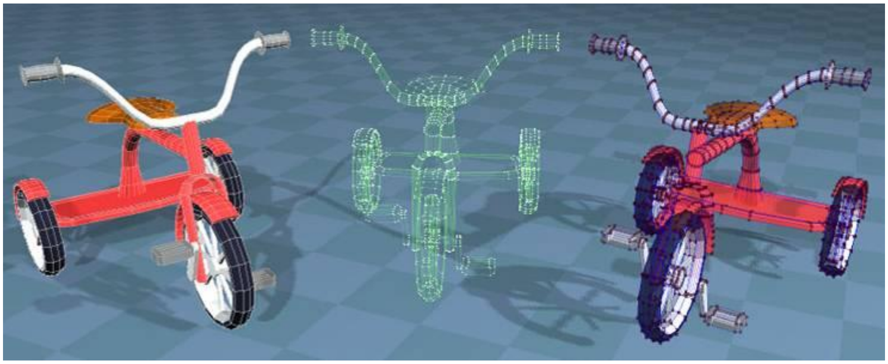
Ground
Ground: Solid
TileCount: 40
Zoom: 40.000
Type: XY

Scene: Default
Surface: Opacity
Edges: Color, Opacity
Points: Color, Opacity, Size
Show Object Attribs
Transparency Mode: No Sort
□ Show hidden lines
□ Show backfaces
☑ Ignore object attributes

Postprocess
□ Bloom
□ SuperSampling
Bloom Intensity
Glow Intensity
Glow Threshold
Scene Intensity
Smoothness
Downsample

Render Preferences with Show Object Attribs enabled to reveal the per-object options

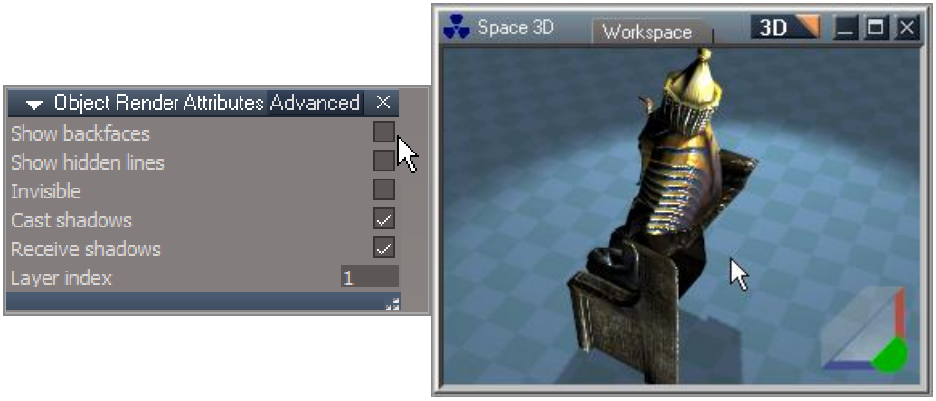
The image below shows the results of different combinations of object render preferences.



Solid surface with translucent edges and 2 transparent surfaces showing both transparent edges and points

If unchecked the *Show Hidden Lines* (under the Advanced aspect), then all vertices (and edges for wireframe mode) that would not be visible in the solid rendering mode will be hidden, regardless of object transparency settings.

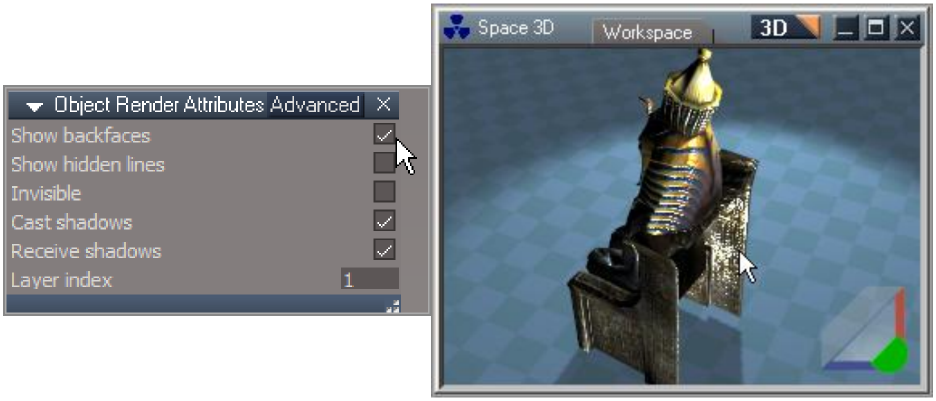
If two-sided rendering mode is enabled by clicking on *Show Backfaces* (under advanced options), then the back sides of single-sided faces will be properly lit and rendered. The following images show the Pharaoh object with and without two-sided rendering mode enabled.



Object Render Attributes Advanced

Show backfaces □
Show hidden lines □
Invisible □
Cast shadows ☑
Receive shadows ☑
Layer index 1

Show backfaces unchecked


Object Render Attributes Advanced

Show backfaces ☑
Show hidden lines □
Invisible □
Cast shadows ☑
Receive shadows ☑
Layer index 1

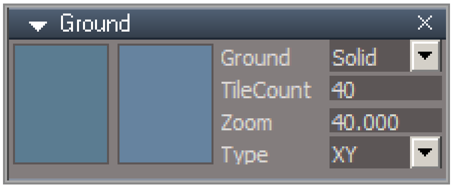
Show backfaces checked

Please note the following performance tips:

- Two-sided rendering mode can cause rendering to be up to two times slower due to the increased number of rendered faces. You should only use it when you specifically require it for a particular object (like a plane or a mesh with holes).
- Wireframe and point modes also have a negative impact on performance because the object is treated as transparent.

2.7.5 Ground Preferences

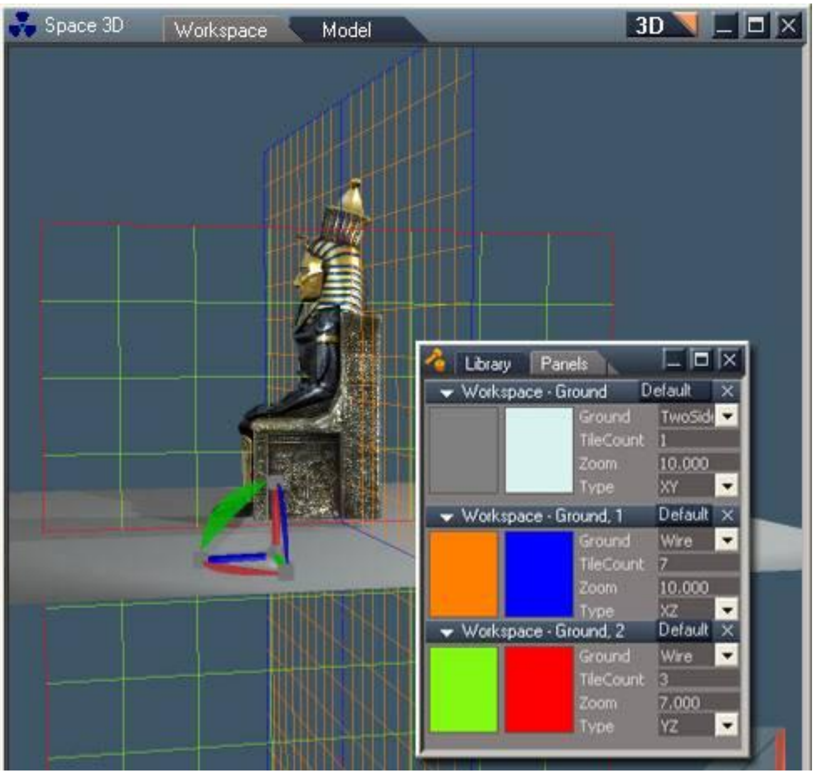
Ground (grid) settings can be changed for each workspace window on the Settings panel. The colors for the grid and number of tiles per patch (*TilesCount*) can be changed. The Grid patch always cover one unit square so using *Zoom* parameter it can be scaled appropriately.


Ground

Ground Solid
TileCount 40
Zoom 40.000
Type XY

Workspace Ground settings

In some scenes (like our Studio for example) you want to disable rendering of the grid for each view or override your own global grid settings, no matter what display layout you use. It is in fact very easy. Just place your own grid object (available in the "DX – View, Components" library as "Scene Ground") into the scene and this grid will override any per-view settings. When the grid is active, it is displayed in preferences as "Workspace –Ground".



Workspace - Ground Default
Ground TwoSid
TileCount 1
Zoom 10.000
Type XY

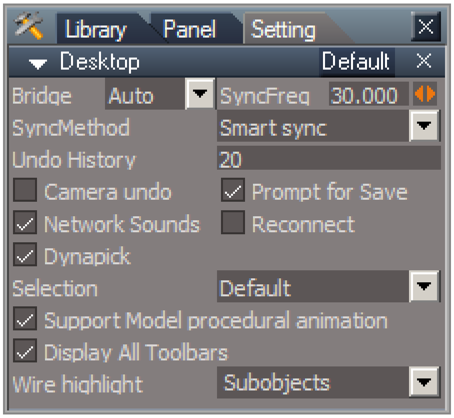
Workspace - Ground, 1 Default
Ground Wire
TileCount 7
Zoom 10.000
Type XZ

Workspace - Ground, 2 Default
Ground Wire
TileCount 3
Zoom 7.000
Type YZ

Three grid objects with various size and color settings

2.7.6 Desktop Preferences

Desktop preferences apply to entire desktop not just to workspace, but we decided to describe them here.



Bridge: Auto
SyncFreq: 30.000
SyncMethod: Smart sync
Undo History: 20
□ Camera undo
☑ Prompt for Save
☑ Network Sounds
□ Reconnect
☑ Dynapick
Selection: Default
☑ Support Model procedural animation
☑ Display All Toolbars
Wire highlight: Subobjects

Desktop Default

Default Aspect:

- BridgeSync: Sets the rate of synchronization between Workspace and Model Views. 30 fps is the default, and you can set the rate slower if you experience a slow down in program response due to the amount of data going between Model View and Workspace View. 0 fps means the synchronization will be executed only in selected cases such as when selection changes. Moving objects will not be synchronized in real-time.

- Bridge Switch: This combo box allows you to turn the bridge on or off. There are three options:
  - On: Bridge is deactivated *only when modeler is closed.* Use this mode if:
    - You do lot of animations with behaviors
    - Switch often among layouts
    - Want to be sure you won't lose any information during switches

  - Off: Bridge is *always* deactivated. Use this mode if:
    - You work with large models, or you convert large models from older trueSpace versions. This is the most memory efficient and fastest mode.
    - You use only the functions available in trueSpace6.6 for quick modeling and KF animation

  - Auto: Bridge is deactivated *when Modeler is closed* or *Modeler is open and all Link Editor Views and View3Ds are closed.* This is a Default mode. Use this mode if:
    - You want to use Model and Workspace and your objects do not use a lot of behaviors combined with key-frames animation which could get lost during bridge activation/deactivation. This is the best compromise for most of the

workflows.

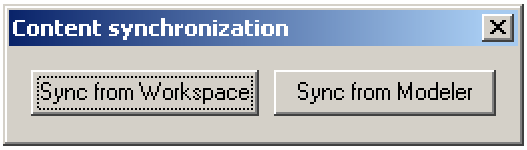
To summarize the philosophy behind using the bridge modes:

Bridge On is a safe mode. It uses more memory and it is slower due to always on sync but if you have a medium size project where it is critical not to lose objects on both side of the bridge this is the mode you would use.

Bridge Off is for large projects which do not require much sync between Model and Workspace.

Bridge Auto is for most everyday work and is the Default mode. It is both fast and memory efficient but in some cases you may lose work on either Model or Workspace side.

When you turn the bridge to On or to Auto after you have worked in Off state, you will be presented with a dialogue box which will allow you to specify the direction of synchronization for Model and Workspace sides:



Sync is smart and will preserve all the work on the destination side that is not in direct conflict with the source side. You may lose some work, however, if you create some Workspace script behavior on a grouped object and then execute Boolean operation on the subobjects of the group in Model side. If there is a conflict, the Source side overwrites destination changes.

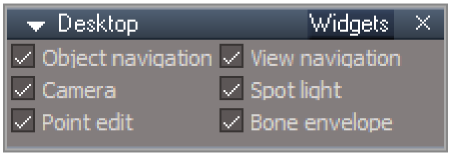
- **SyncMethod:** Called when the bridge is turned On to synchronize content on Workspace and Model side of the bridge.
  - Smart Sync: Source side overrides the destination side if there are conflicting changes on the same object. If changes are on different objects, destination changes are kept.
  - Full Merge: More comprehensive but slower method which attempts to preserve even changes on the same object.

- Support Model procedural animation: Allows you to turn off transfer of procedural animation to model side to increase the speed of bridge synchronization. If you want to see Workspace scripts running in Model you set this checkbox to On. That way you can move for example object in Model View and see color is procedurally changed by Workspace script.

- Dynapick: Allows you to select and move objects with a single click and drag. This applies to the Workspace View only, but there is a similar option in Model Preferences. Make sure that you also select Move Object Icon on Workspace toolbar otherwise Dynapick may not work reliably.

- Camera Undo: Allows you to exclude camera navigation from the Undo stack.

- Undo/Redo History depth: Allows you to set number of Undo steps remembered by program. A larger number is better, but it uses more memory.

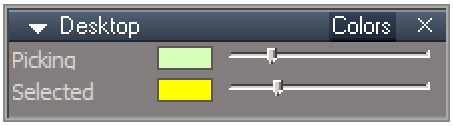


☑ Object navigation ☑ View navigation
☑ Camera ☑ Spot light
☑ Point edit ☑ Bone envelope

Desktop Widgets

Widgets Aspect.

- Object Navigation: Turn on and Off Object Widget.
- View Navigation: Turn on and Off View Widget.
- Camera: Turn on and Off Camera Widget.
- Spotlight: Turn on and Off Spotlight Widget.
- Point Edit: Turn on and Off Point Edit Widget.
- Bone Envelope: Turn on and Off Bone Envelope Widget.



Picking 🟩
Selected 🟨

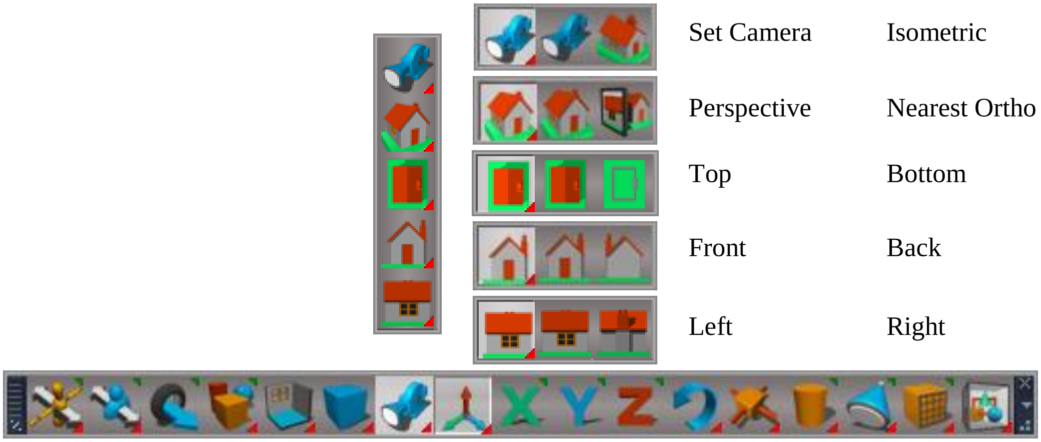
Desktop Colors

Colors Aspect.

- Picking: Double click to Set picked object color.
- Selected: Double click to Set selected object color.

2.7.7 Perspective and Orthogonal Views

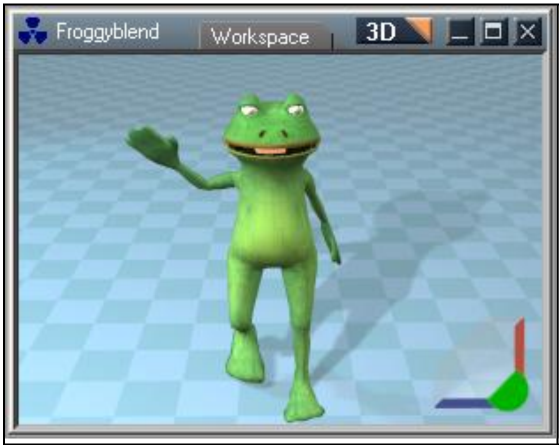
You can switch the view type from perspective projection to special orthogonal projections like top-view and front-view by clicking on the appropriate icons on the Workspace tool bar.



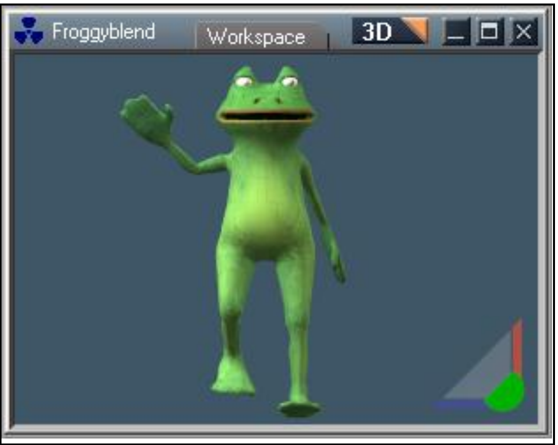
Switching the view to a different type

Perspective View Nearest Ortho

The Perspective view is the default view and is the most flexible. In this view you can move the point of view in any direction on any axis. The Nearest Ortho will determine which orthogonal view is closest to the current view. It is that orthogonal view which will appear in the selected window.



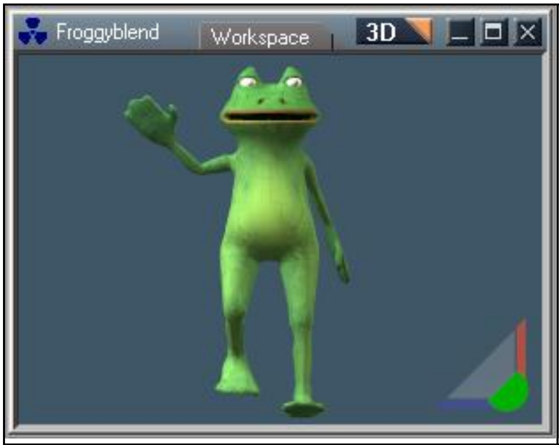
Perspective View



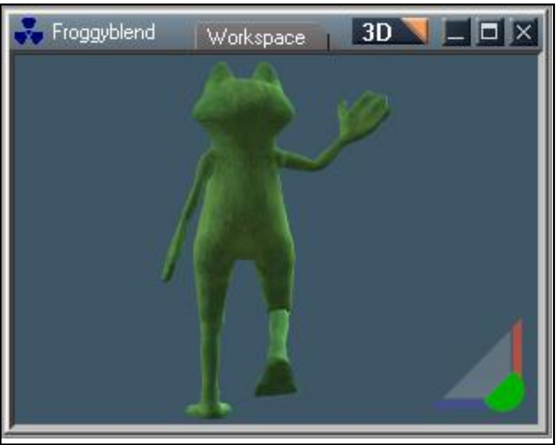
Nearest Ortho

Front View Back View

The front view is orthogonal (no perspective), looking down the world Y axis. Back View is the opposite of the front view and looks from the -Y direction. The view can be moved and scaled (zoomed) but not rotated.



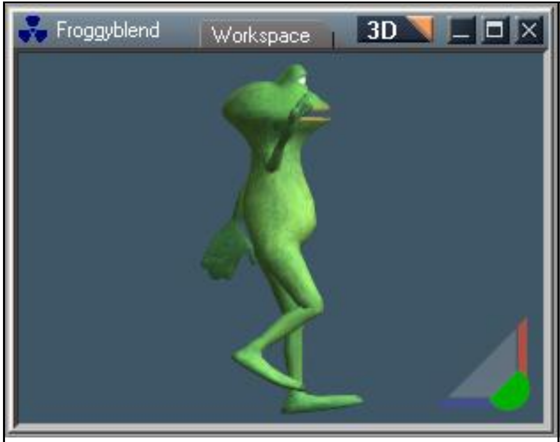
Front View



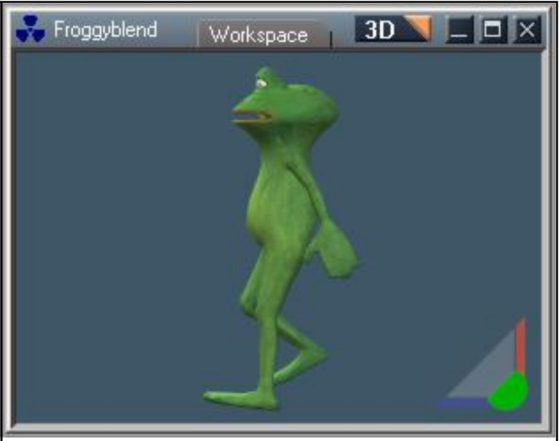
Back View

Left View Right View

The left view is orthogonal, looking down the world X axis. Opposite is the right view, which looks from the -X direction. The view can be moved and scaled but not rotated.



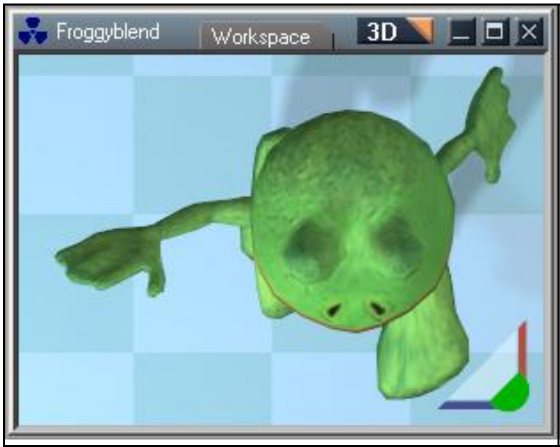
Left View



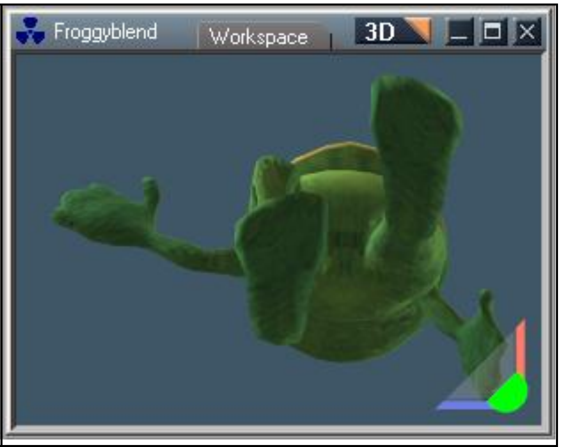
Right View

Top View and Bottom View

The top view is orthogonal, looking down the world Z axis. Opposite is the Bottom view, which is looking up the Z axis.



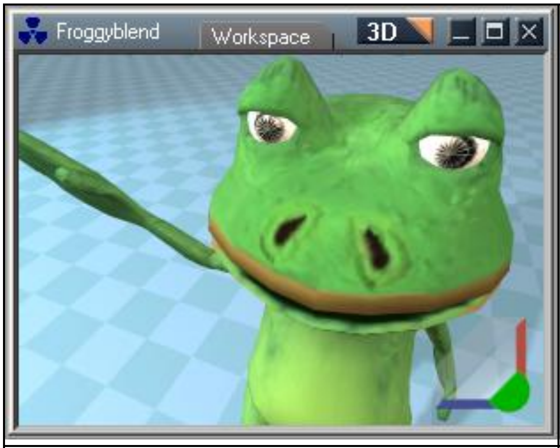
Top View



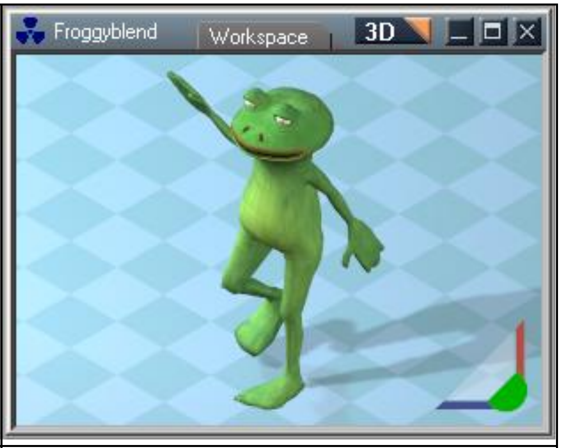
Bottom View

Set Camera Isometric

Set Camera view sets the view from the currently selected camera. The orientation of the view is in regard to the selected object's Z axis. This view can be moved, rotated, and scaled, either with the navigation tools or by manipulating the selected camera object. The Isometric view will set the view to a standard isometric view. Use of view rotation is disabled when in Isometric view.



Set Camera

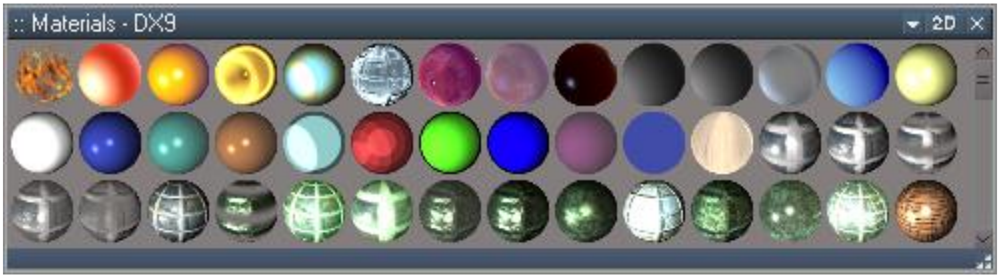


Isometric

You can set the active camera for the view by selecting the camera you wish to use and then clicking on the **Set Active Camera** tool on the Workspace view title bar.

2.7.8 DirectX9 Materials

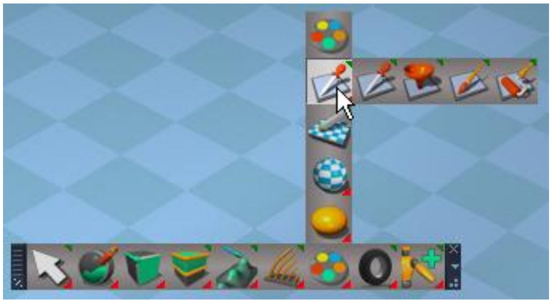
The trueSpace Workspace view provides support for Microsoft DirectX 9, enabling advanced features such as **normal mapping**, **per-pixel rendering**, and **vertex shaders**. You will find some DirectX materials under the Shaders panel of the Materials library.



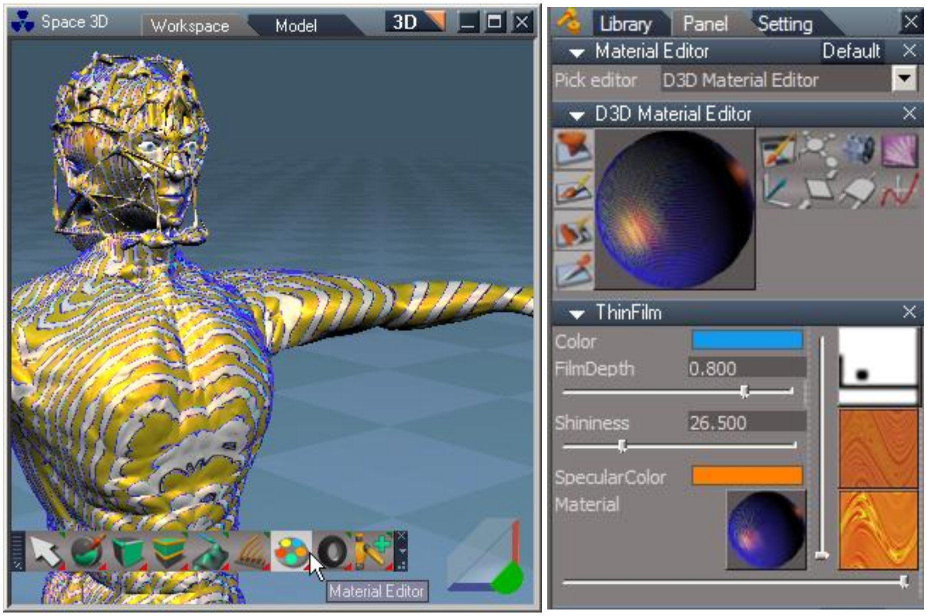
You can inspect the materials of the object by clicking on the material inspect tool on the Workspace toolbar.

Material Inspect

While the tool is enabled, click on objects to show their materials in the stack view, where you can easily change the material parameters. Clicking on the icon again disables the tool.



Location of Material Editor tools in Workspace toolbar.



Pick editor: D3D Material Editor

Color

FilmDepth: 0.800

Shininess: 26.500

SpecularColor

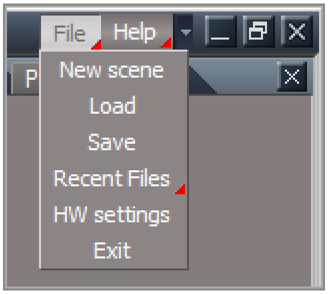
Material

Editing a material using the Inspect material tool

- Reference: Chapter 5 D3D Material Editor

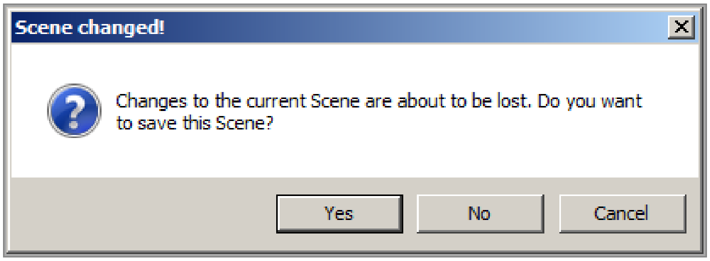
2.7.9 Workspace File and Help Menu

Although most of the actions in trueSpace can be accomplished through icons, there are two menus available: File and Help. The File menu has options for loading and saving objects, as well as access to the Workspace Hardware Settings.



New scene

This removes all objects from your Workspace and starts you off with a fresh scene. It will first prompt you to save your work by pressing no and saving it to a library or to continue and clear the scene.



Scene changed!

Changes to the current Scene are about to be lost. Do you want to save this Scene?

Load

Allows for the loading of tS7 supported object types. Opening an object adds it to your scene. The drop down list in the windows dialogue to shows the supported types.

Save

Will allow for the saving of tS7 objects to a directory. The drop down list in the windows dialogue shows the supported types. If you want to save a particular object then you will need to select it in the LE first.

Recent Files

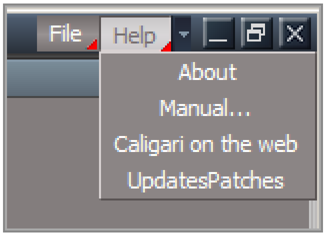
This provides a list of the most recent files loaded through the menu for quick access.

HW settings

Gives access to the Workspace hardware settings.

Exit

This will exit the program. trueSpace will automatically save your current project and the scene for the Workspace. If the bridge is on then the scene will load to the Model window as well when you next launch trueSpace.



Help: About

This displays the trueSpace splash screen with the programming credits, the version information, and the display mode in which trueSpace is running.

Help: Manual

This will take you to the contents page of the PDF version of the manual you are reading now.

Help: Caligari on the web

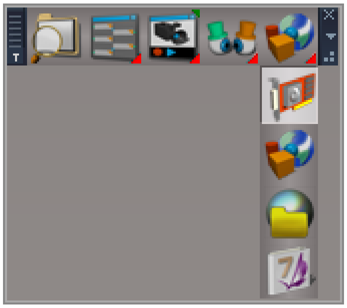
This will take you to the Caligari web site where you will find all the latest information about trueSpace, including plugins, forums, and user galleries.

Help: Updates and Patches

Look to this area for updates as they become available.

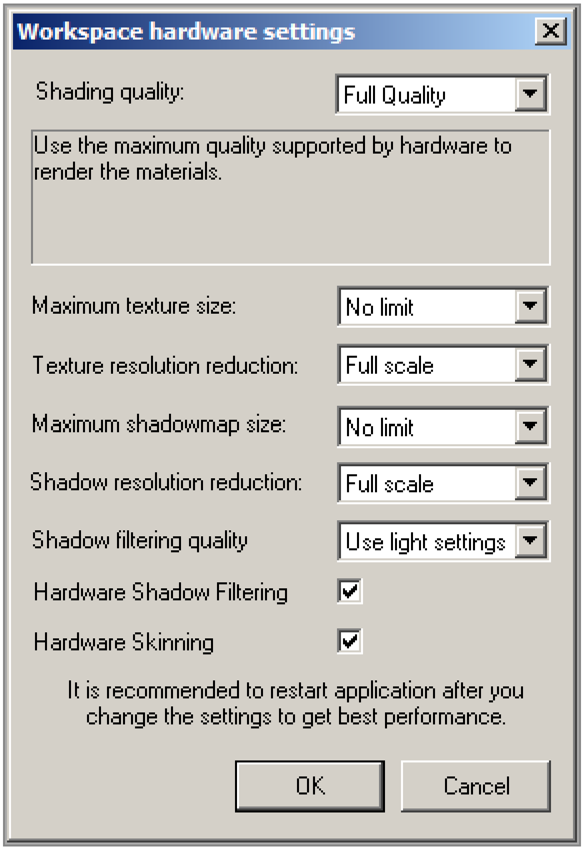
Workspace Hardware Settings

You can enable Workspace hardware settings panel by clicking on the icon on the Workspace tool bar as well.



Workspace hardware settings

Here you can control how your graphics hardware handles rendering the real-time view. This lets you achieve the balance of quality versus performance that best suits your hardware.



Workspace hardware settings

Shading quality: Full Quality

Use the maximum quality supported by hardware to render the materials.

Maximum texture size: No limit

Texture resolution reduction: Full scale

Maximum shadowmap size: No limit

Shadow resolution reduction: Full scale

Shadow filtering quality Use light settings

Hardware Shadow Filtering ☑

Hardware Skinning ☑

It is recommended to restart application after you change the settings to get best performance.

Workspace hardware settings Panel

- **Shading Quality**– You can set trueSpace to display using particular shader versions. Lower versions give better performance but less visual quality. This may be necessary to maintain good performance on graphics cards that are more than 2 or 3 years old. “Full Quality” will use the best pixel shader version that your hardware can support – remember that while your hardware may support a particular version, switching to an earlier version will still give you better performance.
- **Maximum Texture Size** – Overrides any textures in the scene, and specifies the maximum size to be sent to your graphics card. This can help on graphics cards with lower amounts of memory (less than 256Mb), but will give a more blurred look to any textures in the scene.
- **Maximum Shadow Map Size** – This will override any Shadow Map sizes set for shadow casting lights in the scene. If a scene has been created for higher quality, you can quickly reduce the demands on your hardware using this setting, without having to find and edit the light sources yourself.

- **Shadow Resolution Reduction** – This reduces the overall shadow quality for all lights in the scene, giving better performance but lower quality and broader, more blurry shadows.
- **Shadow Filtering Quality** – Overrides the setting in the individual lights, and lets you specify what settings to use. Generally you use this to lower the filtering quality to give better performance, but you could use it to force Poisson disk shadows for all lights in the scene even if the scene was not set up to use them.
- **Hardware Shadow Filtering** – Enables Hardware Shadow Filtering (discussed in section 7.1.4, Shadowing Improvements). You may need to disable this manually if you experience poor performance, or if your hardware / driver does not support this feature and this has not been properly detected by trueSpace.
- **Hardware Skinning** – This uses the GPU to perform calculations relation to skinned objects (objects that are controlled and animated by skeletons). Enabling this will reduce the load on the CPU, and give better performance with newer graphics card hardware. The effect will be most noticeable on either complex models, or where there are many models in the scene being animated by skeletons. Note that it has no effect on an object that is currently selected – selected objects must be processed by the CPU, so be sure to deselect any models animated by skeletons for best real-time animation playback.

The Hardware settings allows you to increase the speed of rendering by lowering the quality of the output. *Shading Quality* allows you to pick a different Pixel Shader level. For example, GeForce FX cards allow you to render with Full Quality mode, but perform better with Pixel Shaders 1.1. You can limit the size of textures to increase the speed and reduce memory consumption by changing *Maximum Texture Size* and *Texture Resolution Reduction* properties. Similarly, you can tweak shadow map quality and speed. You can disable shadows by selecting “No Shadows” in the *Maximum Shadowmap Size* property. Disabling shadows can greatly increase rendering performance. *Shadow Filtering Quality* is another option which can help you to increase the performance of rendering. Choose “No Filtering” to disable smooth edges of shadows for all lights. Or, choose another filtering type to increase the quality of shadow edges.

Workspace hardware settings are stored in the *D3DView.Settings.xml* file in the *Scripts* directory so that you can easily backup your tweaked file or share it with your friends.

General tips for improving performance include:

- Try to lower the shadow filtering quality or shadow map size.
- Try to disable shadow rendering at all.
- Try to lower the resolution of textures.

- Try to lower the shading quality level (recommended for owners of GeForce FX cards).
- ➢ Reference: section 7.1.4, Shadowing Improvements

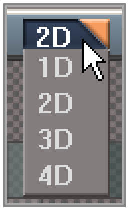
2.8 2D View – Link Editor

2.8.1 Link Editor Description

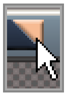
To edit any object in trueSpace requires that you must first enter it. This is true whether you are viewing an object in 1D, 2D, or 3D aspect. Here in the Link Editor, which is the 2D area of trueSpace, you will eventually enter and work with objects. The Link Editor provides you with a toolset to browse, assemble, analyze and modify the structure of objects.

Name

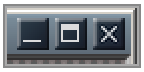
View Tabs



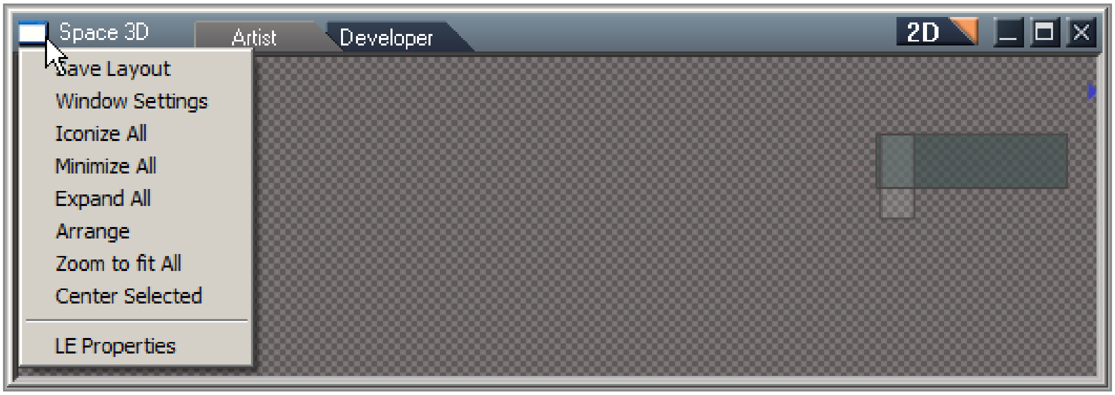
Windows Aspect



Exit



Min/Max/Close



Link Editor settings

On the Link Editor title-bar, from left to right are:

- Dropdown list of commands:
  - Iconize All: this will change all objects in the Link Editor to a small icon (Preserves most space).
  - Minimize All: minimize all object to a title-bar: maximize space with name of object visible.
  - Expand All: expands all objects to their full panel size.

    - Zoom to fit All: zoom the Link Editor so all objects are visible. Makes it easy to find an object.
    - Center Selected: centers the Link Editor view on the selected object.
    - LE Properties: changes the stack view to Preferences and loads the Link Editor settings for modification.
- Name: the name of the current space or object.
- View Tabs:
    - Artist: hides some elements considered unnecessary for creation process.
    - Developer: adds back these elements as they are necessary for development.
- Window Aspect: ability to change the window to desired aspect.
- Window:
    - Minimize
    - Maximize
    - Close

2.8.2 Main Elements of the Link Editor

There are several tools available for navigating in the Link Editor view.

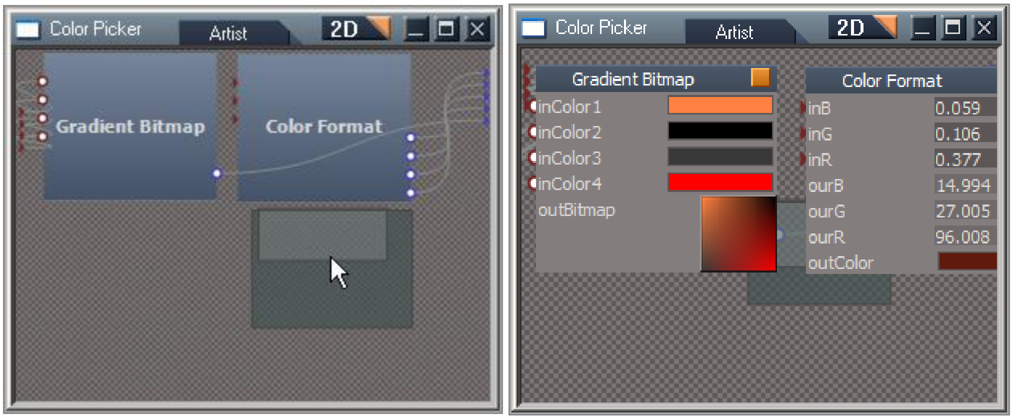
Navigation Control

The **Navigation Control** consists of two rectangles and is located in the lower right-hand corner of the Link Editor view. The gray rectangle represents the entire contents of the object you have entered. The position of this rectangle is fixed. The cyan rectangle represents your view relative to the contents of the object. You can move this view rectangle to encompass different portions of the 2D space.

The navigation control also provides the ability to reset the view, zoom, and perform other navigation tasks as described in the sections below.

Reset View

Right-click the navigation control to reset the view. The Link Editor view will zoom and move so that you see the entire contents of the current space.



Gradient Bitmap

inColor1
inColor2
inColor3
inColor4
outBitmap

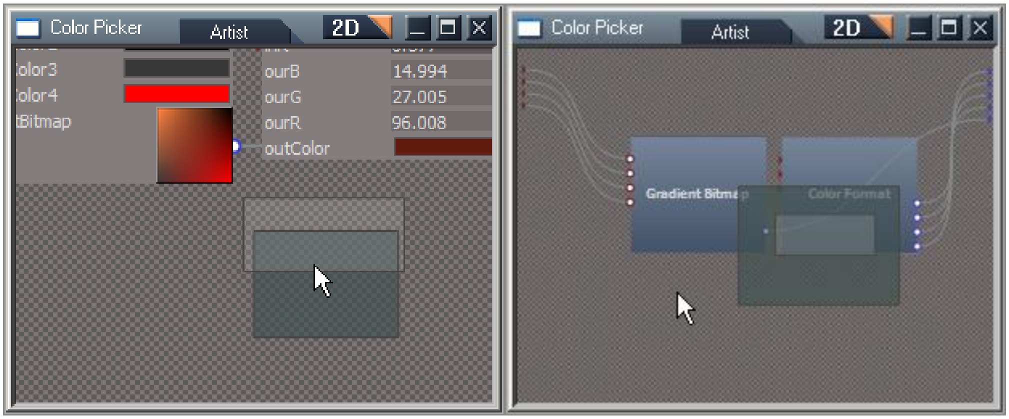
Color Format

inB	0.059
inG	0.106
inR	0.377
ourB	14.994
ourG	27.005
ourR	96.008
outColor	

Reset the Link Editor view by right-clicking the navigation control

Zoom View

Right-click and drag on the navigation control to zoom the view in and out. In zoomed view object controls are not displayed, but you can get a better view of the structure of the space.



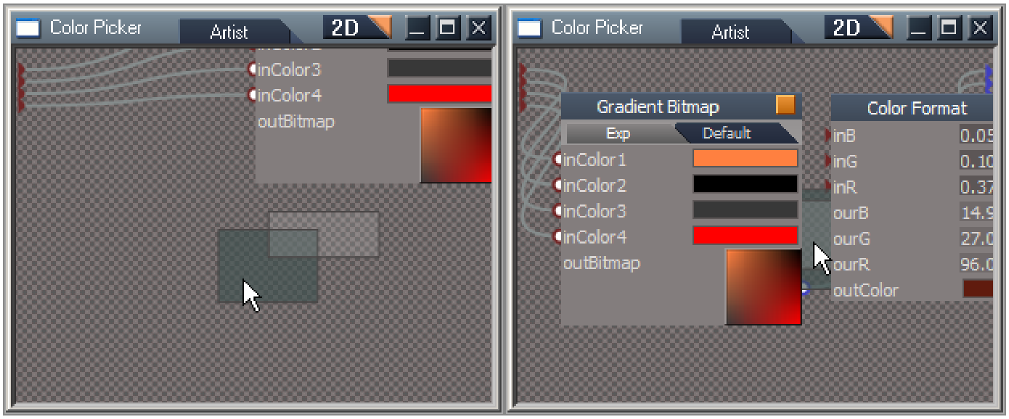
olor3
olor4
tBitmap

ourB	14.994
ourG	27.005
ourR	96.008
outColor	

Zoom the Link Editor view (out and in) by right-clicking and dragging navigation control

Move View

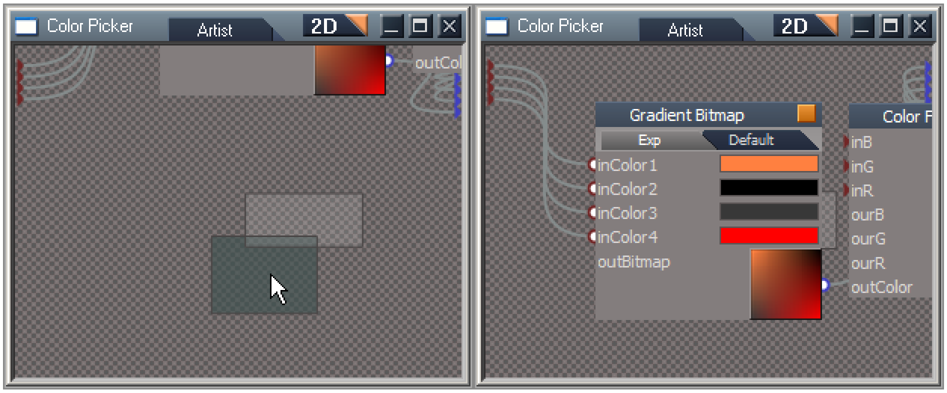
Left-click and drag on the navigation control to move the view location in any direction.



Left-click and drag on the navigation control to move the Link Editor view

Center View on Selected Object

Left-click on the navigation control to center the camera on the selected object. If more than one object is selected then the view will move to the center of all selected objects.

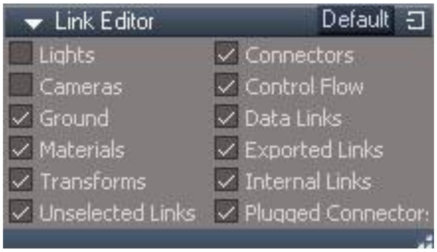


Left-click once on the navigation control to center the view on the currently selected objects

Link Editor View Filters

As you begin to build and work with more complex objects with many attributes and links, you may find your Link Editor view becoming more and more visually confusing, with a “spaghetti-like” mass of links making it difficult examine the structure of your object.

View filters can help reduce object and link clutter in Link Editor view. The filtering tools are accessible from the **Link Editor Preferences Panel**. You can filter objects by type, showing or hiding transformation and material nodes, and you can filter various link and connector types, allowing you to hide items like control flow links and unconnected connectors.

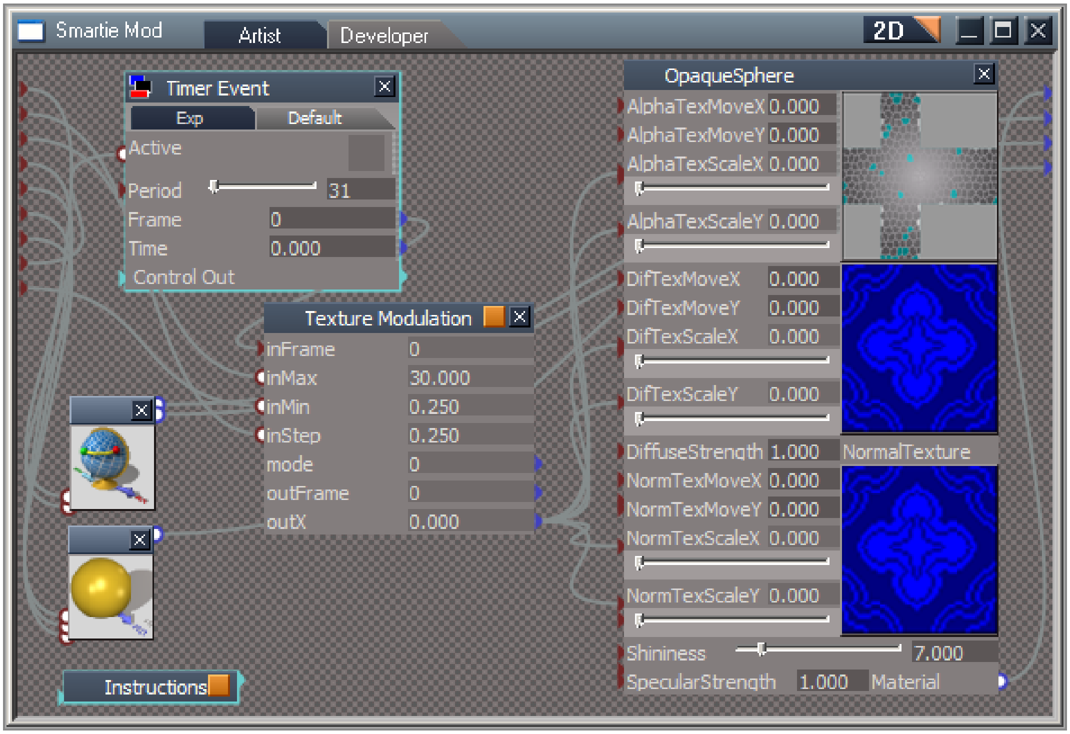


Link Editor Default

☐ Lights ☑ Connectors
☐ Cameras ☑ Control Flow
☑ Ground ☑ Data Links
☑ Materials ☑ Exported Links
☑ Transforms ☑ Internal Links
☑ Unselected Links ☑ Plugged Connector:

Link Editor Filter Panel

The "Developer Aspect" simply shows all object, links, and connectors, as seen in the image below.



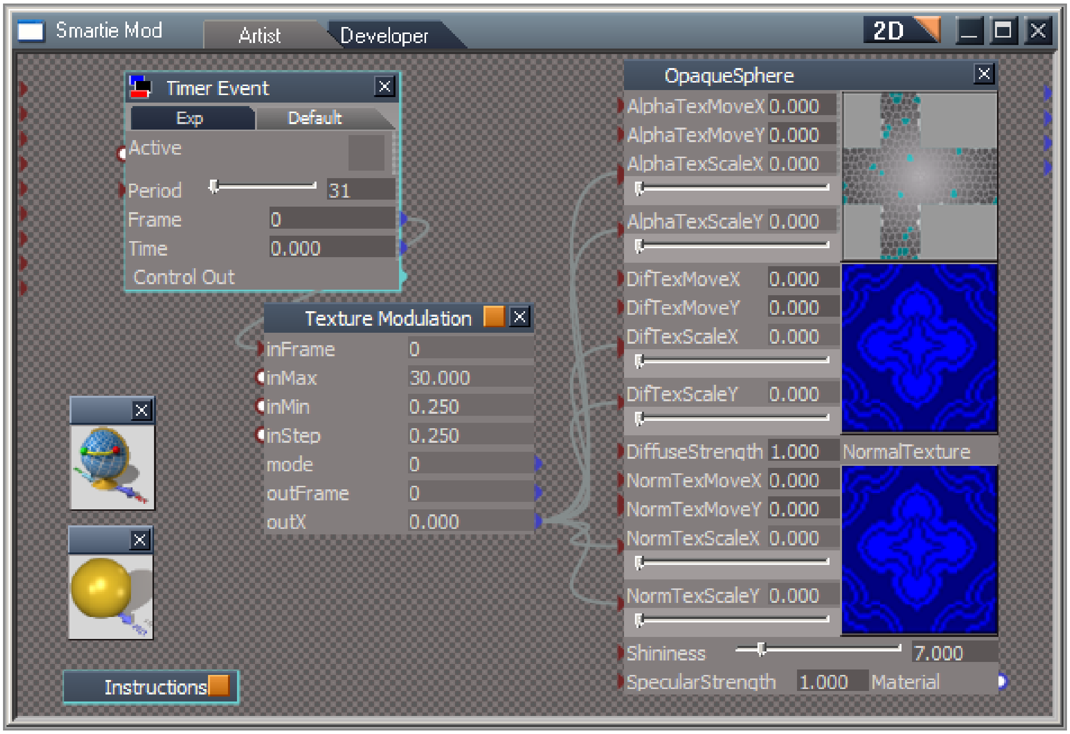
Timer Event
Exp Default
Active
Period 31
Frame 0
Time 0.000
Control Out

Texture Modulation
inFrame 0
inMax 30.000
inMin 0.250
inStep 0.250
mode 0
outFrame 0
outX 0.000

OpaqueSphere
AlphaTexMoveX 0.000
AlphaTexMoveY 0.000
AlphaTexScaleX 0.000
AlphaTexScaleY 0.000
DifTexMoveX 0.000
DifTexMoveY 0.000
DifTexScaleX 0.000
DifTexScaleY 0.000
DiffuseStrength 1.000
NormalTexture
NormTexMoveX 0.000
NormTexMoveY 0.000
NormTexScaleX 0.000
NormTexScaleY 0.000
Shininess 7.000
SpecularStrength 1.000
Material

The Developer Aspect shows all objects, links, and connectors

Showing everything may not always be a good idea because the view can quickly resemble spaghetti with only a few objects. In the image below the Developer Aspect is turned off.



Active
Period 31
Frame 0
Time 0.000
Control Out

inFrame 0
inMax 30.000
inMin 0.250
inStep 0.250
mode 0
outFrame 0
outX 0.000

AlphaTexMoveX 0.000
AlphaTexMoveY 0.000
AlphaTexScaleX 0.000
AlphaTexScaleY 0.000
DifTexMoveX 0.000
DifTexMoveY 0.000
DifTexScaleX 0.000
DifTexScaleY 0.000
DiffuseStrength 1.000
NormTexMoveX 0.000
NormTexMoveY 0.000
NormTexScaleX 0.000
NormTexScaleY 0.000
Shininess 7.000
SpecularStrength 1.000

The same objects with some links and connectors filtered in Artist aspect

By default, all exported links are filtered out (shown as orange or blue half circles), and connectors without visible controls on the object's panel are hidden.

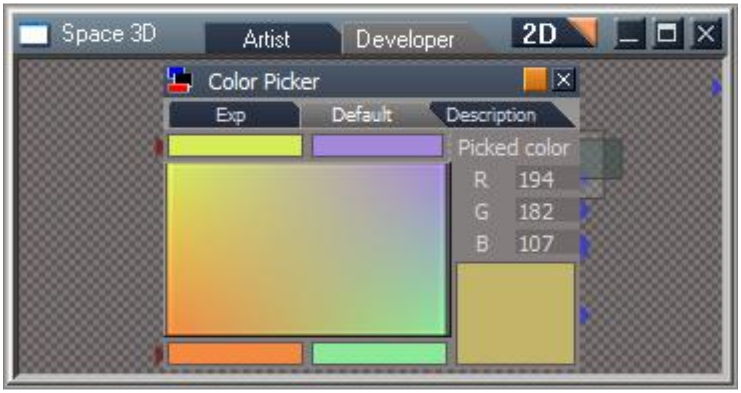
The Link Editor filter should allow you to customize how the objects and links are visible as desired. When the Link Editor becomes too cluttered, just remember to utilize the filter for better workflow.

2.8.3 Objects Description

Objects in trueSpace have a structure about them. Much of what you see in the Link Editor and an object's structure will be familiar. The object's material, size, shape and such are included in the structure (if required). There are other more profound changes or alterations to be made as well. Within the context of the Link Editor, you are creating structural modifications to the very nature of what an object is. You have the power to change how it works, how it looks and how it reacts.

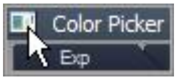
Objects vary in structure. Some objects are very simple, while other objects and creations are of a more complex structure. The trueSpace libraries are full of objects with structure you can explore. You also have the capability to create new objects using existing objects or a combination of custom and existing objects.

Inside the Link Editor, objects are represented by 2D panels. Every objects panel has a title-bar with common associated actions:

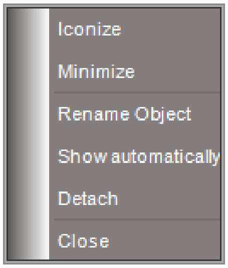


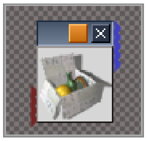
An object in the Link Editor

- A left-Click in the left hand side of the title bar gives a pop-up menu containing a list of commands for the object:

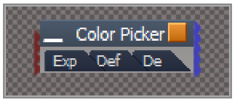


L-Click Menu

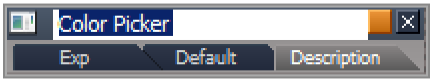




Iconize



Minimize



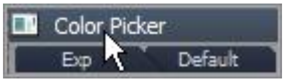
Rename

- ***Object L-click Menu***

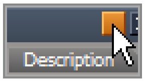
-
    - Iconize: minimize the object in the Link Editor to an icon.
    - Minimize: minimizes the object to just title-bar size.
    - Rename Object: allows you to rename as desired.
    - Show automatically: determines if an object panel is viewable to truePlace users.
    - Detach: loads the objects panel into the stack.

  - Close: will act as a delete function for the object.

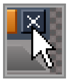
- The name of the object.
- An orange Enter Object button (if appropriate).
- A Close (Delete) button.



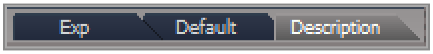
Object Name



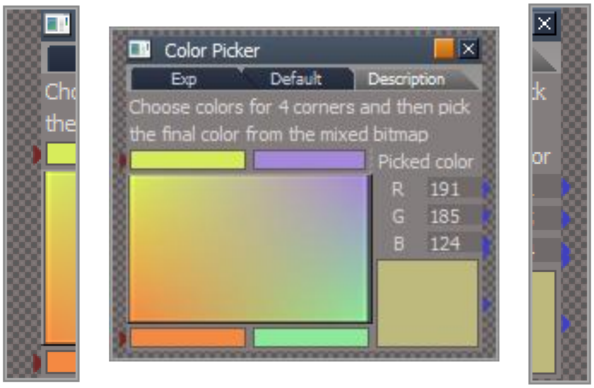
Enter Object



Close Object



Aspects

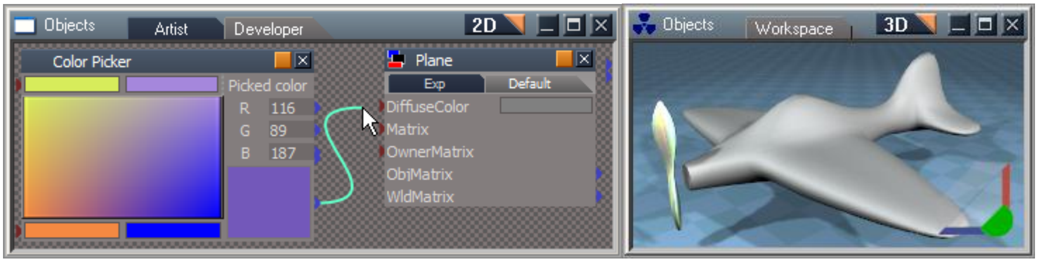


Choose colors for 4 corners and then pick the final color from the mixed bitmap

R 191
G 185
B 124

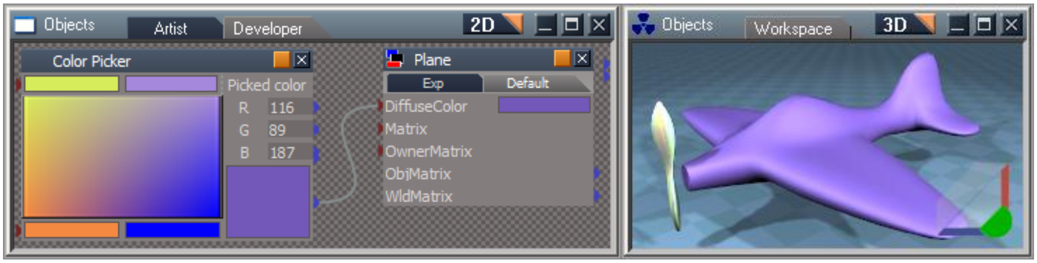
Red Triangle input connectors Blue Triangle output connectors

Located on each object’s panel are attributes. The attributes can be represented by various controls types. Attributes of one object can depend upon attributes of other objects. This relationship of dependency is represented by a link displayed as a spline curve connecting export "blue output" connectors from one object, to import "red input" connections on another object., and is created by holding the left-mouse down and then dragging from one attribute to the other which then connects the attributes of objects and passes the information along the "wire".



R 116
G 89
B 187

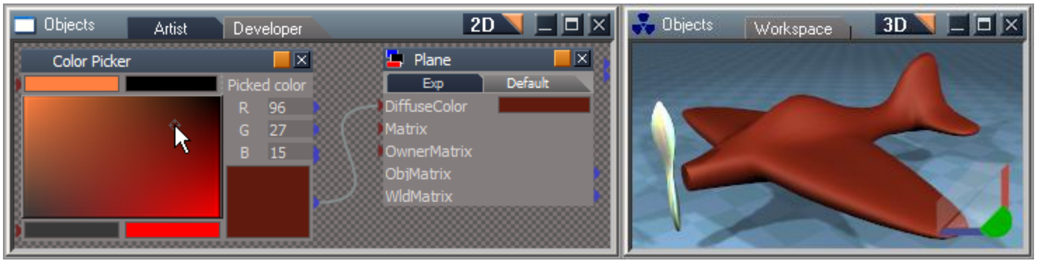
Creating a link to form a dependency



R 116
G 89
B 187

The connection is made and color is passed from one object to the other

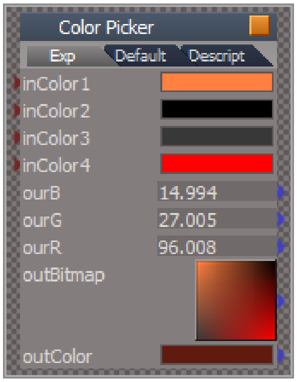
Above, a simple dependency between the Color Picker object’s output attribute “outColor” and the Plane object’s input attribute "DiffuseColor". The result of this dependency is a change in the Plane’s body color. There are other attributes on these objects, both input and output, which are not being utilized in a dependency relationship. Some of these attributes are available for modifying. For instance the Color Picker object has four color attributes, one for each corner of the main color area surrounded by these four color attributes. Double-clicking on the color attributes will bring up a Color selection dialog to allow changing the colors of the mix. The image below shows a change in these colors and the result in real-time is displayed on the body of the Plane object.



R 96
G 27
B 15

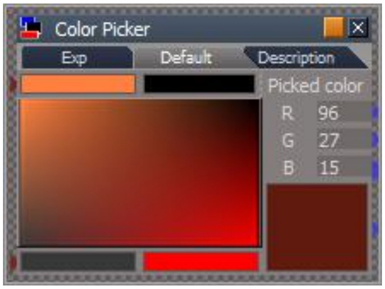
Color changes will be passed and updated to the planes exported material color in realtime through the link

With all this power and capability, objects within the Link Editor may require more than one aspect, to showcase the available attributes. For this reason, an object may have more than one aspect for you to choose. When the aspects were created for the object, or when you create custom objects, you will organize an aspect for the benefit or feature you wish to present. Below shows three aspects of the Color Picker object. Each aspect has a different layout of the attributes with a purpose in mind. By default, all objects will have an expanded and a default aspect. Additional aspects are considered custom aspects as the Description aspect demonstrates in the Color Picker object.



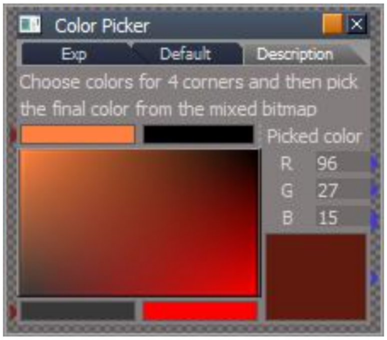
inColor1
inColor2
inColor3
inColor4
ourB 14.994
ourG 27.005
ourR 96.008
outBitmap
outColor

Expanded Aspect



Picked color
R 96
G 27
B 15

Default Aspect

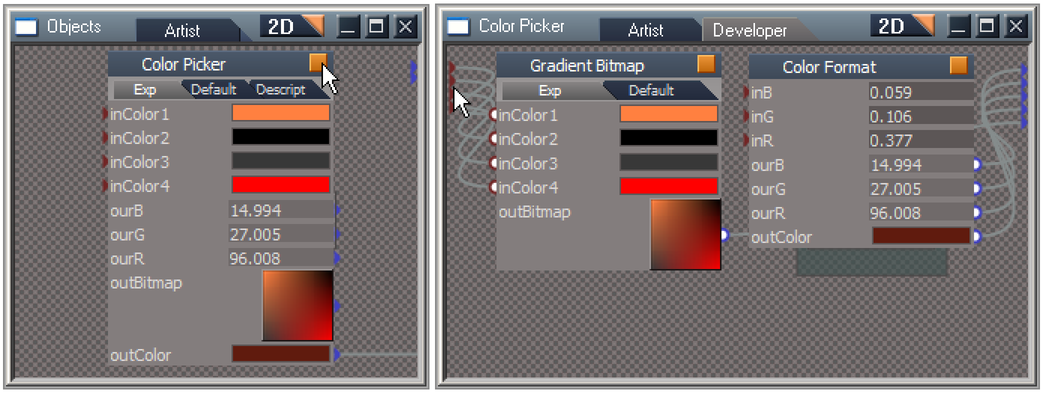


Choose colors for 4 corners and then pick the final color from the mixed bitmap

Picked color
R 96
G 27
B 15

Description

Where do these attributes come from? Take a look inside the Color Picker object for some help with this question. On the left-hand side we see the front panel of the Color Picker object. In the right-hand side we see two objects, which are encapsulated inside the Color Picker object. Some input attributes , inColor1 to in Color4 on the Gradient Bitmap object have been exported or exposed to the front panel. We also see some Output attributes which have been exported or exposed to the front panel from the Color Format object. When exposed together on the front panel of the Color Picker object, we have the ability to modify these attributes and change the outColor attribute at bottom of the Color Picker object image on the left-hand side.



inColor1
inColor2
inColor3
inColor4
ourB 14.994
ourG 27.005
ourR 96.008
outBitmap
outColor

inColor1
inColor2
inColor3
inColor4
outBitmap

inB 0.059
inG 0.106
inR 0.377
ourB 14.994
ourG 27.005
ourR 96.008
outColor

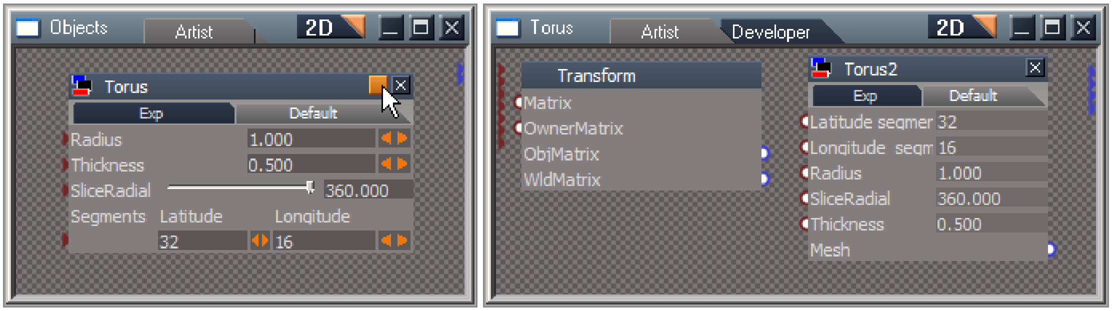
Closer look inside Color Picker object

Once you understand where attributes come from and how easy it can be to modify them, you are in a good position to begin exploring objects, aspects, attributes, links and relationships in the Link Editor. As your knowledge increases, you will be able to create custom objects and relationships as desired.

2.8.4 Creating and Editing objects in the Link Editor

An object in the Link Editor can be either a base object or a compound object:

- **Base objects** cannot be entered. They may have both attributes and a method (a procedure or a function) that can change value of the attributes, but the method is not accessible to you. The methods implemented by an object can be a constant, or a more complex function. The result of the function can depend on the input values of other attributes of the same object. An example of a complex function that depends on other attributes is the procedural Torus2 object. The illustration below shows the Torus2 object, which is located a level inside the default primitive Torus object.



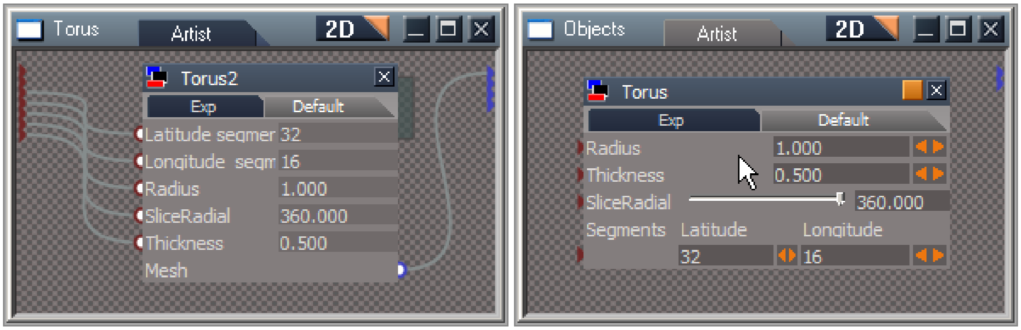
Radius 1.000
Thickness 0.500
SliceRadial 360.000
Segments Latitude 32 Longitude 16

Latitude segmer 32
Longitude segm 16
Radius 1.000
SliceRadial 360.000
Thickness 0.500

Torus2 Base object

  - ○ Based on the input attributes *Lattitude segments*, *Longitude segments*, *Radius*, SliceRadial and Thickness, the Torus2 object calculates the output attribute *Mesh*.

- **Compound objects** allow for grouping an assembly of objects within one compound object. They expose some of the attributes of their inner objects to the outside context. The objects inside of a compound object can be either base objects or compound objects, with their own internal assembly of sub-objects. Using our default primitive Torus object, we see exported attributes available on the Default Aspect of the panel as illustrated below.



Latitude segmer 32
Longitude segm 16
Radius 1.000
SliceRadial 360.000
Thickness 0.500
Mesh

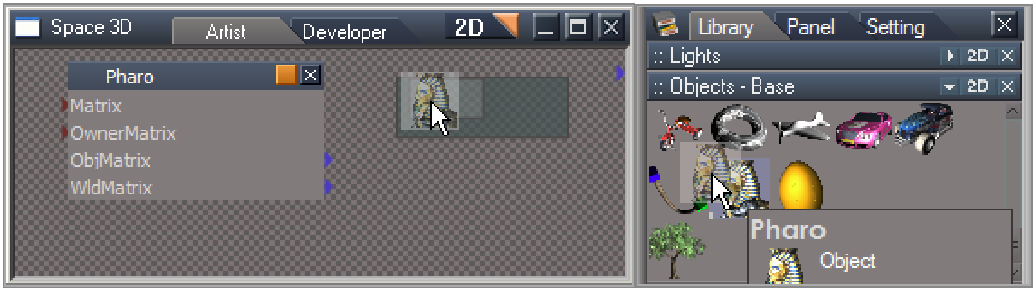
Radius 1.000
Thickness 0.500
SliceRadial 360.000
Segments Latitude 32 Longitude 16

Exposed attributes from internal objects

The Link Editor provides a set of tools for modifying the structure of objects and creating new objects. These tools can be categorized into the following groups: adding/removing objects, adding/removing attributes, and creating/deleting dependency relations among attributes of different objects.

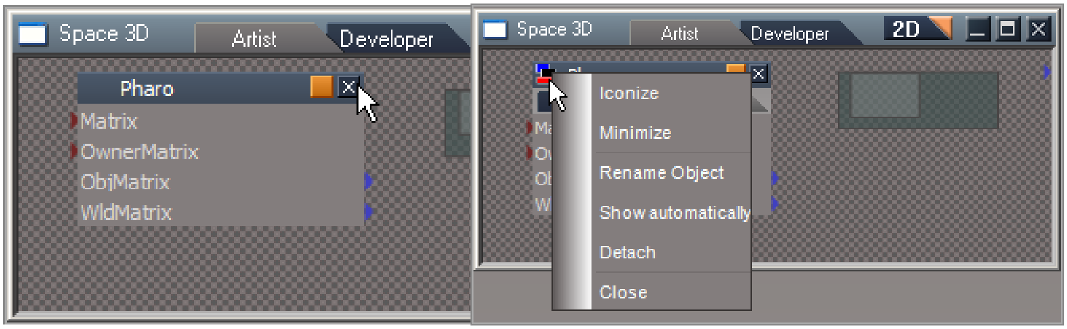
Adding/Removing Objects

To add an object to the Link Editor from a library, simply drag and drop it into the Link Editor view. To do this, first make sure that the container object where you want to add the new object is opened in the Link Editor, and then drag the object from library and release the mouse button over the Link Editor view. The new object will be added to the opened object.



Dragging an object from a library into the Link Editor

To remove an object from the Link Editor view, click the Close button on the object’s panel, or use the menu and choose close , or press Delete on the keyboard with the object selected. All connections to the deleted object will also be destroyed.



deleting an object from the Link Editor

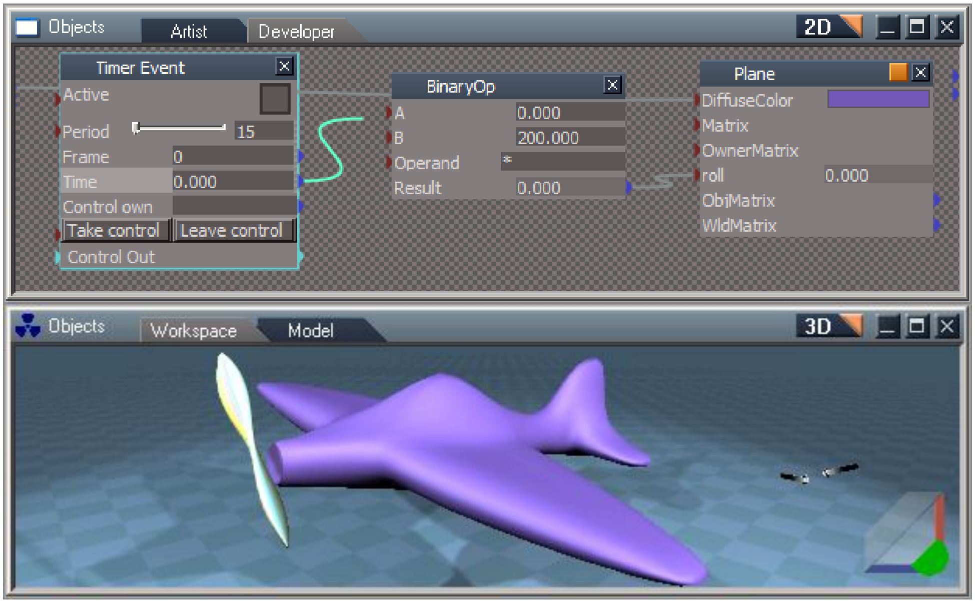
Creating Connections

You can connect the attributes of two or more objects using dependency links. These attributes must be of compatible types. In other words, you must connect colors to colors, numbers to numbers, meshes to meshes, etc. Attempting to connect a mesh type attribute with a matrix type attribute will not work; no link will be formed.

Before creating the connection, switch both objects to the Expanded, Default, or another aspect where you can see their attributes. Mouse over the first attribute and press the left mouse button, then drag the mouse to the second attribute and release the mouse button. If the attributes are compatible, and one is an output and the other is an input (see below), then a connection will be created.

Attributes which send messages to other attributes are called output connectors. They are represented by a triangular blue socket positioned on the bottom or right side of an object’s panel. Attributes that are able to accept messages from other attributes are input connectors. They are represented by red triangles positioned on the top or left side of the panel boundary.

Below a Timer event (Objects-Tutorial-Objects Library) Time is connected to A in the BinaryOp (Math-Operations Library) BinaryOp input B value is changed to 200 and the Operand is left at * (multiply) the result is then linked to the roll on the Plane which has been exported from inside the Plane for the propeller's expanded matrix.



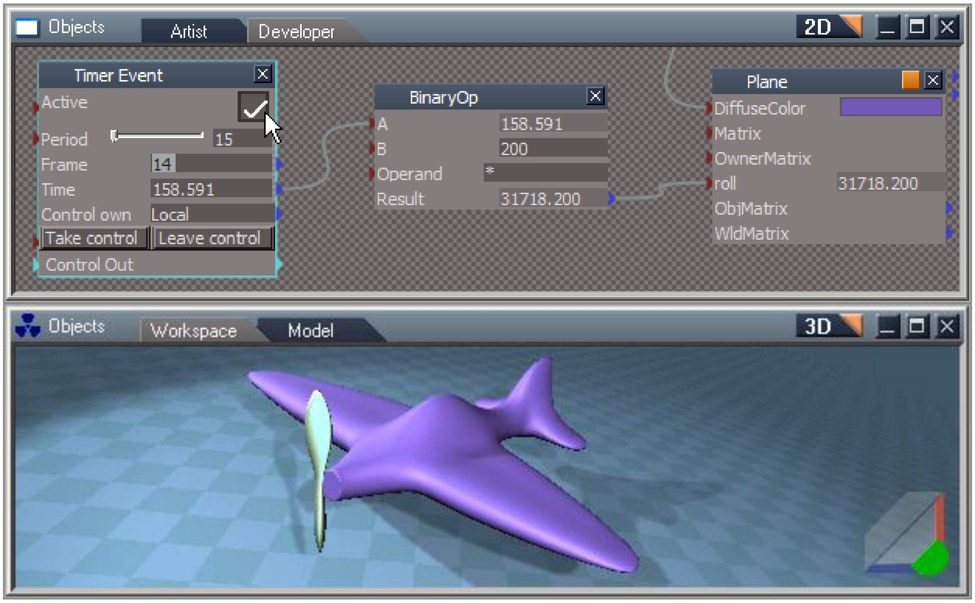
Timer Event
Active
Period 15
Frame 0
Time 0.000
Control own

BinaryOp
A 0.000
B 200.000
Operand *
Result 0.000

Plane
DiffuseColor
Matrix
OwnerMatrix
roll 0.000
ObjMatrix
WldMatrix

Click and drag to connect two attributes in the Link Editor

When the timer is activated the propeller will rotate as the data from the Time is sent through the link to the BinaryOp which multiplies the input value by 200 and passes the result to the roll on the plane.



Timer Event
Active ✓
Period 15
Frame 14
Time 158.591
Control own Local

BinaryOp
A 158.591
B 200
Operand *
Result 31718.200

Plane
DiffuseColor
Matrix
OwnerMatrix
roll 31718.200
ObjMatrix
WldMatrix

When you finish the connection a new dependency link will appear

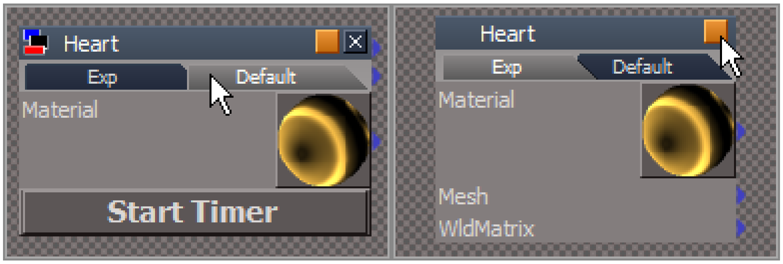
Disconnecting and Reconnecting

You can also disconnect and reconnect dependency links. To change the attribute connected by a dependency link just mouse over the link near that attribute and press the left mouse button. This will disconnect the link at that point. Now, drag the link to another compatible attribute and release the mouse button to create a new link. If you release button without moving to an attribute, the link is deleted.

Exporting Attributes

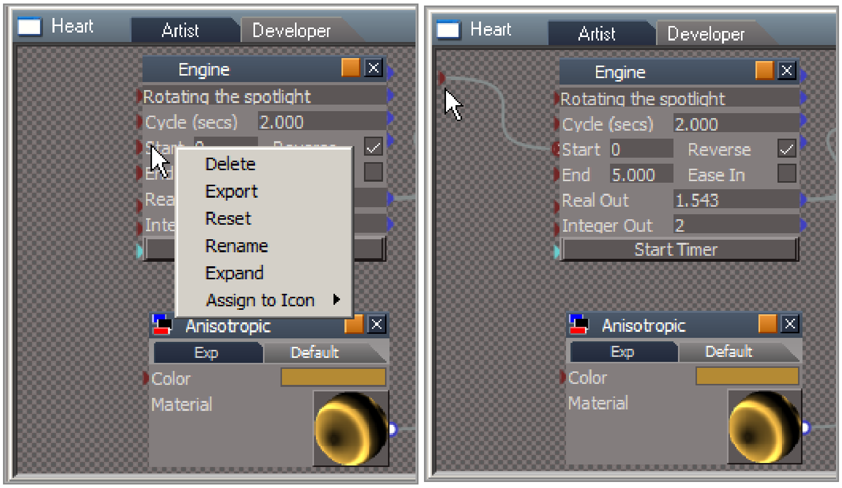
You can export attributes to the outside of a compound object so that it can be connected to other objects or attached to a control on outer panel. By doing this you can build an **interface** which other objects or you can access and change, and with which the object can communicate through links with other objects.

To see how you can export internal attributes, load the "Heart" object from the Objects-Base Library into the Link editor. Change its panel to the Exp aspect. You will see that this object has 3 exported attributes: *Material*, *Mesh* and *WldMatrix*.
Let us export two more attributes (start and end of pulsating cycle) from the inside of the Heart object to the front topmost panels aspect by first entering the Heart object in the Link Editor by clicking on the orange triangle in its title-bar.

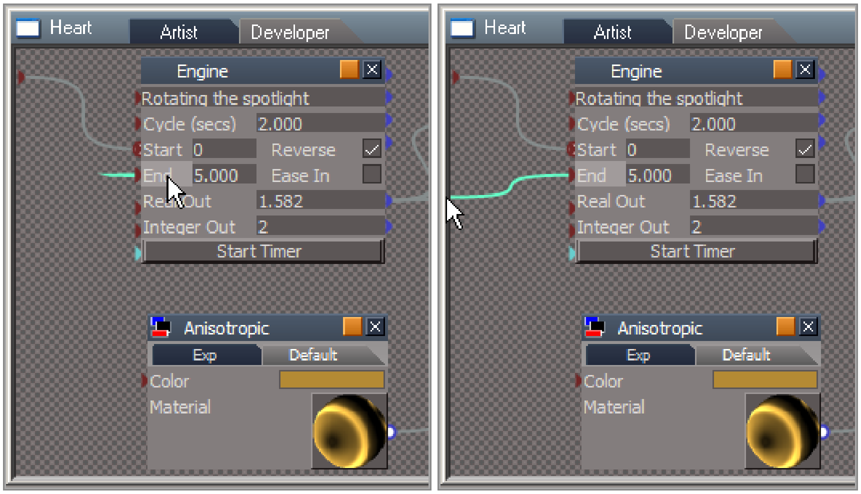


Original panel

Make sure you turn the Developer aspect on so that you can see all the interior links. Then left-click on the two controls (which are on the panel of the internal object called Engine), and drag both attributes outside of the Link Editor view's left edge before releasing the mouse button. You should see bright green link as you drag. You can also right-click an attribute and select the Export option to perform this task.

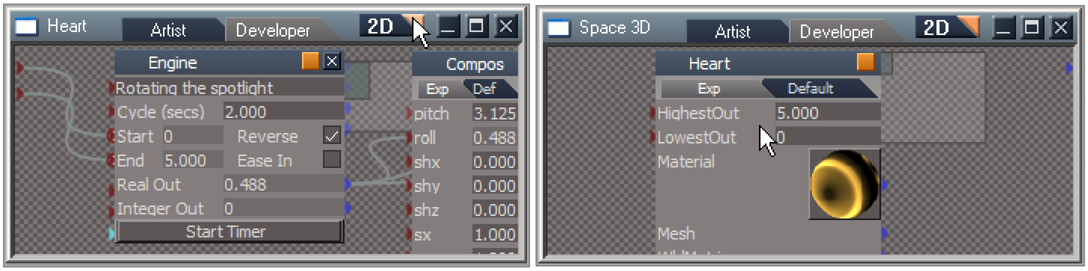


Right-click commands for attributes



Start and End being exported outside of the Heart object

Depending on the type of attribute exported (input or output) the new attribute will now appear on the panel of the outer container object. Now, both attributes are exported to the Heart object Exp aspect panel these two additional attributes allow you to specify the range of Heart's beat directly from the Heart object panel.



Rotating the spotlight
Cycle (secs) 2.000
Start 0
Reverse ☑
End 5.000
Ease In □
Real Out 0.488
Integer Out 0

pitch 3.125
roll 0.488
shx 0.000
shy 0.000
shz 0.000
sx 1.000

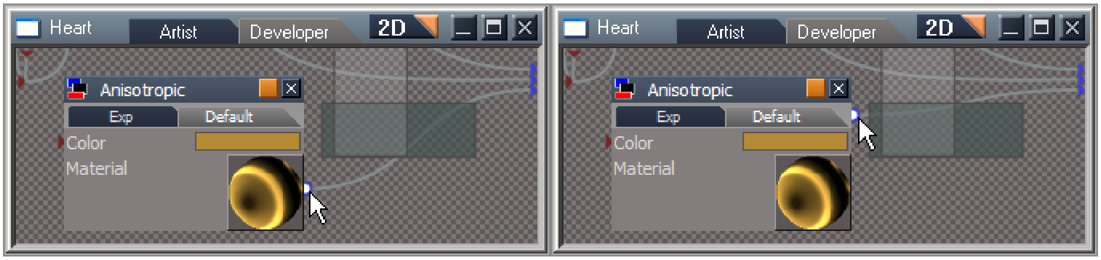
HighestOut 5.000
LowestOut 0
Material
Mesh

new attribute will now appear on the panel of the outer container object

- Reference: Panel Editing

Moving Connectors

To help simplify and organize the visual content of the object and to reduce link clutter in the Link Editor. The position of an object’s connector sockets (red and blue triangles) can be adjusted in any aspect state.
Mouse over the desired connector and press CTRL and the left mouse button, then drag the mouse to position the connector along the outside edge of the object.

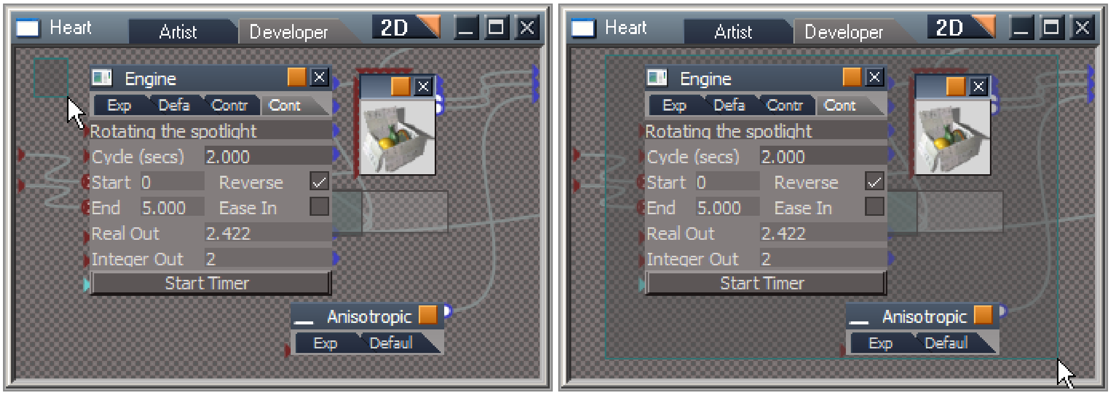


Position connectors by holding CTRL and dragging

Dragging connectors around an object’s panel can be very useful when you want to achieve a clean and uncluttered layout inside the Link Editor:

Rectangle Selection

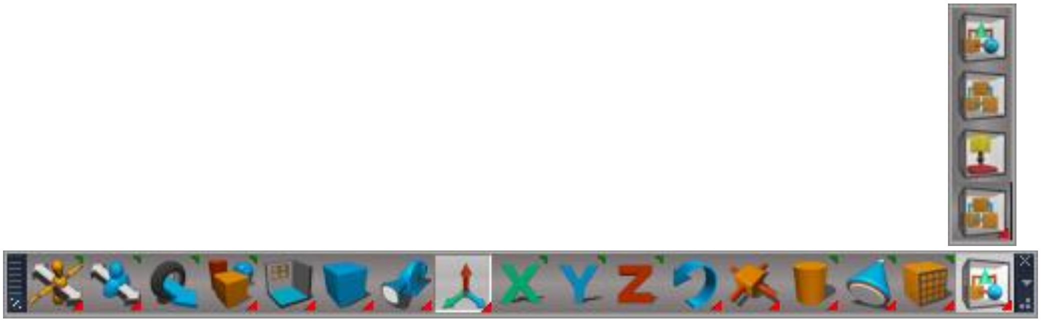
Use the **Rectangle Selection** tool to select multiple objects in the Link Editor. This tool allows you to select a group of objects. Click and drag anywhere on the Link Editor background.
Left-click and drag your mouse, to draw a rectangle around the desired objects then release the mouse button to select the objects encompassed by the rectangle.



Rotating the spotlight
Cycle (secs) 2.000
Start 0 Reverse ☑
End 5.000 Ease In □
Real Out 2.422
Integer Out 2
Start Timer

Left-click and drag to create a rectangular selection

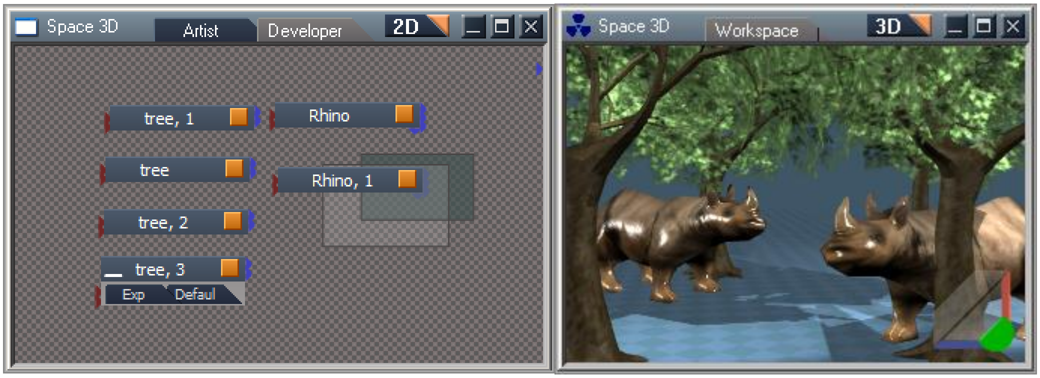
2.8.5 Encapsulating Objects



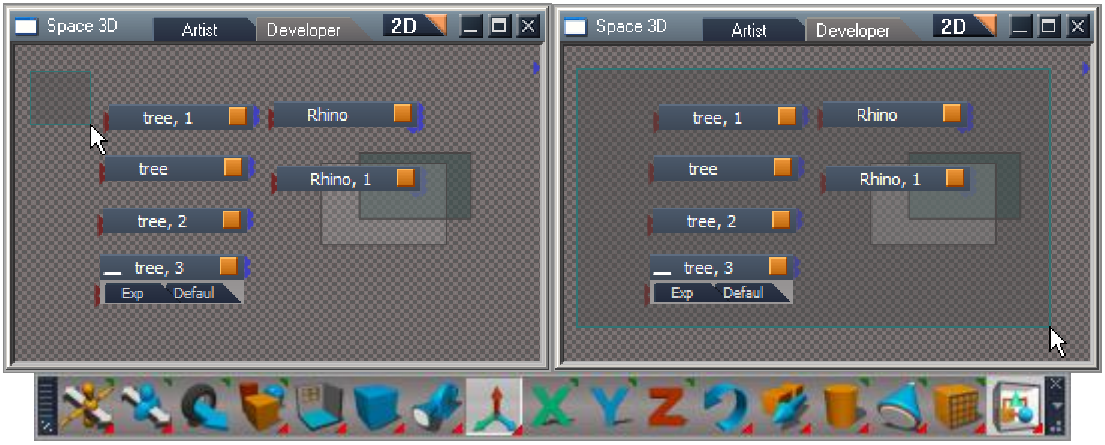
Encapsulate in 3D

The Encapsulate In 3D tool makes a group, and within that group each original object is still a separate individual entity, which can be manipulated separately from the other objects in the encapsulated group.

Example: the scene has several elements , rhinos and trees which can be managed better by combining them into a single object, this will allow for easier and less cluttered scene management and also the group of objects can be saved and copied the same as any regular object.



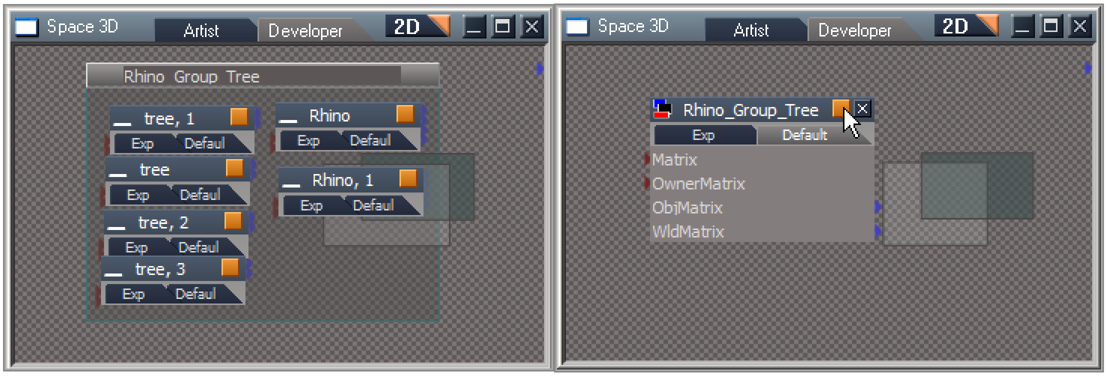
To encapsulate a group of objects, first select the objects that you want to assemble into the new object using rectangle selection, and then click the encapsulate in 3D button.



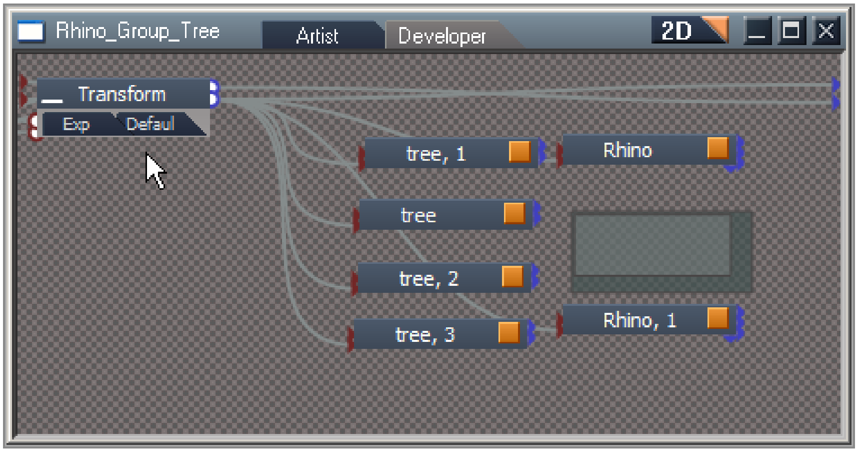
Rectangle select the elements and click on encapsulate in 3D

Enter a name for the new object that will contain the grouped elements.
After you press Return, the assembly of selected objects are contained inside the new object.

When entering the newly created object by clicking on its orange enter-context button. the encapsulated group now share a common transform, and can be moved as a group when the object is selected, or if navigated into then each object still exists as a separate entity and can be selected and moved independently from the entire group.



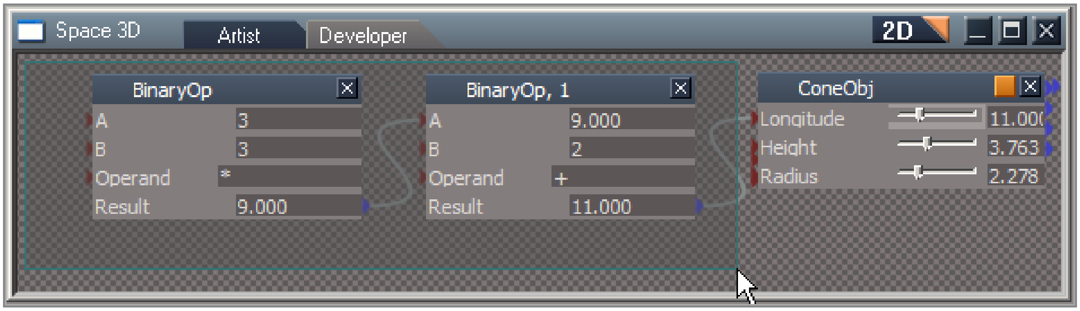
Give a name and press Return on the keyboard, then enter the new object by its orange square



Inside the group each original object is still separate and can be selected individually

Encapsulate Objects

The **Encapsulate** tool, located in the Link Editor toolbar, is used to enclose a group of selected objects inside a new container object. Any connected attributes will be automatically exported from the container object, and existing links will be maintained. This tool is best used on objects that don't have a mesh representation e.g. for math functions or scripts , if you need to be able to see and move a mesh based object in the Workspace then you would use the encapsulate in 3D tool instead.



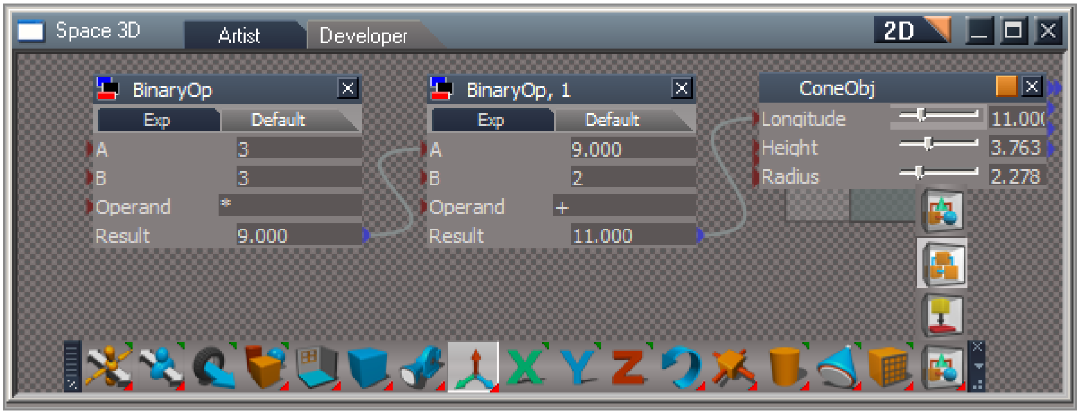
BinaryOp
A 3
B 3
Operand *
Result 9.000

BinaryOp, 1
A 9.000
B 2
Operand +
Result 11.000

ConeObj
Longitude 11.000
Height 3.763
Radius 2.278

Existing context with three objects

To encapsulate a group of objects, first select two or more objects to assemble into the new object using rectangle selection, and then click the encapsulate button. Finally, enter a name for the new object.

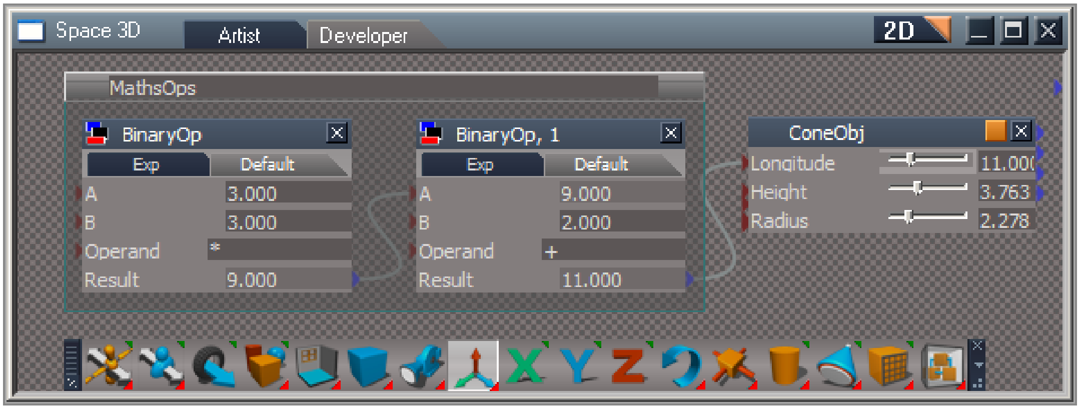


BinaryOp
A 3
B 3
Operand *
Result 9.000

BinaryOp, 1
A 9.000
B 2
Operand +
Result 11.000

ConeObj
Longitude 11.000
Height 3.763
Radius 2.278

Two selected objects just before creation of a new one



MathsOps

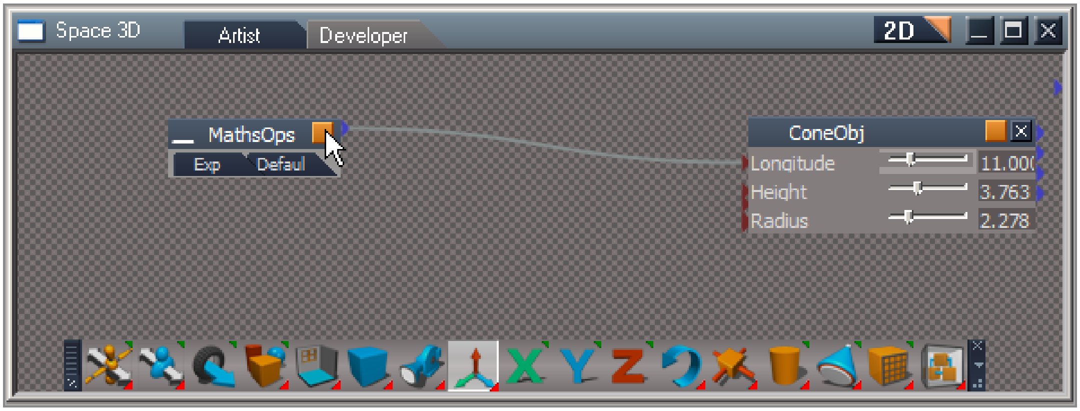
BinaryOp
A 3.000
B 3.000
Operand *
Result 9.000

BinaryOp, 1
A 9.000
B 2.000
Operand +
Result 11.000

ConeObj
Longitude 11.000
Height 3.763
Radius 2.278

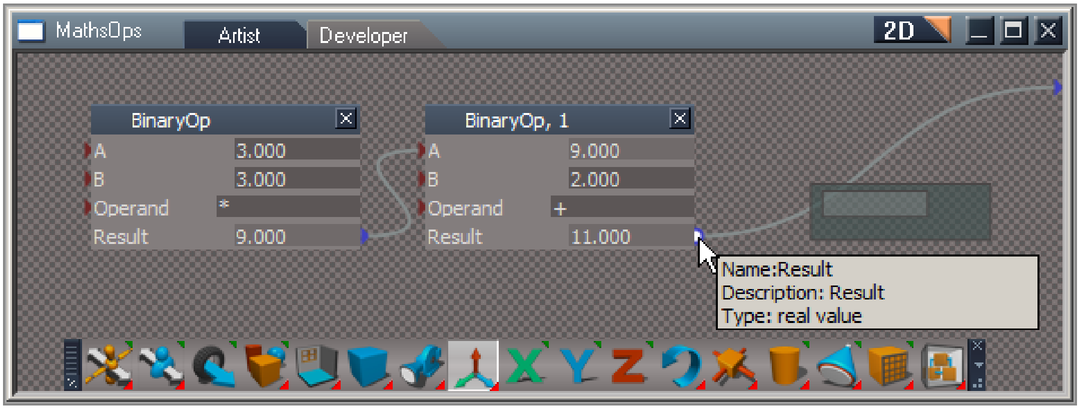
Illustration showing encapsulation and naming of new object

After pressing Return, the assembly of existing objects are contained inside the new object.
Enter the newly created object by clicking on its orange enter button and export any attributes that are needed on the top level of the new object, e.g. A from BinaryOp and B from BinaryOp 1 and changing those values will change the result linked to the ConeObj Longitude.



Longitude 11.00(
Height 3.763
Radius 2.278

New object replaces previous selected objects

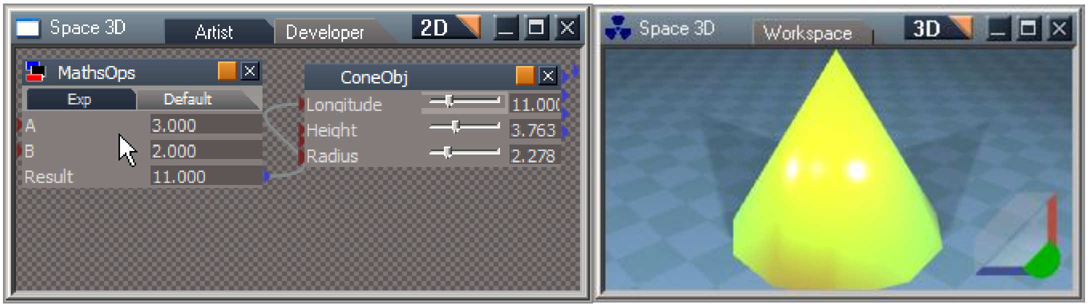


BinaryOp
A 3.000
B 3.000
Operand *
Result 9.000

BinaryOp, 1
A 9.000
B 2.000
Operand +
Result 11.000

Name:Result
Description: Result
Type: real value

MathOp object entered the linked result to the cone obj is retained and exported automatically



MathsOps

A 3.000

B 2.000

Result 11.000

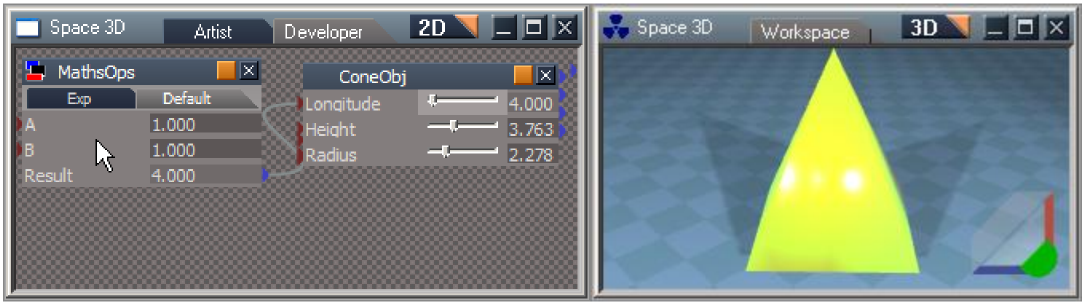
ConeObj

Longitude 11.000

Height 3.763

Radius 2.278

exported attributes on the top level of the new object



MathsOps

A 1.000

B 1.000

Result 4.000

ConeObj

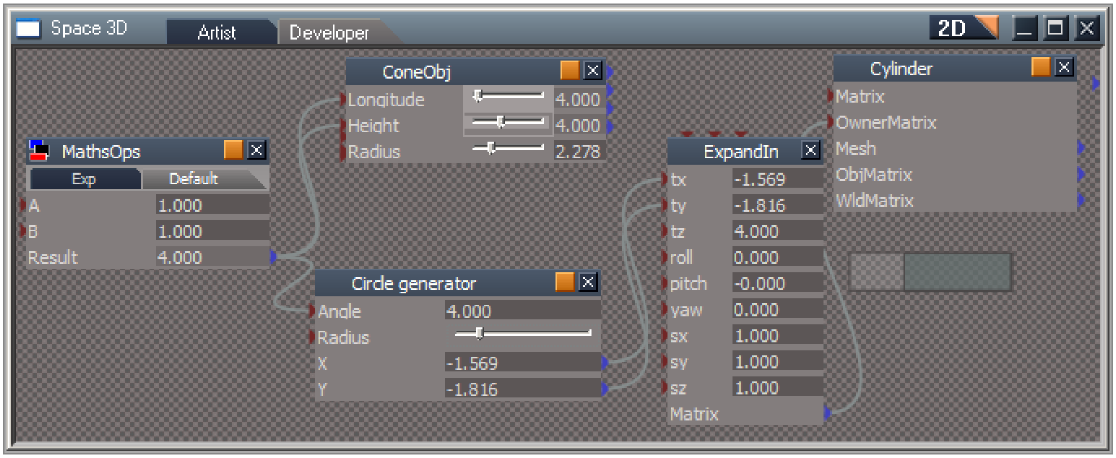
Longitude 4.000

Height 3.763

Radius 2.278

Changing the values on the exported attribute for the A and B values changes the longitude of the Cone

The output Result from the MathsOps could be linked to other objects that accept its type of values as well or even link it to multiple objects to drive the data and results..



MathsOps

A 1.000

B 1.000

Result 4.000

ConeObj

Longitude 4.000

Height 4.000

Radius 2.278

Circle generator

Angle 4.000

Radius

X -1.569

Y -1.816

ExpandIn

tx -1.569

ty -1.816

tz 4.000

roll 0.000

pitch -0.000

yaw 0.000

sx 1.000

sy 1.000

sz 1.000

Matrix

Cylinder

Matrix

OwnerMatrix

Mesh

ObjMatrix

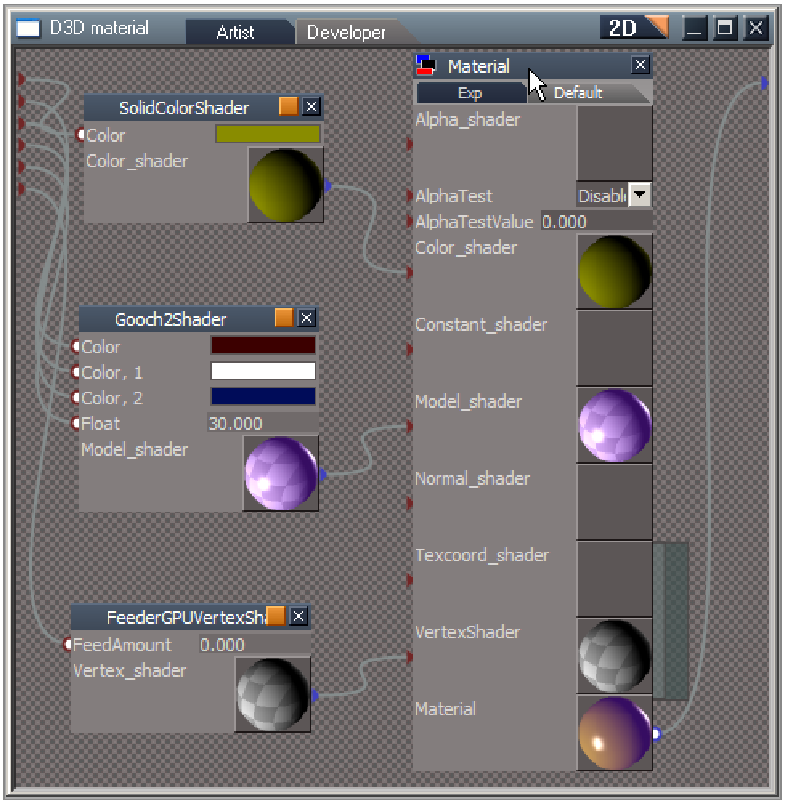
WldMatrix

MathsOps delivers its result to multiple inputs on different objects

2.9 Editing Materials

The Link Editor can also be used to create materials for your objects using various components. These components work much like other objects in the Link Editor. They have attributes and panels, and you can connect them together to create more complex structures from simple objects.
The Material editing capabilities and use is covered in detail in chapter 5 and can be accessed by following the link below.

- Reference PDF Manual Section 5.3 Workspace Material Editor



Float 30.000

FeedAmount 0.000

AlphaTest Disabl

AlphaTestValue 0.000

Inside a Material

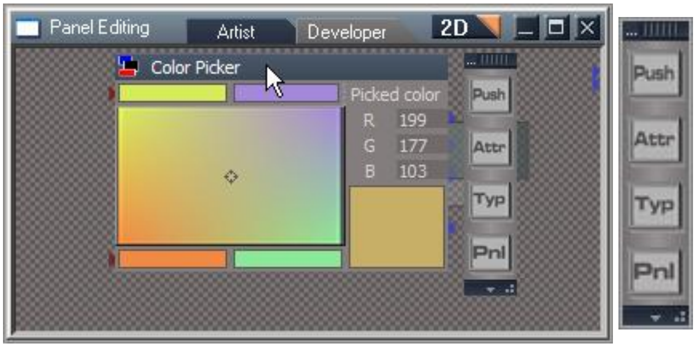
2.10 2D View – Panel Editor

2.10.1 Panel Editing

You can quickly and easily change the layout of an object’s panel, add new controls using the trueSpace Panel Editor, to give your objects interesting and meaningful and aesthetic interfaces for yourself or others to use and interact with.

Panels display user-friendly controls of an object in a 2D aspect. These controls are linked to one of the object’s attributes and allow for changing their values using the mouse (by clicking or dragging a slider) or the keyboard (by typing into an edit field).

Panel Editing mode is activated by right-clicking on the object’s title bar where you can change the position, appearance, and even the type of controls for a given attribute. In this mode the main Panel Edit library will appear to the right of the object,



Right-click in the title bar enters Panel Edit mode.

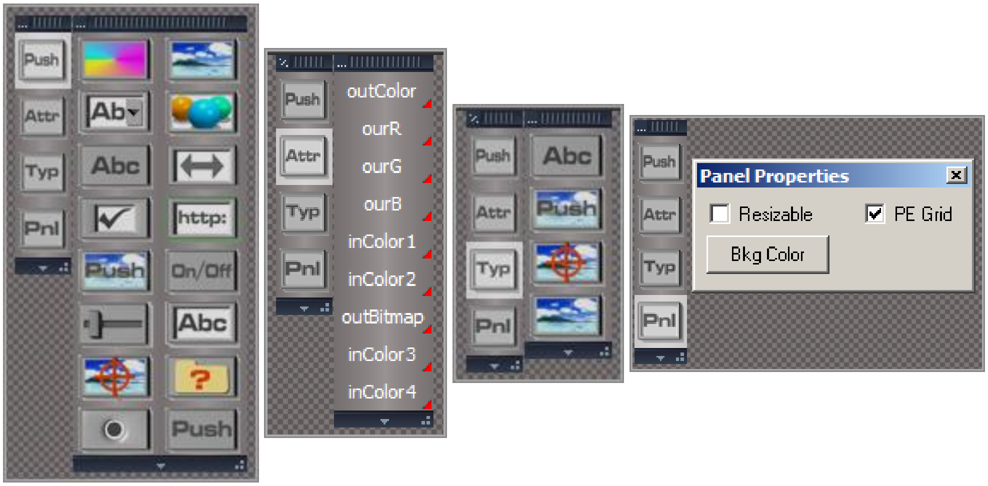
The Panel Editor Toolbar has four library categories:

- Push **Push Button Icon**: Opens the Controls library. This library contains all of the controls available for the Panel. Left-clicking any control adds this control to the panel.

Attr **Attr**: Opens the Attribute library. This library contains all the visible attributes of the object this will differ from object to object depending on the Attributes that already exist.

Left-click and hold on any attribute in this library, will show all the control types that are available for that attribute. Select a control to add it to the panel and automatically attach it to that attribute. Left clicking an attribute without holding the button down will add a text control (a label) displaying the name of the attribute.

- Typ **Typ**: Opens the Type library. This library will show just those type-compatible controls that can replace the currently selected control. When added, the new control is linked to the same attribute as the old control.
- Pnl **Pnl**: Opens the Panel properties dialog. In this dialog for changing panel settings like background color, whether controls are resized and moved with big steps (PE grid), or if the control is resizable.

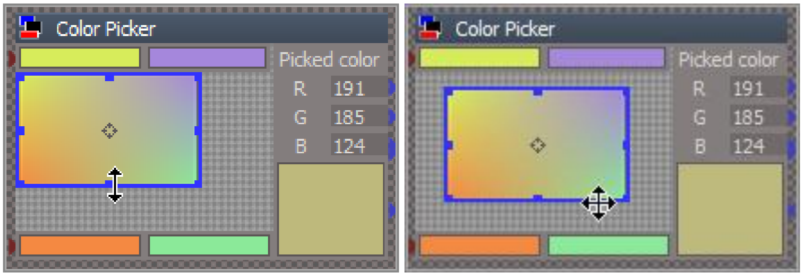


☐ Resizable ☑ PE Grid

Panel Edit Mode toolbars

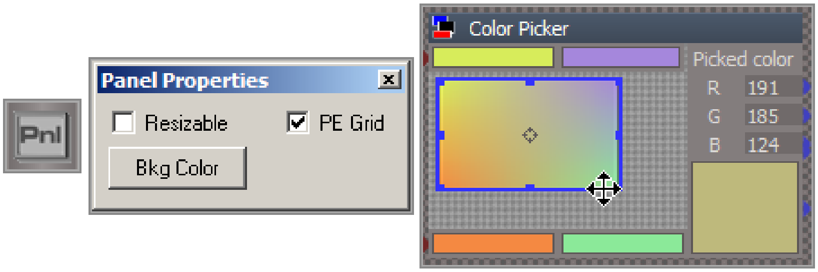
When in Panel Edit mode, left-click on any control to select it.
A blue rectangle with drag able handles will highlight the control. The handles can be used to move and resize the control. The control can also be moved using the arrow keys.

Controls can be resized by pressing SHIFT+arrow combinations, and CTRL+SHIFT+arrow will change the control size by 1 pixel.



Using the handles to move and resize a selected control

If the Panel Editor grid is enabled, any movement or resizing will snap to the grid; when off, the control can be moved and resized freely. The panel grid can be temporarily disabled by pressing the **CTRL** key. The same applies to moving the controls with the keyboards arrow keys.



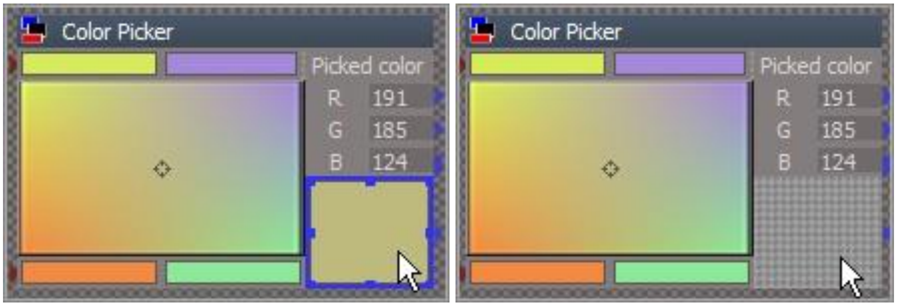
Panel Properties

☐ Resizable ☑ PE Grid

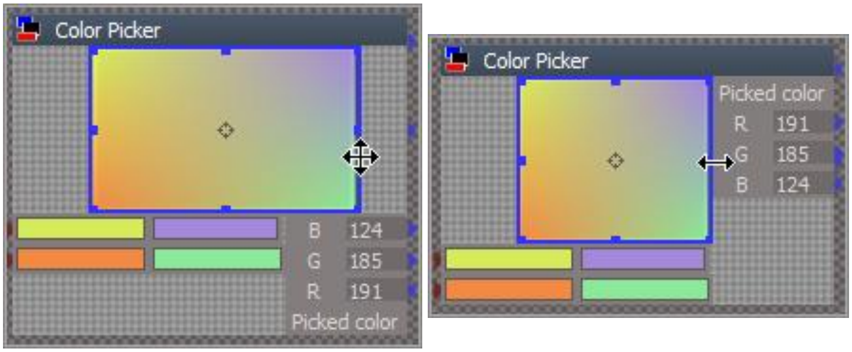
Bkg Color

Holding CTRL and using the arrow keys or the move with the cursor allows for finer movement

Pressing the **DEL** key will remove the selected control from the panel.

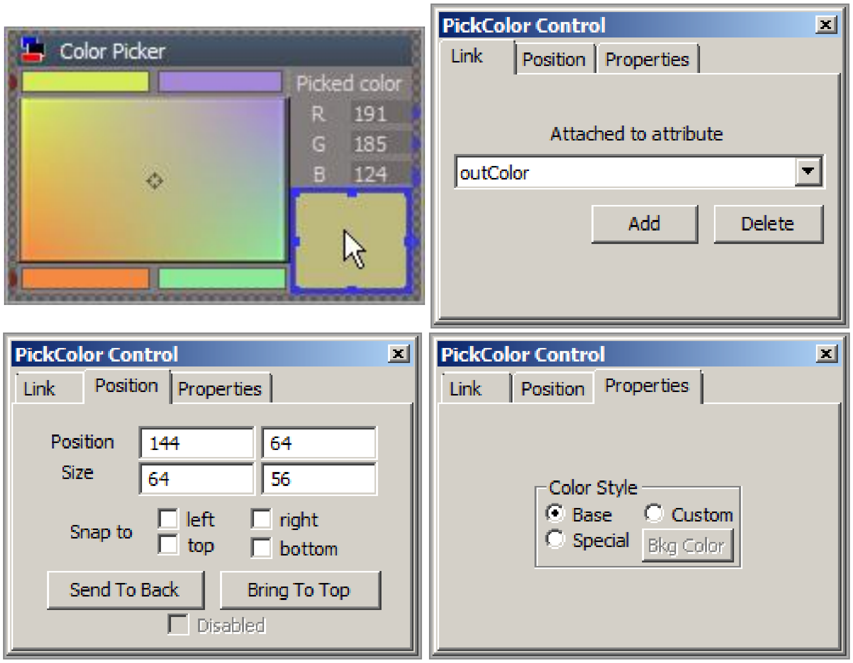


When a control is dragged in edit mode they collide and push each other away. If the control is wide, it pushes other controls beneath it; if it is tall, it pushes other controls sideways. When the selected control is moved out of the way, the displaced controls will tend to return to their previous positions.



Picked color, R,G,B, Values, displaced, when the control collides with them, then when the controls moved out the way they move back into their original positions

Right-clicking on a selected control will open up its property panel with three tabs: **Link**, **Position** and **Properties**.



PickColor Control — Link

Attached to attribute: outColor

PickColor Control — Position

Position	144	64
Size	64	56

Snap to □ left □ right □ top □ bottom

□ Disabled

PickColor Control — Properties

Color Style: ● Base ○ Custom ○ Special

Properties tab

- **Link**: Select the attribute connected to this control from the drop-down list. The control is then attached to this attribute and displays the attribute's values if the control is of a compatible type.

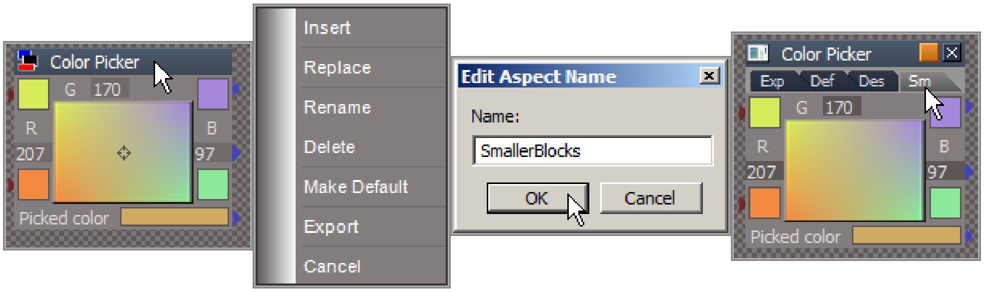
- **Position**: Set the control's exact position and size. You can also set snapping options, i.e. whether the control is aligned with the border when resizing or moving. The "Bring to Top" button moves the selected control to the top of the z-order so that all other controls are beneath it. "Send to Back" causes the selected control to be moved beneath all other controls in z-order.
- **Properties**: This displays properties that are different for each control. These property dialogs are described in the next section.

Any changes to panel properties take effect immediately. To close the properties dialog, use the close button at the upper right corner of the title bar, or right-click the selected control again.
When another you control is selected (by left-clicking it), this dialog will display the properties for the newly selected control.

2.10.2 Save or exit panel edit mode.

Panel edit mode is exited by right-clicking the object's title bar, and selecting a choice of options from a menu:

- **Insert**: Adds the newly created aspect to the list of aspects. You can give the aspect a name using the small dialog or the expanded aspect popup.
- **Replace**: Replaces the old aspect (the aspect that was active before editing) with the newly created aspect.
- **Rename**: rename the selected aspect.
- **Make Default**: allows you to make current aspect the default for panel.
- **Export**: Exports the current aspect for the panel to make the settings available in the stack.
- **Cancel**: Exit edit mode and leave aspects unchanged.



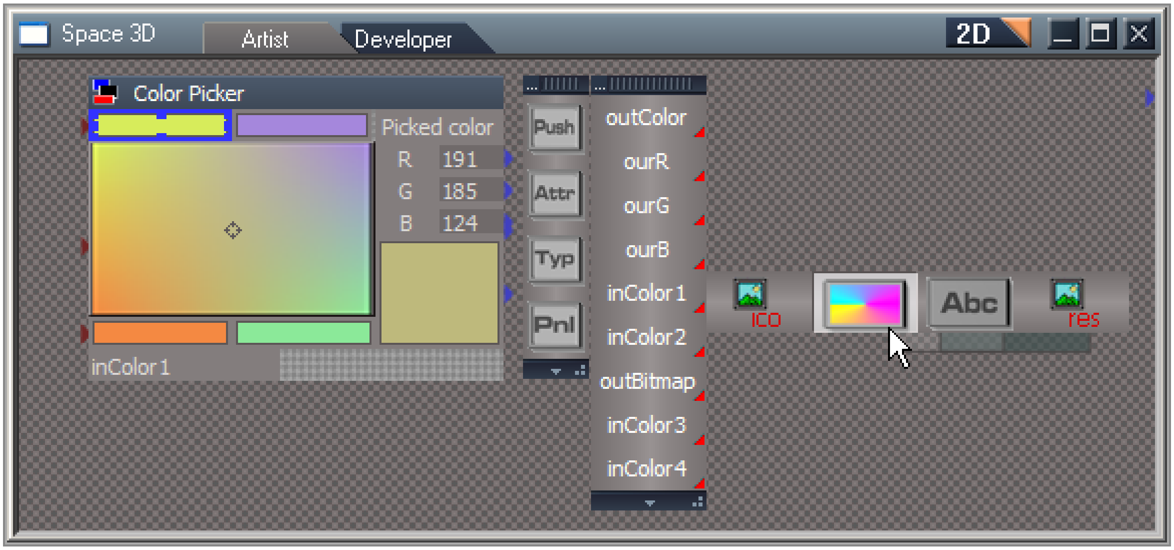
Name: SmallerBlocks

R-Click the title bar Select Insert, Name the edited aspect, and its added to the tabs on the object

Some important points to note when building your own panels:
When a Panel is created, trueSpace creates three default aspects: **Minimized**, **Expanded**, and **Default**.

The **Minimized** aspect displays only the title bar, with no controls. The **Expanded** aspect only displays all visible attributes with no edit controls. The **Default** aspect is editable by the users and should show only most useful controls.

You can build panels using a **top-down** approach, adding controls using the Panel Editor's Control library and linking them to attributes with the Link tab of the control's property dialog.
Or, you build them using a **bottom-up** approach, adding controls from the Attribute library, which will be automatically attached to the selected attribute.

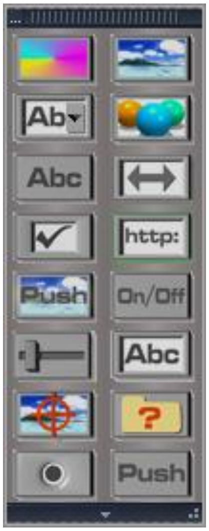


Adding and modifying user controls on a panel

2.10.3 Control Types

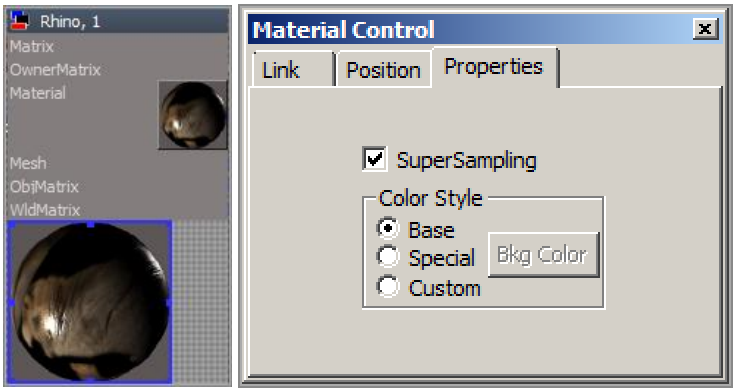
A wide variety of controls can be added to an object's panel from the Panel Editor's Control and Attributes libraries: Here is a list of controls and a description of the properties available for each item.

Push ToolButton Controls:

Main	*Push ToolBar*	*Controls*			
		🟪	***PickColor***	🖼	***Bitmap***
		Ab▾	***Combo***	●	***Material***
		Abc	***Text***	↔	***ToolBar Scrubber***
		☑	***CheckBox***	http:	***HTML View***
		Push	***Bitmap Button***	On/Off	***ToggleButton***
		⊣	***Slider***	Abc	***Edit Control***
		⊕	***Clickable Bitmap***	?	***OpenFile***
		◉	***RadioButton***	Push	***Button***

Material Preview

The Material Preview control renders the materials, multi-materials, or shaders in real-time, and reacts to attribute changes.



Rhino, 1
Matrix
OwnerMatrix
Material
Mesh
ObjMatrix
WldMatrix

Material Control

Link | Position | Properties

☑ SuperSampling

Color Style
● Base
○ Special
○ Custom

Bkg Color

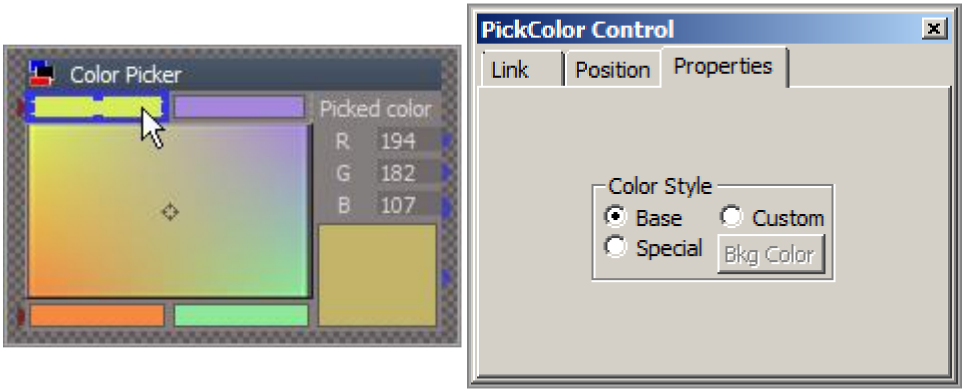
Material Preview

- **Supersampling**: Smoothes the appearance of the material.

    - Color style
- **Base**: use trueSpace default color scheme.
- **Special**: gives the selected field a lighter background color. Useful for title areas or explanatory areas of the panel..
- **Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color of the control.

Pick Color

Pick Color allows you to select a color as an attribute. Select a new color by double-clicking the control. The chosen color will be assigned to the attribute and displayed in the control.



Color Picker

Picked color

R 194

G 182

B 107

PickColor Control

Link | Position | Properties

Color Style

● Base ○ Custom

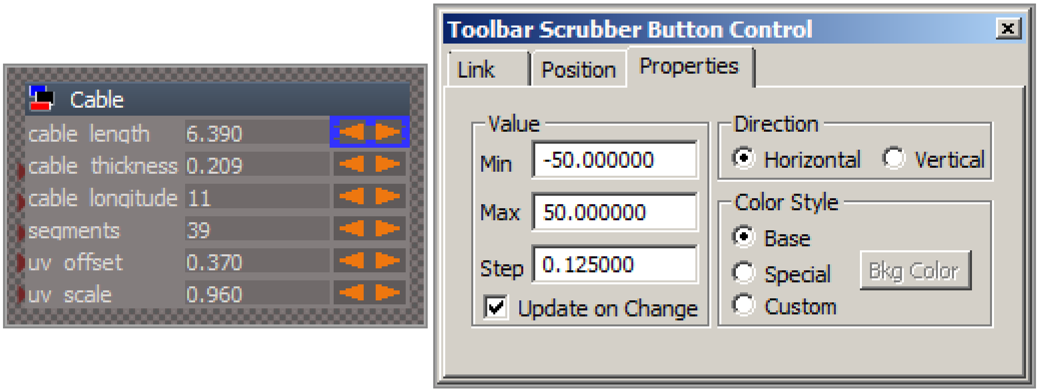
○ Special Bkg Color

Pick Color Control

- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color of the control.

Toolbar Scrubber

The Toolbar Scrubber can be used as a replacement for the slider control. It can take less space than a slider control .



Cable

cable length	6.390
cable thickness	0.209
cable longitude	11
segments	39
uv offset	0.370
uv scale	0.960

Toolbar Scrubber Button Control

Value

Min -50.000000

Max 50.000000

Step 0.125000

☑ Update on Change

Direction

● Horizontal ○ Vertical

Color Style

● Base

○ Special

○ Custom

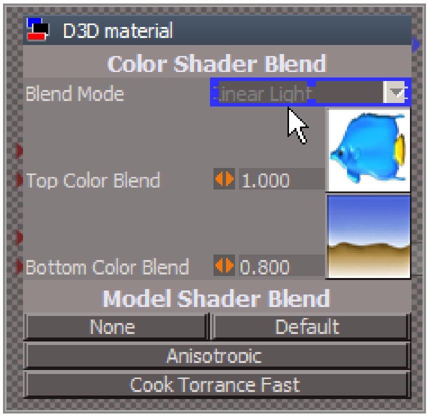
Cable object from Base Library uses Scrubber Control for adjusting its values

- **Min Value**: minimum value for scrubber.
- **Max Value**: maximum value for scrubber.
- **Step Value**: each click on scrubber changes value by this amount.
- **Update on Change**: When checked scrubber updates values when control is dragged on, if unchecked then values update on mouse release.
- **Direction**: set scrubber for horizontal or vertical orientation.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color of the control.

Combo Control

The Combo Control allows you to populate a drop-down list (one of several options) with values that can be called from a script. Depending on which option in the Combo Control you select, the script/code will react accordingly. In the example shown, the Script Editor would be programmed to execute a particular part of the code depending on which string has been selected in the drop down list.

This Shader was kindly supplied by Marcel Barthel as an example of using the combo control it uses various image blending processing techniques on the bitmaps loaded into its interface.


D3D material

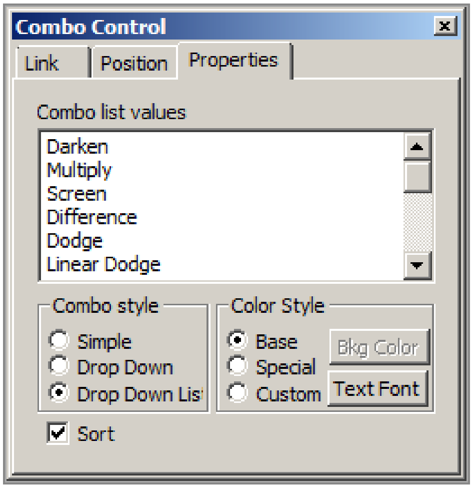
Color Shader Blend

Blend Mode: Linear Light

Top Color Blend 1.000

Bottom Color Blend 0.800

Model Shader Blend


Combo Control

Combo list values

Darken
Multiply
Screen
Difference
Dodge
Linear Dodge

Combo style

○ Simple
○ Drop Down
● Drop Down Lis

Color Style

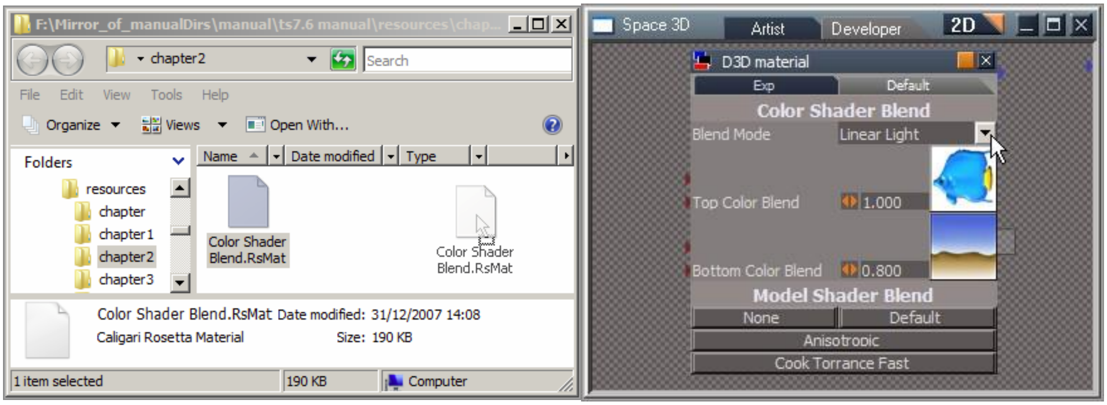
● Base
○ Special
○ Custom

☑ Sort

- **Combo List Values**: as shown above, a number of choices are input into the value area for selection.
- **Combo Style**: choose which style of Combo you wish to use depending on the scenario.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: set color as desired.
- **Bkg Color**: Opens a dialog for setting the background color of the control.
- **Sort**: check box determines if the values will be sorted or not.

➢ **Resource Color Blend Shader**

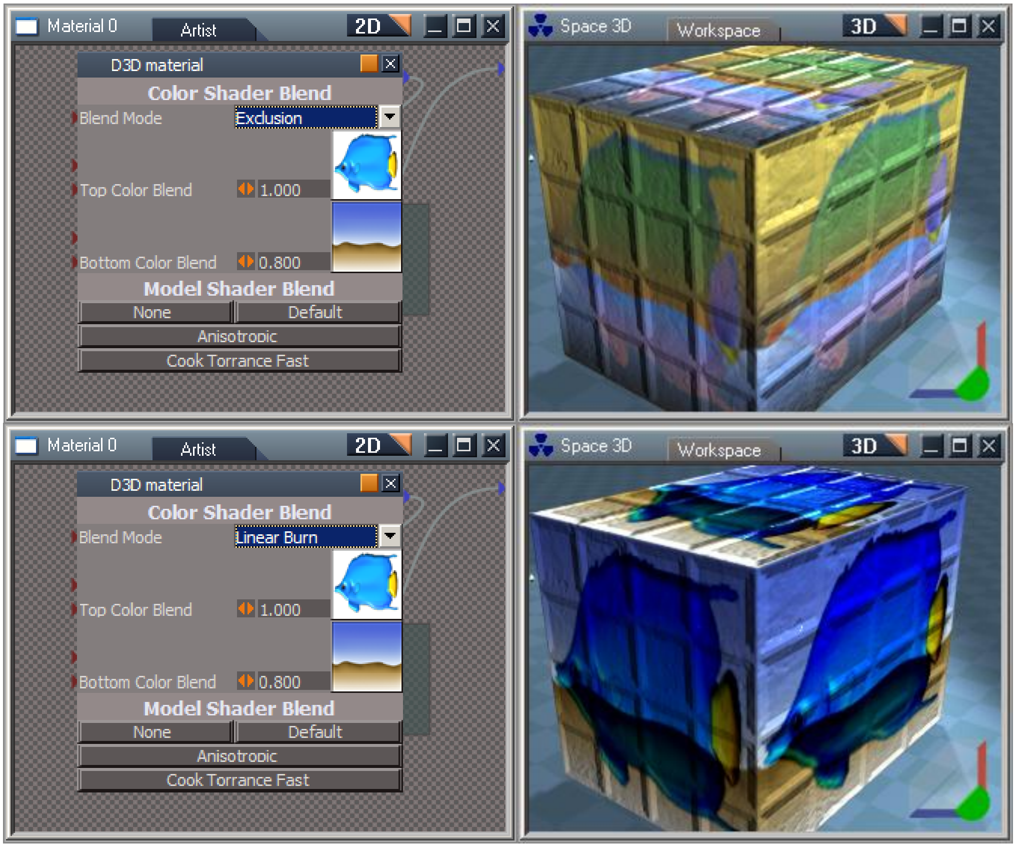
Follow the link and drag and drop the Color Shader Blend object to the Link Editor to study the way its uses the combo control , it is quite a complex shader so some digging around will be necessary to understand how this works .



Color Shader Blend.RsMat Date modified: 31/12/2007 14:08
Caligari Rosetta Material Size: 190 KB

Blend Mode: Linear Light
Top Color Blend: 1.000
Bottom Color Blend: 0.800

Color Blend dragged and dropped to the Link Editor



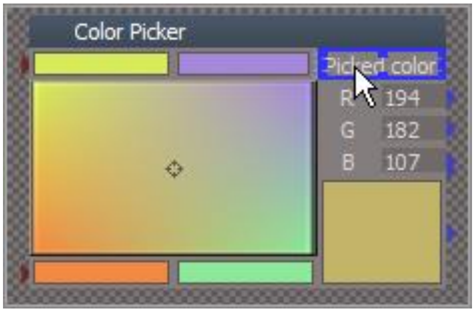
Blend Mode: Exclusion
Top Color Blend: 1.000
Bottom Color Blend: 0.800

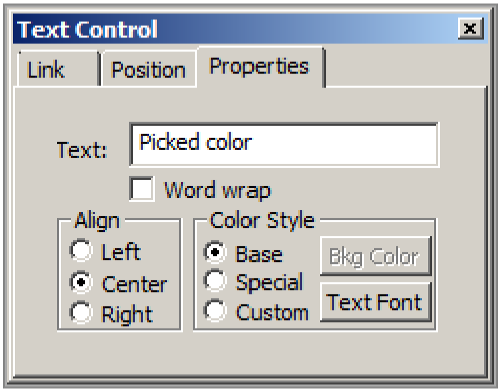
Blend Mode: Linear Burn
Top Color Blend: 1.000
Bottom Color Blend: 0.800

Shader applied to a cube and the selections picked from the combo dropdown list

Text Label

Labels are text strings that you can place on the panel. They are usually placed next to controls to explain their meaning, but they can be placed freely and display any text you want.





Text: Picked color

□ Word wrap

Align: ○ Left ● Center ○ Right

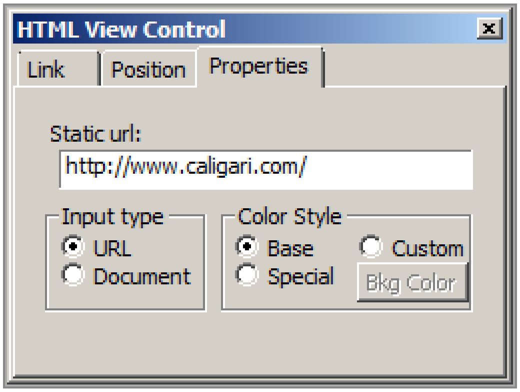
Color Style: ● Base ○ Special ○ Custom

Example Text Labels Text Control Panel

- **Text**: The text to be displayed in the control.
- **Word wrap**: Display long text as multiple lines.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used..
- **Bkg Color**: Opens a dialog for setting background color.
- **Text Font**: Opens a dialog for setting the font and color of text.
- **Align**: Changes the alignment of displayed text.

HTML View Control

The HTML View Control allows you to load up html files inside the Link Editor.
If Document input type is chosen it can be attached to any of string type attributes (connectors). The string value of the connector is converted to html document and displayed in the controls area. It also accepts html tags so it is possible to format the output. A main advantage is that the display and content can be dynamically changed using the connectors value.



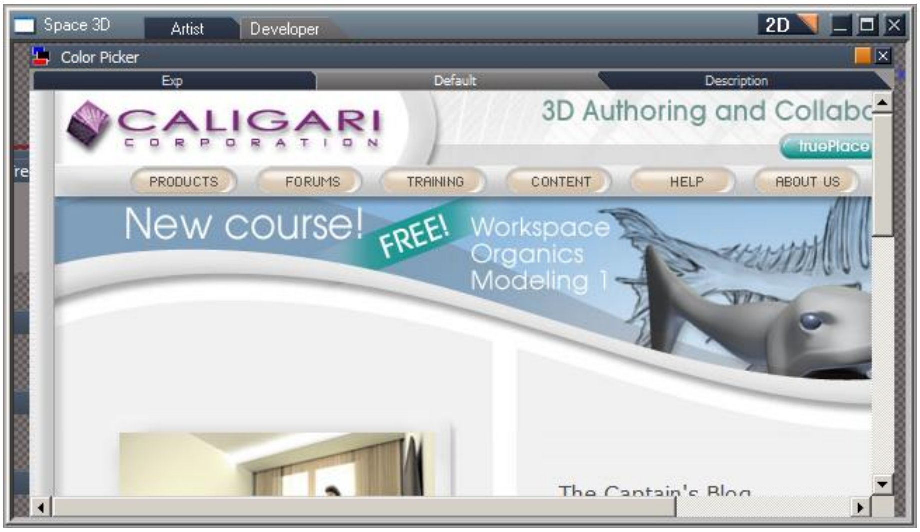
HTML View Control

Static url: http://www.caligari.com/

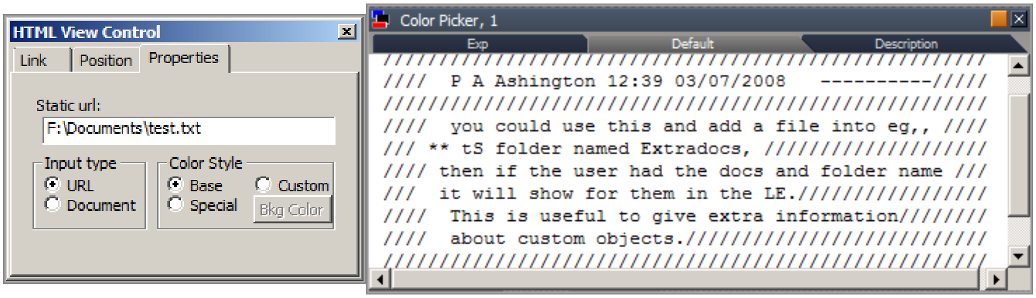
Input type: ● URL ○ Document

Color Style: ● Base ○ Special ○ Custom

- **Static URL**: as shown above, enter url of web page you wish to load into the control.
- **Input Type URL**: tells trueSpace you wish to enter a url.
- **Input Type Document**: tells trueSpace you wish to load **other than** a URL. For instance you may wish to place HTML code that would open a PDF or a Media file of some type. You could also add formatted text in HTML code.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color of the control.



Load HTML pages or html code into trueSpace Link Editor



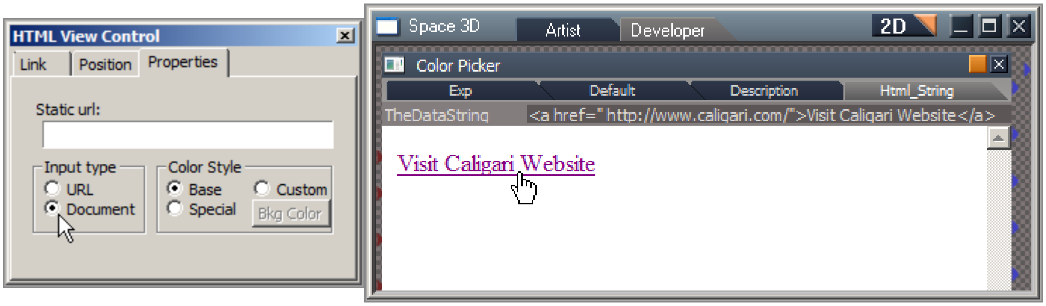
Static url: F:\Documents\test.txt

Input type: ● URL ○ Document

Color Style: ● Base ○ Custom ○ Special

```
//////////////////////////////////////////////////////
////  P A Ashington 12:39 03/07/2008   ----------/////
//////////////////////////////////////////////////////
////  you could use this and add a file into eg,, ////
/// ** tS folder named Extradocs, ////////////////////
//// then if the user had the docs and folder name ///
///  it will show for them in the LE./////////////////
////  This is useful to give extra information////////
////  about custom objects.///////////////////////////
//////////////////////////////////////////////////////
```

Txt document set as local url



Static url:

Input type: ○ URL ● Document

Color Style: ● Base ○ Custom ○ Special

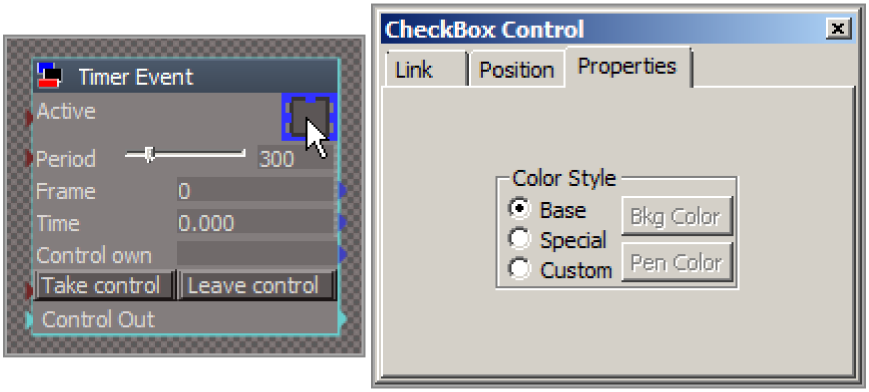
TheDataString <a href="http://www.caligari.com/">Visit Caligari Website</a>

Visit Caligari Website

Document string set with html formatting , the hyperlink will open the page in the Html controls window

Check Box

The Check Box control has two states: checked and unchecked. You can use it to control Boolean (true and false) or numerical object (0 and -1) attributes. Using this control, you can change any attribute with only two states.



Active

Period 300

Frame 0

Time 0.000

Control own

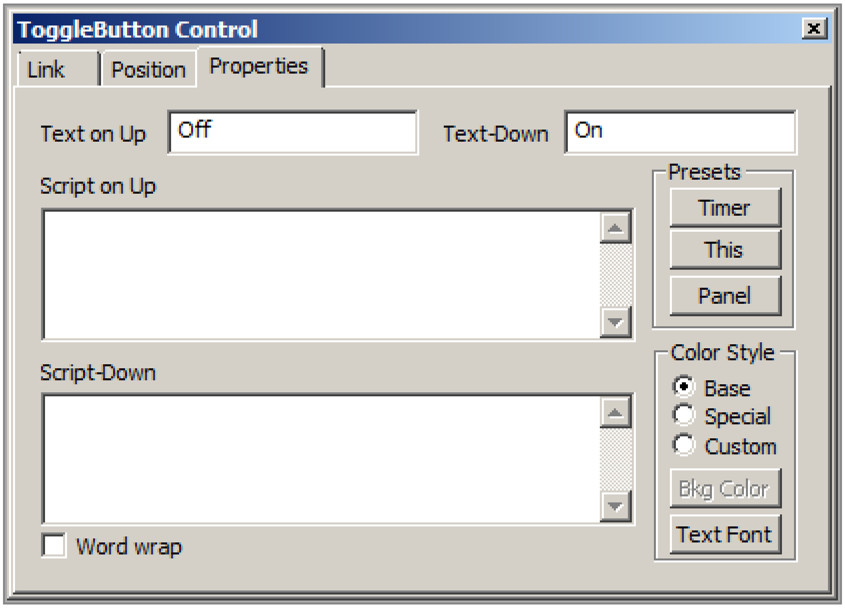
Color Style: ● Base ○ Special ○ Custom

Check Box control on Timer Event object

- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens the dialog for setting the background color.
- **Pen Color**: Opens the dialog for setting the pen color.

Toggle Button

The Toggle Button is a control that executes script commands any time the button is clicked, or can replace activity . The Timer Event object uses a checkbox by default to start or stop the Timer. This button has two states: released and pressed. Left-clicking the button will change its state, and one script is executed depending on the last state.



ToggleButton Control

Text on Up: Off

Text-Down: On

Script on Up:

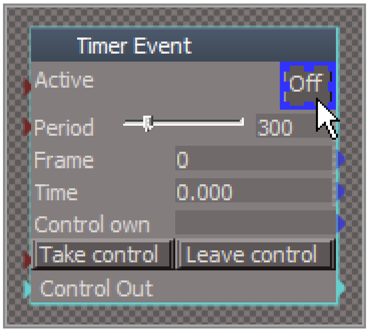
Script-Down:

□ Word wrap

Color Style: ● Base ○ Special ○ Custom

- **Text on Up**: Text that is displayed when the button is released.
- **Text-Down**: Text that is displayed when the button is pressed.
- **Script on Up**: Script that is executed when you click on the button while in its un-pressed (released) state.

- **Script-Down**: Script that is executed when you click on the button while in its pressed state.
- **Word wrap**: Automatically wrap text to more lines in script edit fields.
- **Timer**: Fills Script fields with commands that start and stop the system timer.
- **This** and **Panel**: Contain identifiers for this node and this panel.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color.
- **Text Font**: Opens a dialog for setting the font, size and color of the text.



Active Off
Period 300
Frame 0
Time 0.000
Control own

Toggle Button Example

Note: It is recommended that you link the toggle button with a boolean connector. This will help the control to remember its pressed or released state. (For example, try to activate the Living Circle demo, enter the Living Circle object in the LE, and then return back – the toggle button state becomes confused.)

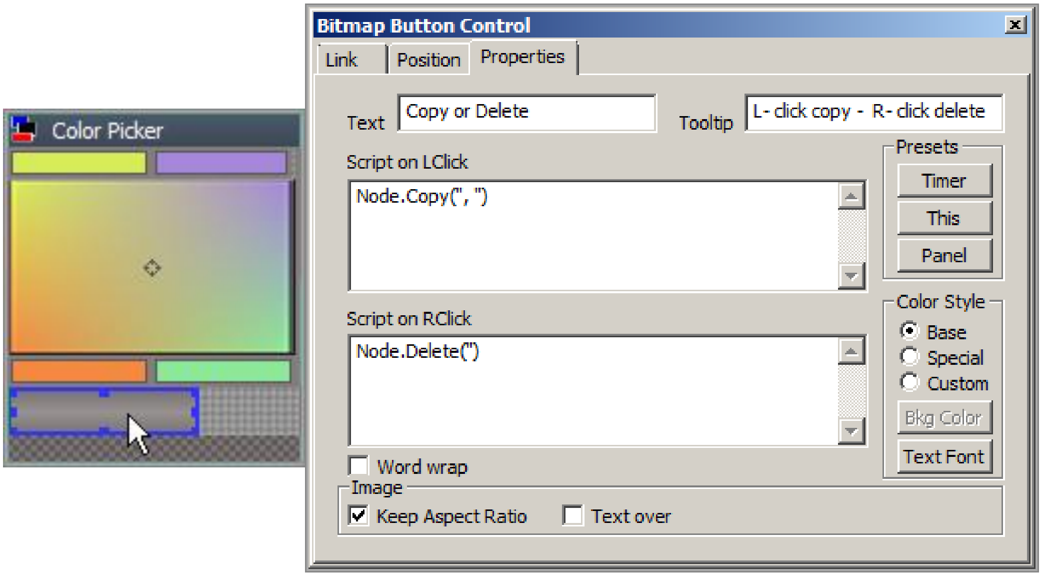
Bitmap Button Control

Use this to drive scripts or commands that can be executed on right -click or left click whilst over the button, it also allows for a tooltip to be shown when mouse is over it. Use a Bitmap to specify an image input for the control by Control-key + double Left-click on the control (when not in Panel Editor mode). You can choose an image from your hard drive.

Note: this control needs to be linked to an image type input attribute for it to allow the loading of an image to the button. If there is no attribute of a suitable type to use as a link for the button , add a new attribute, of type bitmap (common data) to your object to be able to load an image to the button.

The example below has the Toolbar commands for the copy and delete functions added to it.

The Commands are executed on the selected object in this case.



Text: Copy or Delete

Tooltip: L- click copy - R- click delete

Script on LClick:

```
Node.Copy(", ")
```

Script on RClick:

```
Node.Delete(")
```

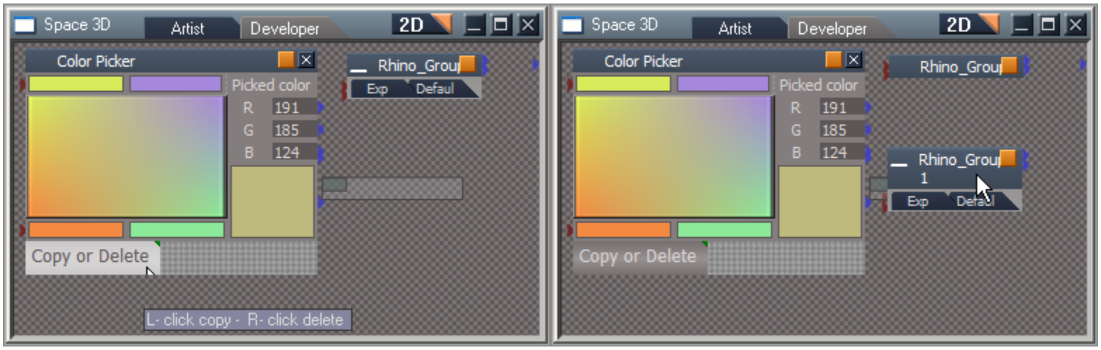
□ Word wrap

● Base ○ Special ○ Custom

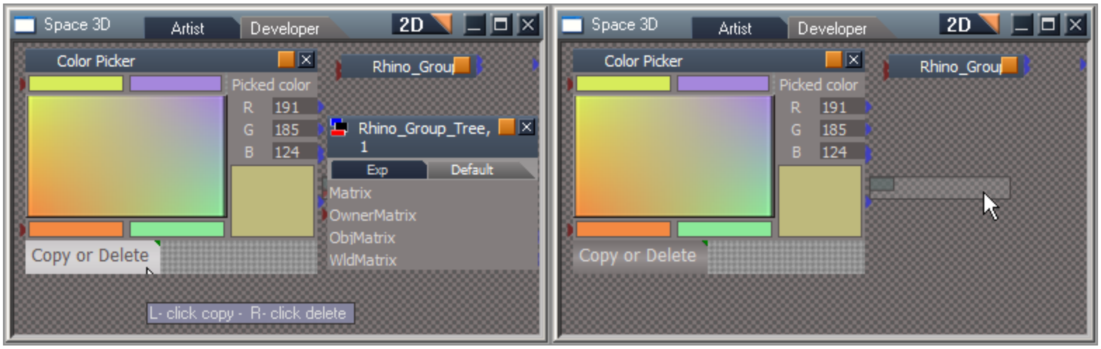
☑ Keep Aspect Ratio □ Text over

Bitmap Push Button

- **Text** : Text that is displayed on the button.
- **Tooltip**: Text that is displayed when the mouse is over the button.
- **Script on LClick**: Script that is executed when you L-click over the button.
- **Script on RClick**: Script that is executed when you R-click over the button.
- **Word wrap**: Automatically wrap text to more lines in script edit fields.
- **Timer**: Fills Script fields with commands that start and stop the system timer.
- **This** and **Panel**: Contain identifiers for this node and this panel.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color.
- **Text Font**: Opens a dialog for setting the font, size and color of the text.
- **Keep Aspect Ratio**: When checked, the displayed bitmap retains the aspect ratio of the image.
- **Text Over**: Checked the text shows over the image , unchecked the text is hidden.



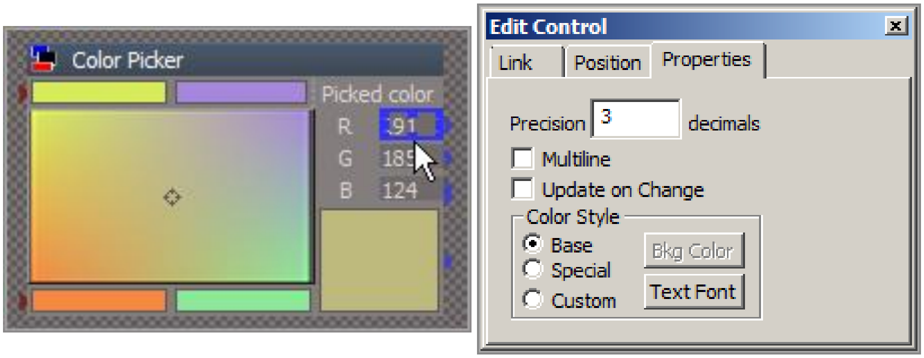
L-click over the control and command to Copy is executed on the selected object



R-click over the control and command to Delete is executed on the selected object

Abc Text and Number Field

Text and Number Fields allow you to type text or numbers to set the value of an attribute.



Edit Control

Link | Position | Properties

Precision 3 decimals

□ Multiline

□ Update on Change

Color Style

● Base
○ Special
○ Custom

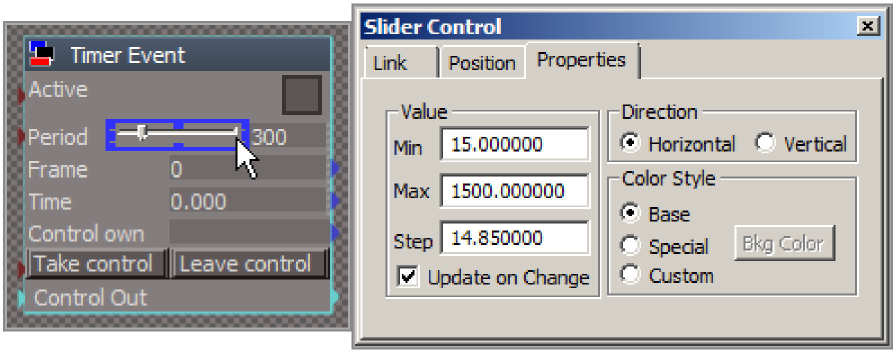
Bkg Color
Text Font

Text and Number Fields and the Edit Control Panel properties

- **Precision**: set decimal precision. How many decimal places show on form/field.
- **Multiline**: check this box if you require more than one line for display in the control.
- **Update on Change**: gives feedback when changed in form of update.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: this button opens up color picker to set background color of the control.
- **Text Font**: Opens a dialog for setting the font, size and color of text.

Slider

With a Slider you can change attribute values by clicking and dragging the slider’s knob.



Timer Event

Active □

Period 300

Frame 0

Time 0.000

Control own

Slider Control

Value

Min 15.000000

Max 1500.000000

Step 14.850000

☑ Update on Change

Direction

● Horizontal ○ Vertical

Color Style

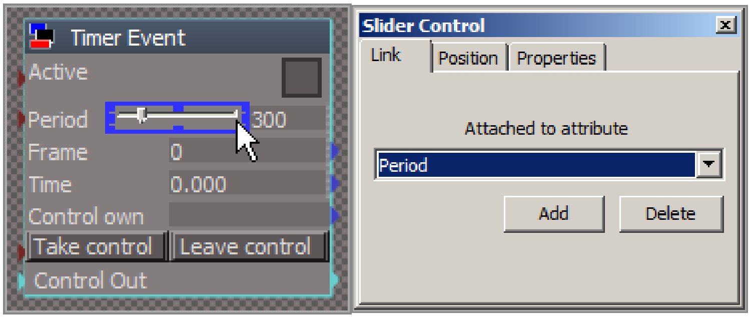
● Base

○ Special

○ Custom

Slider Control

- **Min Value**: minimum value for scrubber.
- **Max Value**: maximum value for scrubber.
- **Step Value**: each click on scrubber changes value by this amount.
- **Update on Change**: When checked scrubber updates values when control is dragged on, if unchecked then values update on mouse release.
- **Direction**: set scrubber for horizontal or vertical orientation.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color of the control.



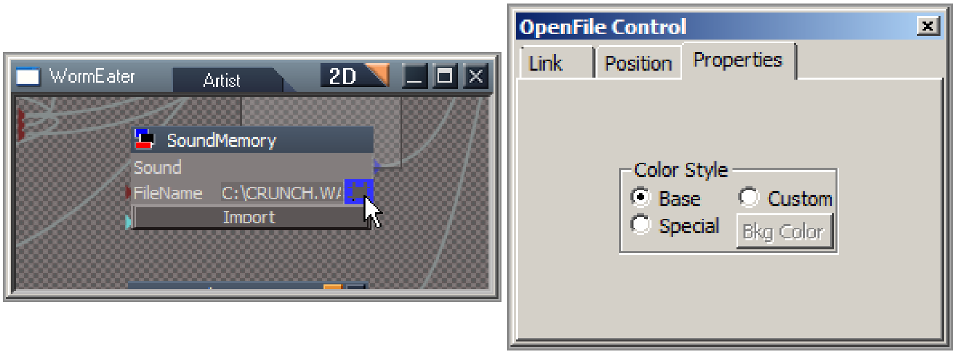
Period 300
Frame 0
Time 0.000
Control own

Attached to attribute
Period

Slider Control attached to attribute Period

Open File

Open File allows you to specify a file on disk as an attribute for the object. This loaded file can contain graphics, data, models, etc.



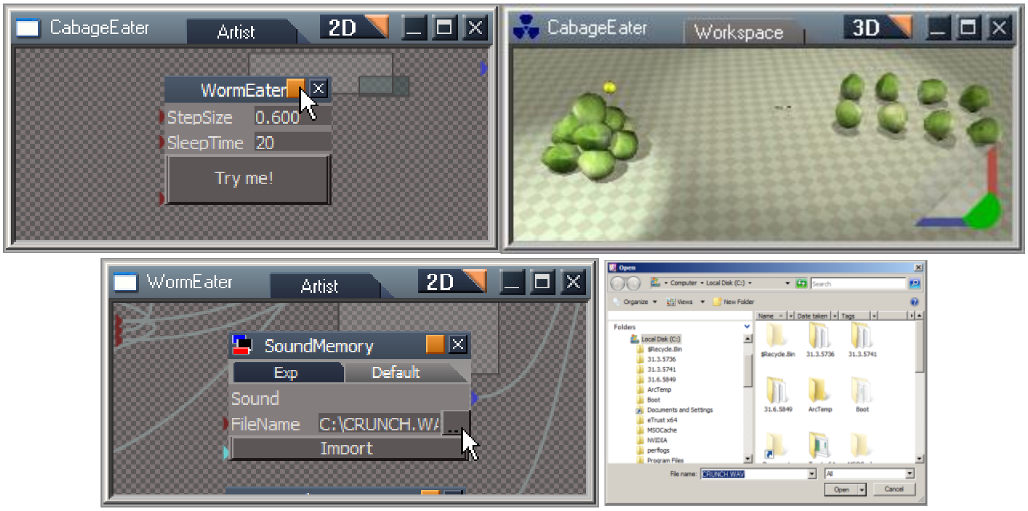
FileName C:\CRUNCH.WA

Color Style
● Base ○ Custom
○ Special

Open File

- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color.

The Cabage Eater scene has an example of load file , if you look inside the WormEater it uses the sound memory , where you can load a wav file using standard window dialog for the object then use the import button to "fix the wav" into the objects memory.

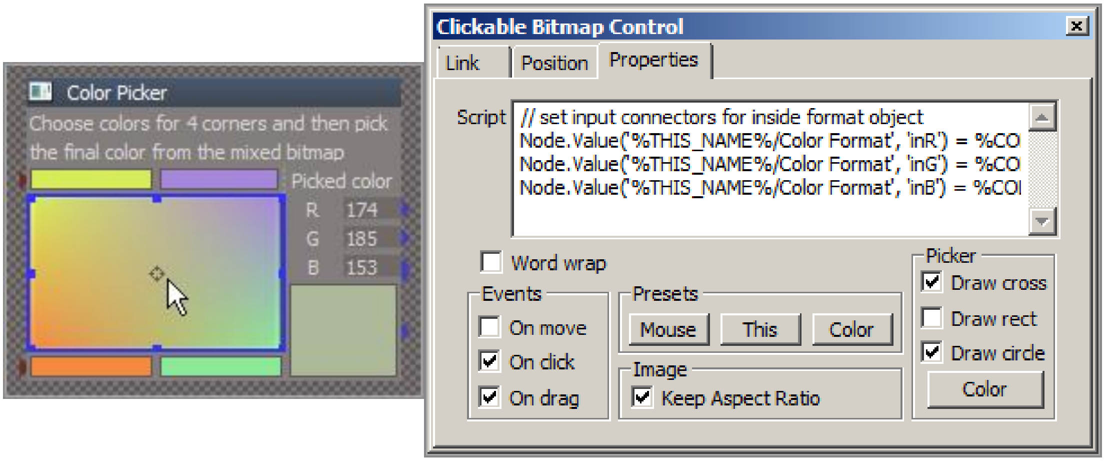


StepSize 0.600
SleepTime 20
FileName C:\CRUNCH.WA

Example of Open File in Scenes Base Library

Clickable Bitmap

Advanced version of the Bitmap control enhanced by mouse input. Mouse events, like moving the cursor over the bitmap and clicking on it are sent to user defined scripts.



Choose colors for 4 corners and then pick the final color from the mixed bitmap

R 174
G 185
B 153

Script
```
// set input connectors for inside format object
Node.Value('%THIS_NAME%/Color Format', 'inR') = %CO
Node.Value('%THIS_NAME%/Color Format', 'inG') = %CO
Node.Value('%THIS_NAME%/Color Format', 'inB') = %COI
```

□ Word wrap

Events: □ On move ☑ On click ☑ On drag

Image: ☑ Keep Aspect Ratio

Picker: ☑ Draw cross □ Draw rect ☑ Draw circle

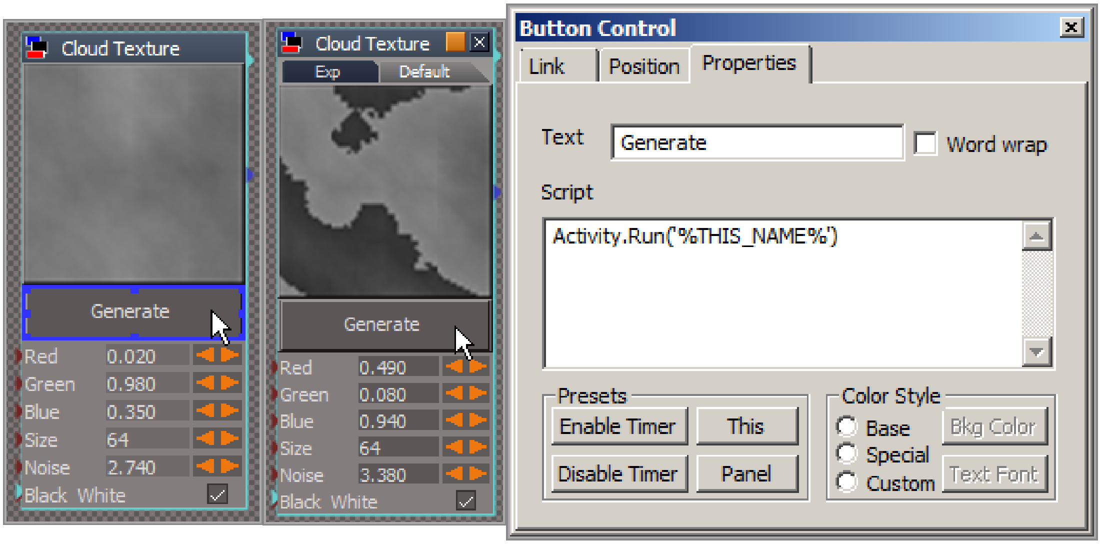
Clickable Bitmap Control Example - Objects-Script Object Library

- **Script**: Here you can handle mouse messages. Use presets to obtain mouse position and button status.
- **Events**: Select which events you want to handle.

- **Presets**: Select allows for preset scripts to populate the Script area rather than build scripts from scratch.
- **Picker**: select one or more cursor rendering styles.
- **Keep Aspect Ratio**: When checked, the displayed bitmap retains the aspect ratio of the image.

Push Button

The Push Button is a control that executes a script command any time the button is pressed. You will need to manually specify what command is executed.



Red	0.020	0.490
Green	0.980	0.080
Blue	0.350	0.940
Size	64	64
Noise	2.740	3.380
Black White	☑	☑

Text: Generate ☐ Word wrap

Script:

```
Activity.Run('%THIS_NAME%')
```

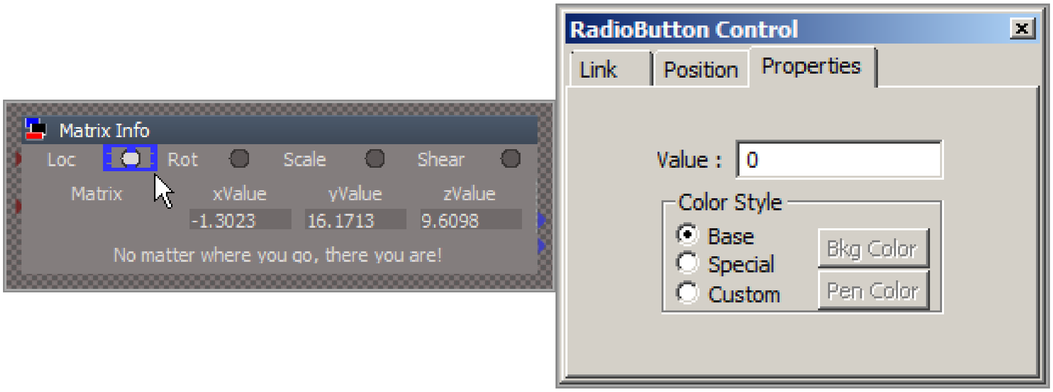
Example of Button Control from Objects-Script Object Library

- **Text**: Text to display on the button control.
- **Word wrap**: Automatically wrap text to more lines in the script editor.
- **Script**: The script that is executed when the button is pressed.
- **Enable Timer**: Fills the Script field with script that starts the timer.
- **Disable Timer**: Fills Script field with script that stops the timer.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color.

- **Text Font** : Opens dialog for setting the font, size and color of the text.
- **This** and **Panel**: Contain identifiers for this node and this panel.

Radio Button

Similar to the checkbox, the radio button has two states. The difference is that the radio button is checked when the input attribute has the same value as the radio button. This value can be set in control properties. This control can be accessed via scripting to group several Radio Buttons to work in tandem.



Matrix Info

Matrix xValue -1.3023 yValue 16.1713 zValue 9.6098

No matter where you go, there you are!

RadioButton Control

Value : 0

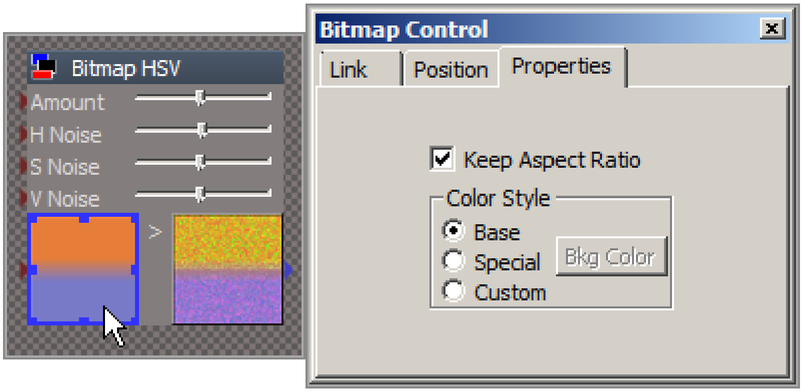
Color Style
● Base
○ Special
○ Custom

Matrix Info object in the Scripts-Objects library uses 4 Radio Buttons in a group configuration: only one Radio Button can be selected at any given time

- **Value**: This is set to the attached object attribute when you click the radio button. If the radio button is checked, then object attribute has the same value.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color.
- **Pen Color**: Opens a dialog for setting the pen color.

Bitmap

Use a Bitmap to specify an image input for a control by Control-key + Double Left-click on the control (when not in Panel Editor mode). You can choose an image from your hard drive(s).

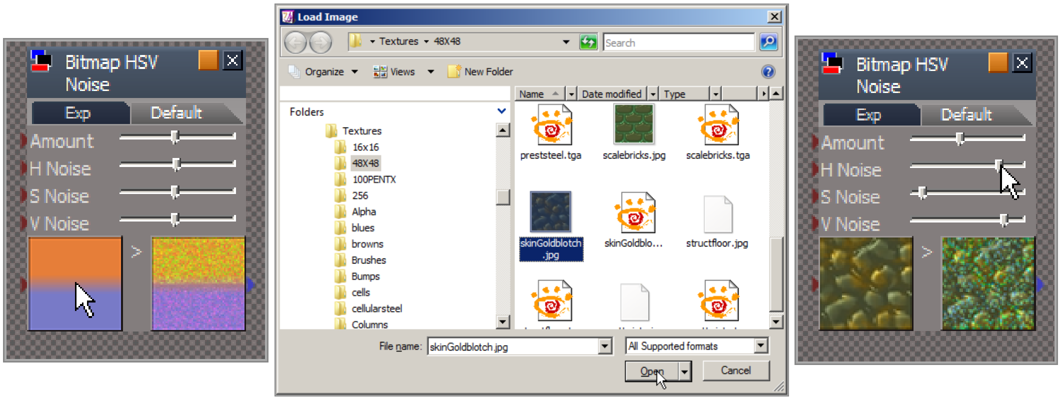


☑ Keep Aspect Ratio

Color Style
● Base
○ Special
○ Custom

Bitmap Controls

- **Keep Aspect Ratio**: When checked, the displayed bitmap retains the aspect ratio of the image.
- **Color Style Base**: use trueSpace default color scheme.
- **Color Style Special**: gives the selected field a lighter background color. Useful for title areas, or explanatory areas of the panel.
- **Color Style Custom**: activates the **Bkg Color** so a custom color can be used.
- **Bkg Color**: Opens a dialog for setting the background color of the control.
- **Load Image Dialog**: Control-key + left-click on the Bitmap control will open the Load Image dialog. You can load an image from your hard drive in this dialog.



File name: skinGoldblotch.jpg

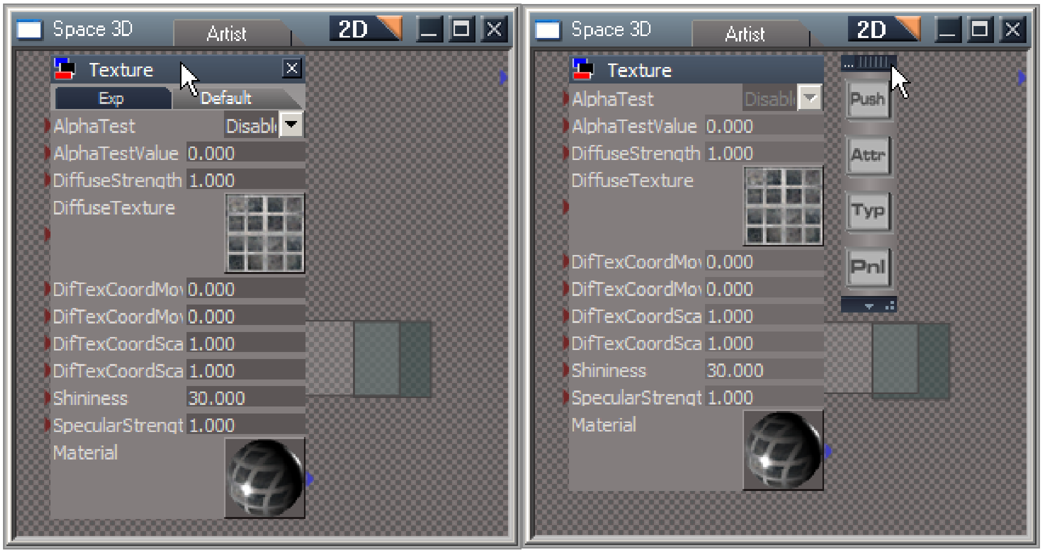
All Supported formats

Bitmap Controls use ctrl + left-click to load images from the hard drive

2.10.4 Tutorial: Editing a Simple Panel

In this tutorial you will learn how to edit the Default aspect of a trueSpace panel. Start by dragging a simple material, called “Texture” (located in the Materials – DX9 library) into the Link Editor and selecting the Default aspect.
On the left you see the automatically-generated default aspect of the simple material that we just loaded. Right-click the material’s title bar to switch to edit mode (right side of image).

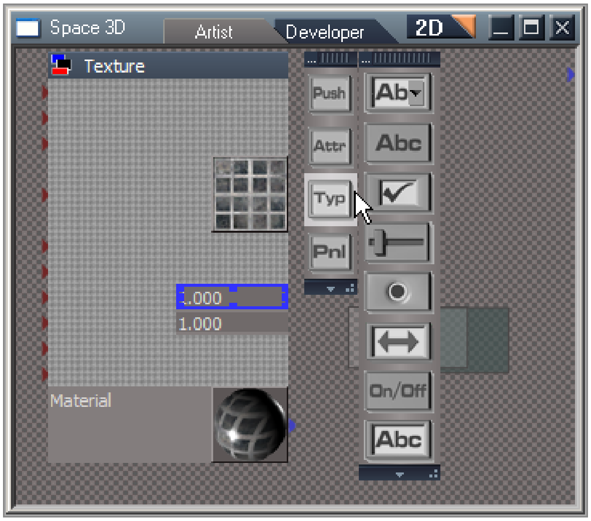
Entering Panel edit Mode creates the Main Edit Toolbar, which has additional controls and toolbars that can be accessed for adding/editing the panel. When editing a panel this Main Edit Toolbar always appears. Other toolbars are accessed from this toolbar.



AlphaTest Disabl
AlphaTestValue 0.000
DiffuseStrenqth 1.000
DiffuseTexture
DifTexCoordMo 0.000
DifTexCoordMo 0.000
DifTexCoordSca 1.000
DifTexCoordSca 1.000
Shininess 30.000
SpecularStrengt 1.000
Material

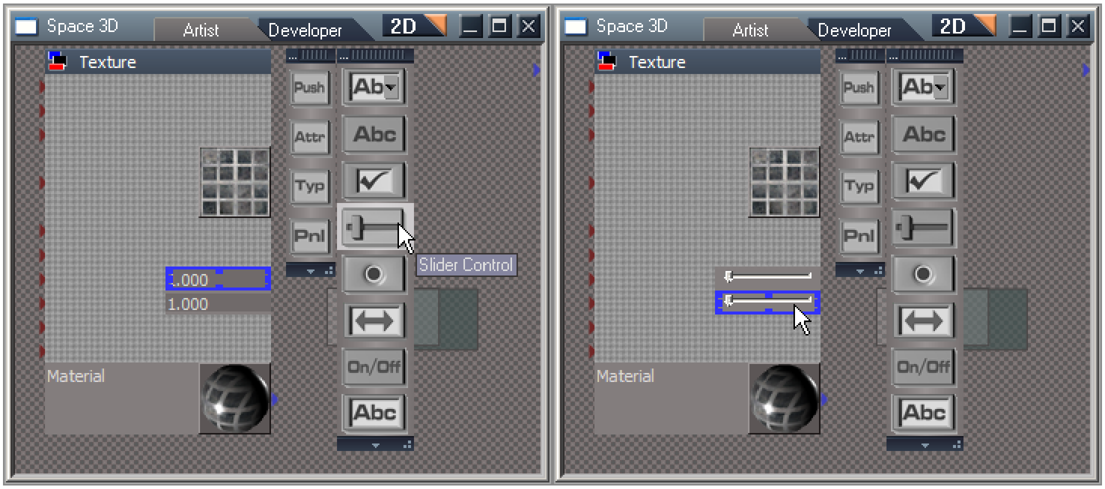
Preparing to edit the panel of a simple material R-Click title to show Panel edit tool bar

In this mode individual 2D controls can be selected , then moved around, scaled , etc.
First, select all unneeded controls, and eliminate them with the **DEL** . Then select the text edit control for the *DefTexCoordScaleX* attribute, and left-click on the Type library at the right to show all of the alternative controls for this attribute type. The Type toolbar is context sensitive in that it will show only the types of controls that are available for the selected control on an object’s panel.

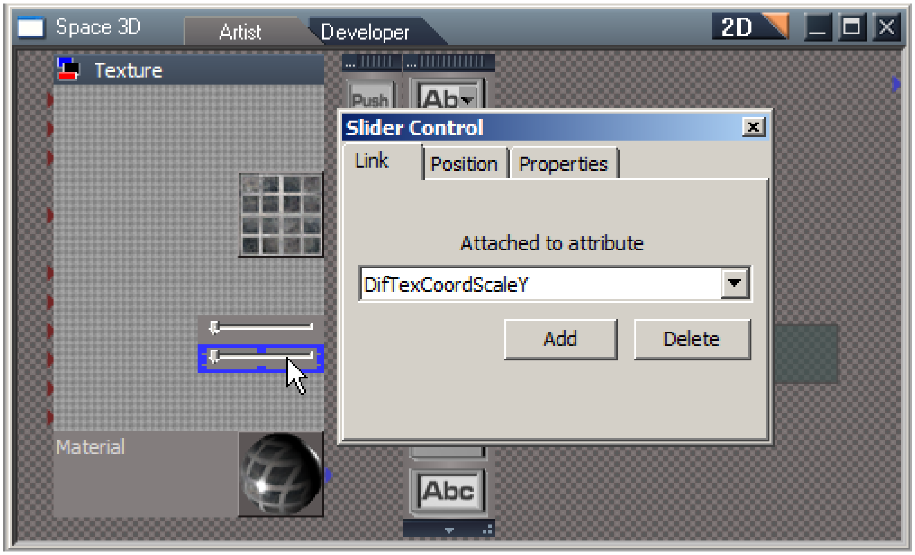


Deleting unneeded controls from a panel

Choose the Slider control and it immediately appears within the blue rectangle that highlights the selected control on the panel. Do the same for *DefTexCoordScaleY*.



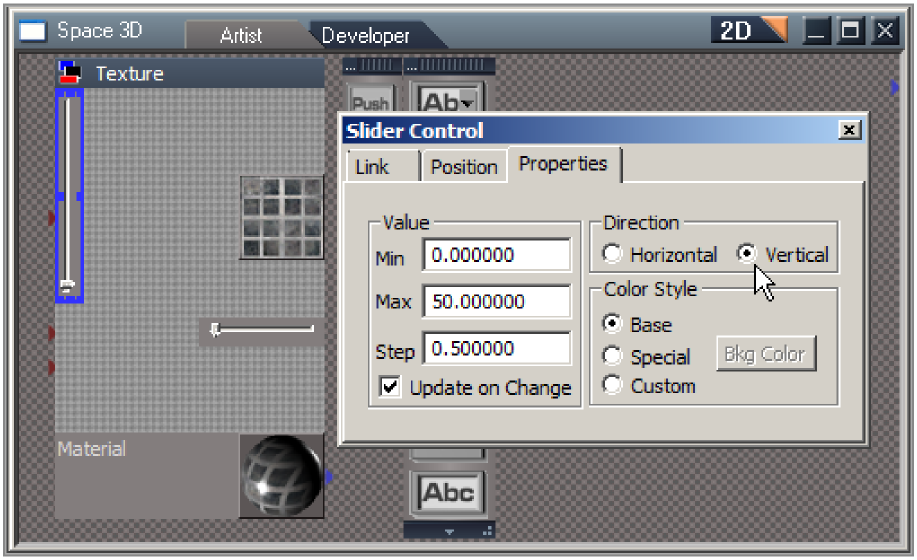
Now, we will change the slider for *DefTexCoordScaleY* to be oriented vertically instead of horizontally. Right-click the slider control to display its properties, and click the Properties tab.



Attached to attribute: DifTexCoordScaleY

R-Click to show properties

Once there click the Vertical option under Direction to change the orientation of the slider.



Value

Min 0.000000

Max 50.000000

Step 0.500000

☑ Update on Change

Direction

○ Horizontal ● Vertical

Color Style

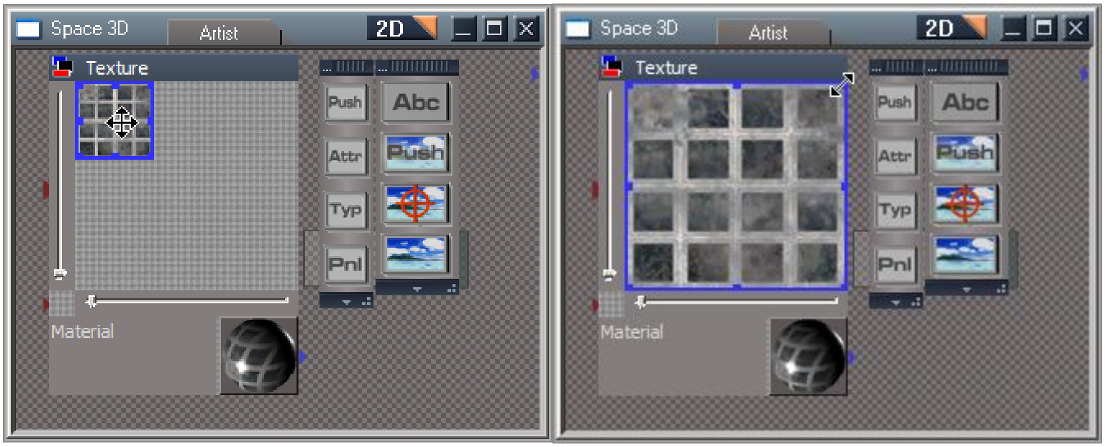
● Base

○ Special

○ Custom

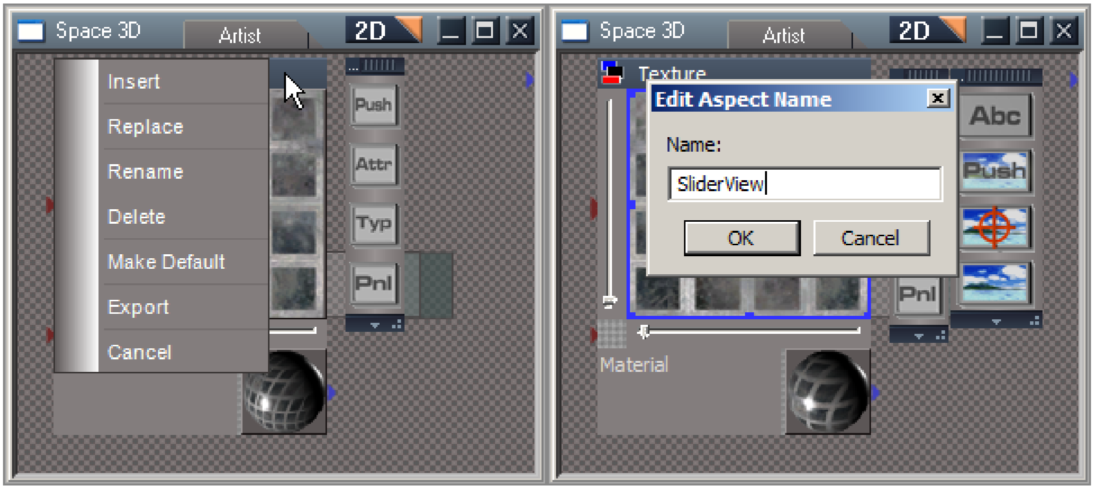
Change it to vertical and resize it by dragging the bottom handle

Use the blue selection rectangle to move and scale both sliders so that they appear as in the illustration. Select the Bitmap Control and move it to the upper left hand corner of the panel, near the top of the vertical slider. Now, scale the bitmap by dragging the scale handle on the blue selection rectangle until your layout is similar to the example.



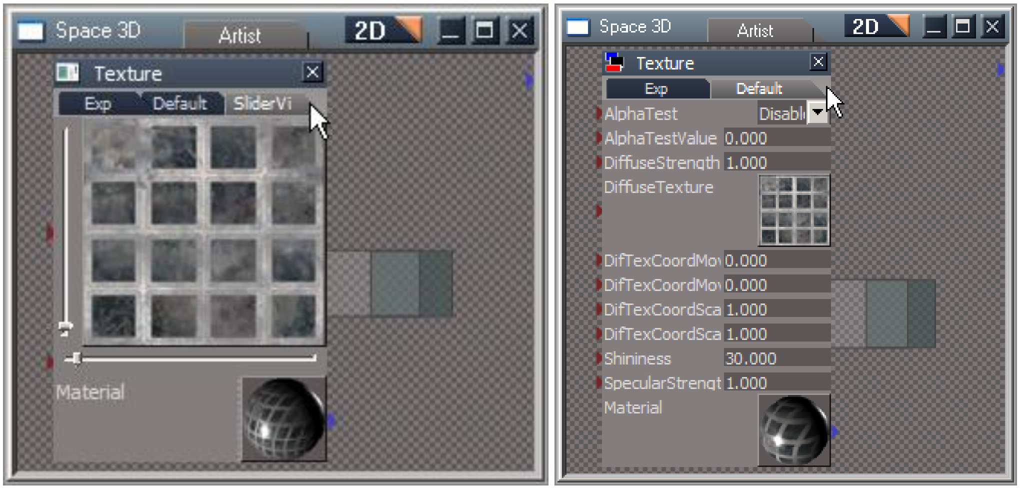
Adding and modifying user controls on a panel

In a few simple steps we have created a direct-manipulation 2D control. By aligning the sliders that control texture scale with the correct axes, the users of this material will now have a visual clue as to what the controls actually do, rather than rely on arcane attribute names like *DifTexCoordScaleX*.
To save your edited panel, just right-click again on the panel's title bar and select **Insert** to name your new aspect.



Name: SliderView

Insert and name the new aspect to save the changes



AlphaTest Disabl
AlphaTestValue 0.000
DiffuseStrength 1.000
DiffuseTexture
DifTexCoordMo 0.000
DifTexCoordMo 0.000
DifTexCoordSca 1.000
DifTexCoordSca 1.000
Shininess 30.000
SpecularStrengt 1.000
Material

Finished Panel with added aspect and original panel before adding new aspect

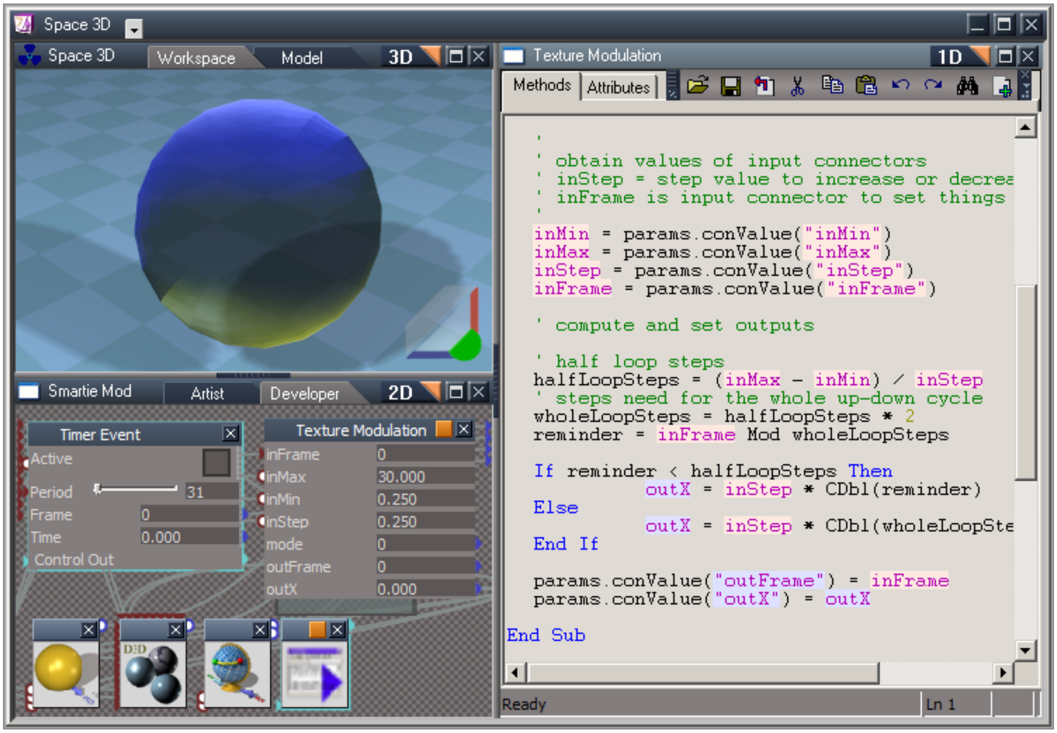
Now you can change the panel's aspect to this new aspect at any time. If you edit and select the **Replace** option, the new panel layout will replace the existing aspect with that name.

2.11 1D View – Script Editor

2.11.1 Script Editor Overview

Almost every object in trueSpace also has a structure which can be explored by entering the object. Once inside, you can look around using a 3D, 2D, or 1D view simply by changing the aspect of the opened window. The **3D Workspace** and **Model** views are great for creating and modifying the *visual* aspects of objects, while the **2D Link Editor** view is intended for building the internal structure and defining the way an object behaves. There is still another level, which allows for an increase in control over the way objects interact. We call this 1D view the **Script Editor**.

Not all objects expose their internal functioning in script form, but some do. In the image below you can see one such script-based object (called Smartie Mod) as it is represented in the Workspace view. You can also see inside of that same object in the 2D Link Editor. It contains an object called Texture Modulation, and when you enter that object, it automatically shows the 1D or script aspect (right-side image below), which you can edit using the Script Editor.



Timer Event

- Active
- Period 31
- Frame 0
- Time 0.000
- Control Out

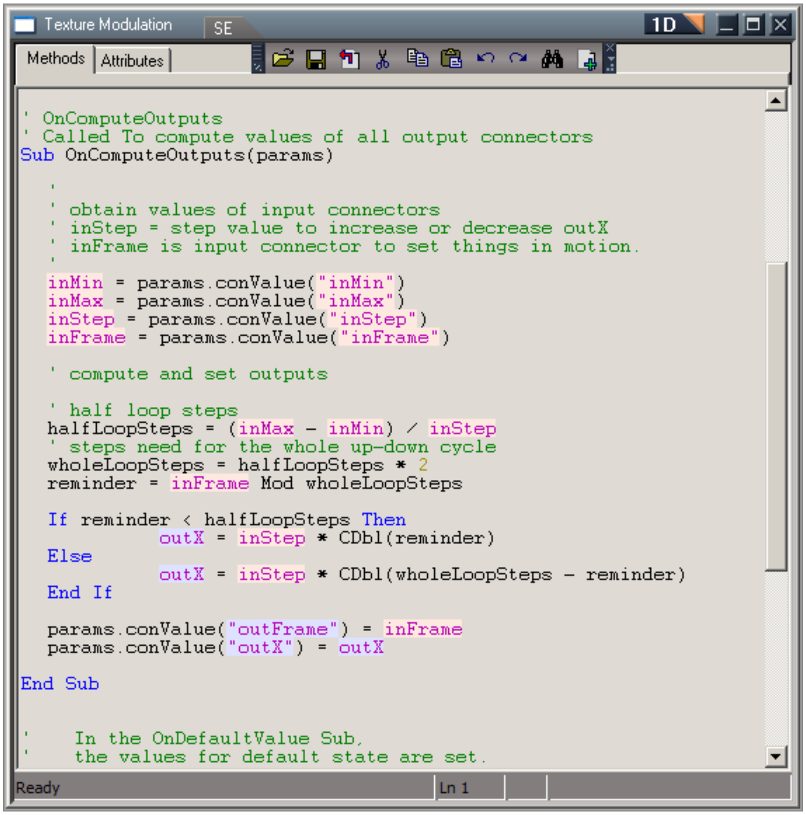
Texture Modulation

- inFrame 0
- inMax 30.000
- inMin 0.250
- inStep 0.250
- mode 0
- outFrame 0
- outX 0.000

```
'
' obtain values of input connectors
' inStep = step value to increase or decrea
' inFrame is input connector to set things
'
inMin = params.conValue("inMin")
inMax = params.conValue("inMax")
inStep = params.conValue("inStep")
inFrame = params.conValue("inFrame")

' compute and set outputs

' half loop steps
halfLoopSteps = (inMax - inMin) / inStep
' steps need for the whole up-down cycle
wholeLoopSteps = halfLoopSteps * 2
reminder = inFrame Mod wholeLoopSteps

If reminder < halfLoopSteps Then
          outX = inStep * CDbl(reminder)
Else
          outX = inStep * CDbl(wholeLoopSte
End If

params.conValue("outFrame") = inFrame
params.conValue("outX") = outX

End Sub
```

The Smartie Mod object shown in 3D, 2D, and 1D Views

The Script Editor allows you to create and edit scripts. Scripts are sets of instructions, written in a scripting language such as **VBScript.** With scripts, you can create and modify the behavior of objects in trueSpace. You can create scripts that define new models parametrically, modify existing models, produce animation and movement, cause objects to interact in some ways and much more.



```
' OnComputeOutputs
' Called To compute values of all output connectors
Sub OnComputeOutputs(params)

   '
   ' obtain values of input connectors
   ' inStep = step value to increase or decrease outX
   ' inFrame is input connector to set things in motion.
   '
   inMin = params.conValue("inMin")
   inMax = params.conValue("inMax")
   inStep = params.conValue("inStep")
   inFrame = params.conValue("inFrame")

   ' compute and set outputs

   ' half loop steps
   halfLoopSteps = (inMax - inMin) / inStep
   ' steps need for the whole up-down cycle
   wholeLoopSteps = halfLoopSteps * 2
   reminder = inFrame Mod wholeLoopSteps

   If reminder < halfLoopSteps Then
            outX = inStep * CDbl(reminder)
   Else
            outX = inStep * CDbl(wholeLoopSteps - reminder)
   End If

   params.conValue("outFrame") = inFrame
   params.conValue("outX") = outX

End Sub


'    In the OnDefaultValue Sub,
'    the values for default state are set.
```

The Script Editor (1D View)

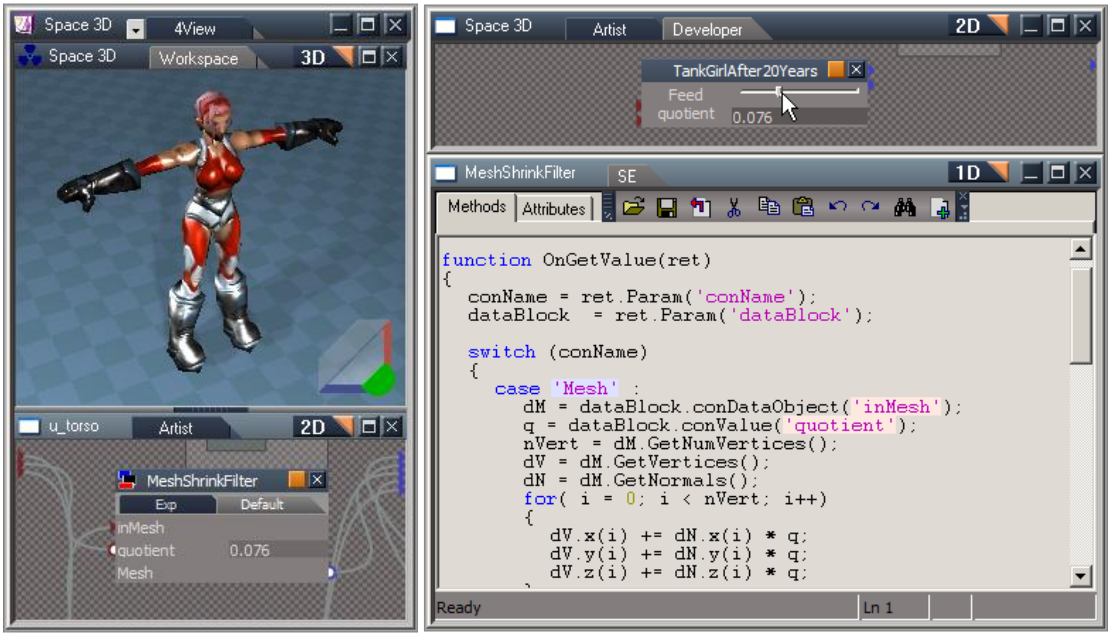
Script creation is not difficult, but it *is* technically programming. To create scripts you will have to research a scripting language and become familiar with it. You will also need to learn how the trueSpace system and other objects interact with the scripts you create.

Script Languages and Types

You have the ability to write your scripts using the **jScript** and **VBScript** languages. You can find more information on using these languages in trueSpace scripts, including where to find additional language-specific documentation, in the **DEVELOPER'S GUIDE.**

Scripting in trueSpace applies primarily to two types of script-based objects: **Script Commands** and **Script Objects**. Scripting can also be used to define shaders. Both types are viewable and editable using the Script Editor.

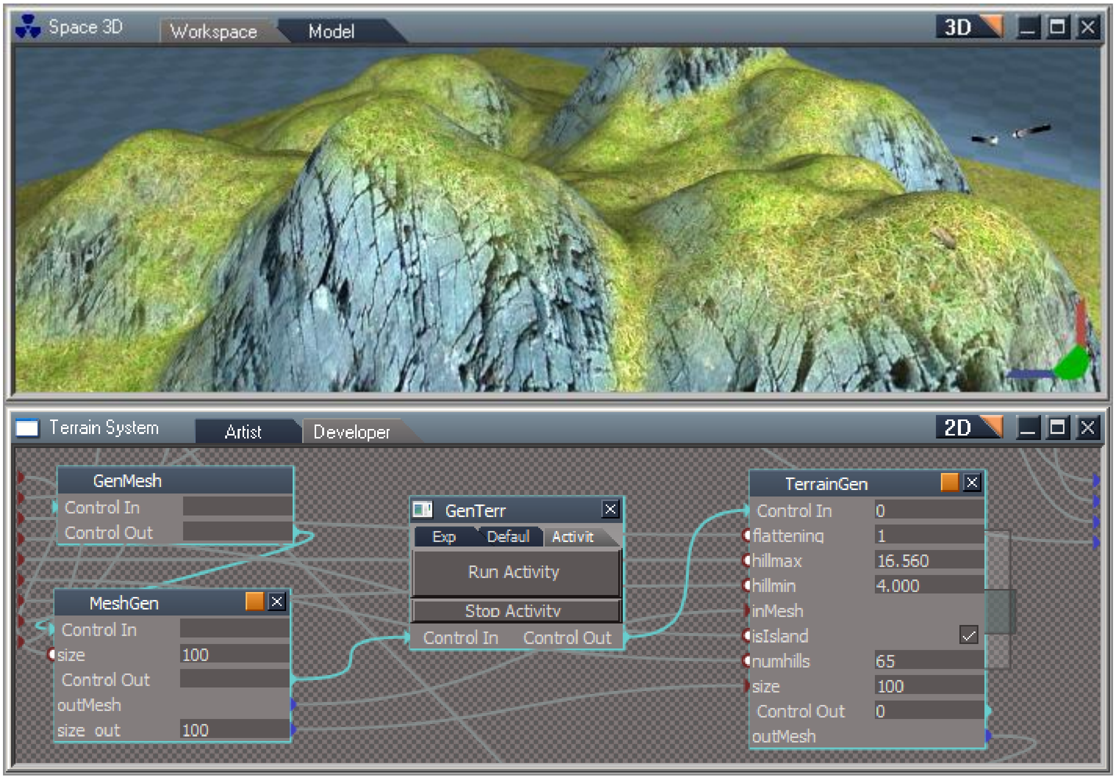
Script Objects are objects that interact continuously with other objects in trueSpace. You can use a Script Object for just about any purpose, such as generating procedural geometry, deforming models, performing mathematical operations, creating procedural animation and motion, etc. You can use a Script Object to react to various events in almost any way you can imagine. You can also link your Script Objects with other objects in the Link Editor to create even more complex interactions. Nested inside the Tank Girl after 20 years object we find a scenario where mesh is modified by using a slider control. The illustration below shows a MeshShrinkFilter used for the feed /quotient values.



```
function OnGetValue(ret)
{
   conName = ret.Param('conName');
   dataBlock  = ret.Param('dataBlock');

   switch (conName)
   {
     case 'Mesh' :
       dM = dataBlock.conDataObject('inMesh');
       q = dataBlock.conValue('quotient');
       nVert = dM.GetNumVertices();
       dV = dM.GetVertices();
       dN = dM.GetNormals();
       for( i = 0; i < nVert; i++)
       {
         dV.x(i) += dN.x(i) * q;
         dV.y(i) += dN.y(i) * q;
         dV.z(i) += dN.z(i) * q;
```

A Script Object used to modify a mesh

Script Commands are the basic building blocks of activities, responding to requests from the user and other objects in the Link Editor to perform whatever action you specify. Unlike script objects, which react to events and requests for data on a constant basis, script commands must be specifically requested to perform their action. This is accomplished by attaching the script command to the Control flow in the Link Editor. The Terrain System object utilizes several command script objects, all connected by special control (green) links.



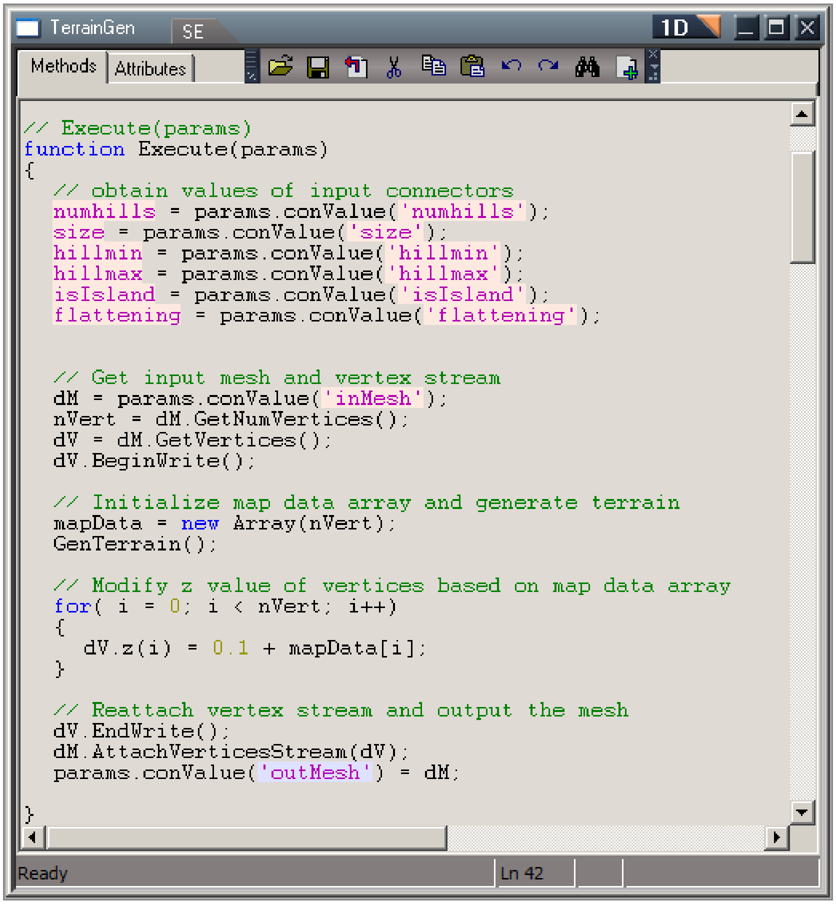
MeshGen:
Control In
size 100
Control Out
outMesh
size_out 100

TerrainGen:
Control In 0
flattening 1
hillmax 16.560
hillmin 4.000
inMesh
isIsland ☑
numhills 65
size 100
Control Out 0
outMesh

Script Commands in the Link Editor

2.11.2 Script Editor Workflow

Access to the Script Editor is done by entering a Script Object or Command in the Link Editor (using the orange enter button, found in the upper right-hand corner of the object panel). If the object is script-based, its script will be displayed in the Script Editor as shown below.

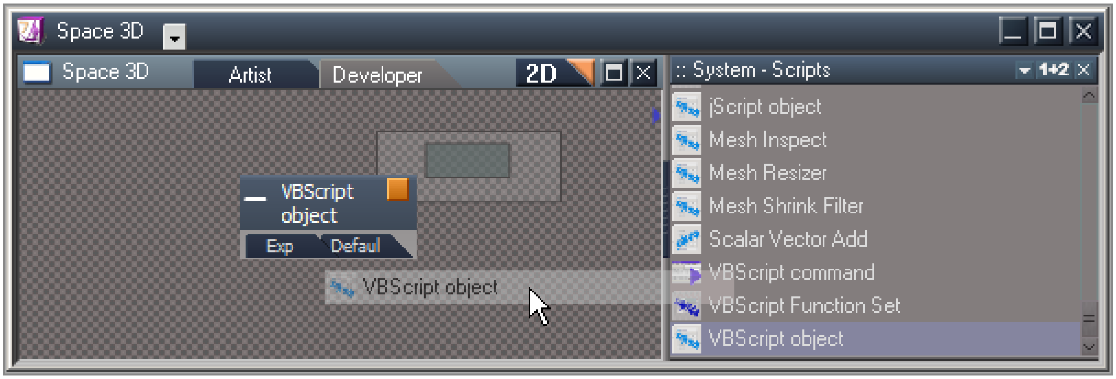


```
// Execute(params)
function Execute(params)
{
   // obtain values of input connectors
   numhills = params.conValue('numhills');
   size = params.conValue('size');
   hillmin = params.conValue('hillmin');
   hillmax = params.conValue('hillmax');
   isIsland = params.conValue('isIsland');
   flattening = params.conValue('flattening');


   // Get input mesh and vertex stream
   dM = params.conValue('inMesh');
   nVert = dM.GetNumVertices();
   dV = dM.GetVertices();
   dV.BeginWrite();

   // Initialize map data array and generate terrain
   mapData = new Array(nVert);
   GenTerrain();

   // Modify z value of vertices based on map data array
   for( i = 0; i < nVert; i++)
   {
      dV.z(i) = 0.1 + mapData[i];
   }

   // Reattach vertex stream and output the mesh
   dV.EndWrite();
   dM.AttachVerticesStream(dV);
   params.conValue('outMesh') = dM;

}
```

A Script Command viewed in the Script Editor

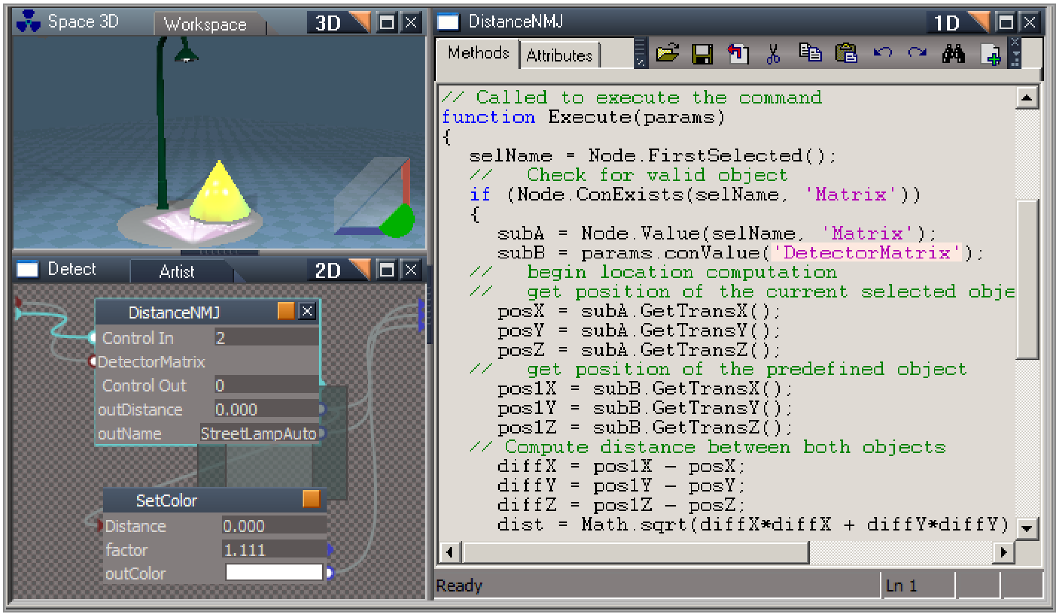
All scripting activities in trueSpace begin with a script-based object. To create a script-based object:

- Open the System library, select Scripts, and drag a jScript/VBScript Object or jScript/VBScript Command into the Link Editor. This creates an empty, generic script object and provides a basic template that you can use to create your script.



Dragging a VBScript Object into the Link Editor

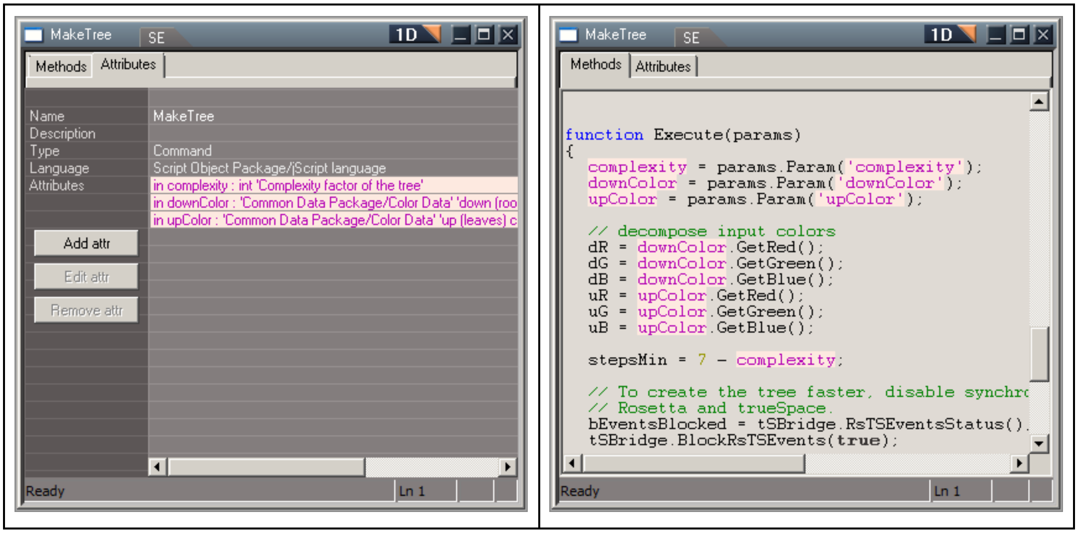
- Alternatively, you can also use an existing script-based object from one of the libraries as a starting point, which you can then edit. For an example, drag the StreetLampAuto object from the Activities - My Activities library into the Link Editor and examine its contents and script.



```
// Called to execute the command
function Execute(params)
{
  selName = Node.FirstSelected();
  //   Check for valid object
  if (Node.ConExists(selName, 'Matrix'))
  {
    subA = Node.Value(selName, 'Matrix');
    subB = params.conValue('DetectorMatrix');
  //   begin location computation
  //   get position of the current selected obje
    posX = subA.GetTransX();
    posY = subA.GetTransY();
    posZ = subA.GetTransZ();
  //   get position of the predefined object
    pos1X = subB.GetTransX();
    pos1Y = subB.GetTransY();
    pos1Z = subB.GetTransZ();
  // Compute distance between both objects
    diffX = pos1X - posX;
    diffY = pos1Y - posY;
    diffZ = pos1Z - posZ;
    dist = Math.sqrt(diffX*diffX + diffY*diffY)
```

The Script Object found in the StreetLampAuto object

Now that you have created (or copied) a script-based object, you can edit it as desired. To edit the object simply enter it by clicking on the orange enter button. This will open the Script Editor and allow you to edit the object's attributes and the text of its script. If it is a new object, you would begin by defining Attributes in the Script Attributes view (see below), and then add your specific commands to the template script pre-generated by trueSpace.

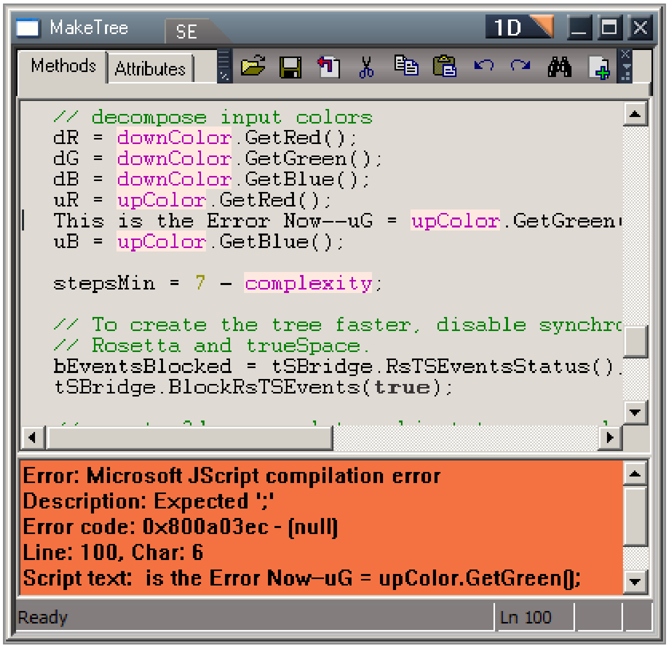


Name MakeTree
Description
Type Command
Language Script Object Package/jScript language
Attributes in complexity : int 'Complexity factor of the tree'
in downColor : 'Common Data Package/Color Data' 'down (roo
in upColor : 'Common Data Package/Color Data' 'up (leaves) c

```
function Execute(params)
{
   complexity = params.Param('complexity');
   downColor = params.Param('downColor');
   upColor = params.Param('upColor');

   // decompose input colors
   dR = downColor.GetRed();
   dG = downColor.GetGreen();
   dB = downColor.GetBlue();
   uR = upColor.GetRed();
   uG = upColor.GetGreen();
   uB = upColor.GetBlue();

   stepsMin = 7 - complexity;

   // To create the tree faster, disable synchro
   // Rosetta and trueSpace.
   bEventsBlocked = tSBridge.RsTSEventsStatus().
   tSBridge.BlockRsTSEvents(true);
```

Script Attributes view (left) and Script Methods view (right)

When you have edited a script and want to see the results of your changes, you should click the Commit button on the Script Editor toolbar. This will check the script for errors, and if there are none, the system will begin using your updated script. If errors are detected, then you will be notified of the problem and given some idea of its location and type within your script.



```
// decompose input colors
dR = downColor.GetRed();
dG = downColor.GetGreen();
dB = downColor.GetBlue();
uR = upColor.GetRed();
This is the Error Now--uG = upColor.GetGreen(
uB = upColor.GetBlue();

stepsMin = 7 - complexity;

// To create the tree faster, disable synchro
// Rosetta and trueSpace.
bEventsBlocked = tSBridge.RsTSEventsStatus().
tSBridge.BlockRsTSEvents(true);
```

```
Error: Microsoft JScript compilation error
Description: Expected ';'
Error code: 0x800a03ec - (null)
Line: 100, Char: 6
Script text:  is the Error Now--uG = upColor.GetGreen();
```

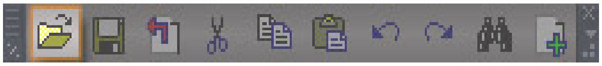
Ready | Ln 100

Click the Commit button to update your script and check for errors

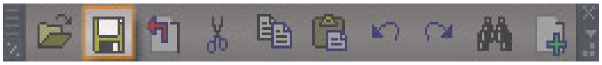
Once you have completed your scripting tasks, you can exit the Script Editor by clicking the Go Back button (the orange triangle near the upper-right of the Script Editor window).

2.11.3 Script Editor Interface

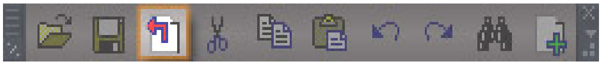
The trueSpace Script Editor has two primary views: the **Script Attributes** view, where you will create and modify your script's attributes, and the **Script Methods** view, where you will write and edit your script. The following toolbar buttons are available in both views and provide commonly used functions when working with scripts.



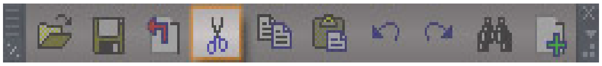
Import Script: Imports script text from an external file.



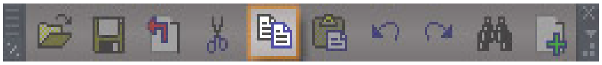
Export Script: Export script text to an external file.



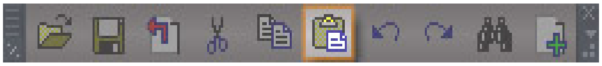
Commit: Performs a syntax check and updates the object.



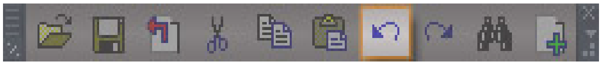
Cut: Deletes highlighted text **and** stores it on the clipboard.



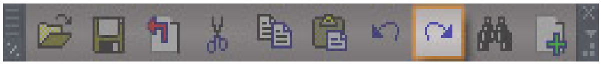
Copy: Copies highlighted text to the clipboard.



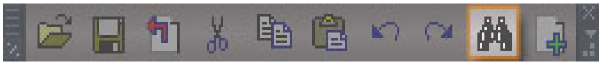
Paste: Pastes text from the clipboard into your script.



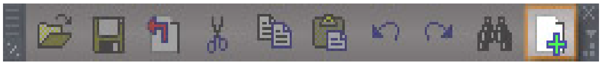
Undo: Undoes the last action.



Redo: Restores a previously undone action.



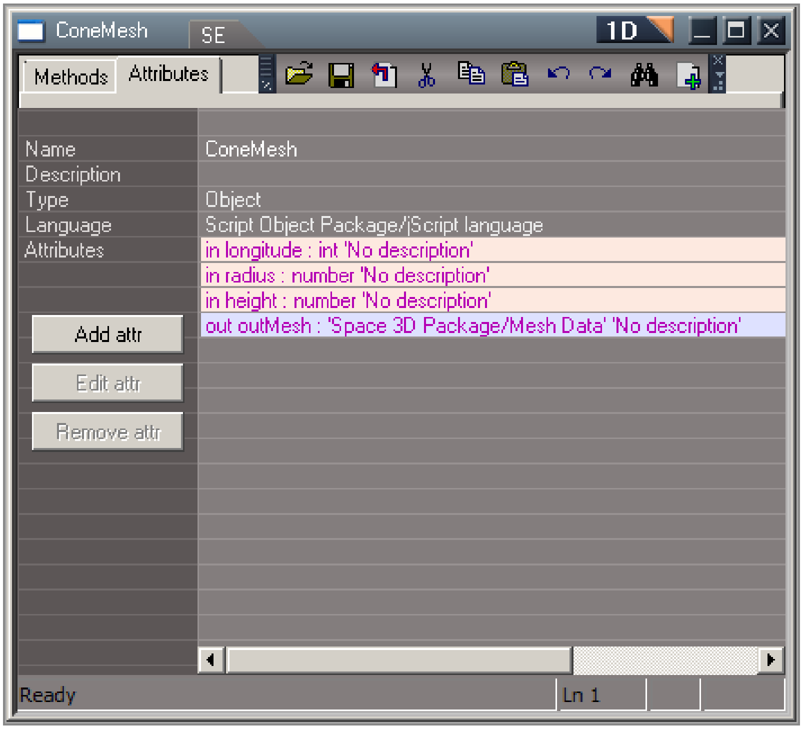
Find: Searches for text in your script.



Add advanced handlers: Add advanced methods handlers to your script. A basic set of handlers are generated by default, but you can add additional handlers for reacting to changes in the shared space, receiving custom notifications, changing attribute dependencies, etc.

Script Attributes View

Select the Attributes tab in the Script Editor to show the Script Attributes view. This is where you will create the input and output parameters that will connect your script to other objects. When creating a new script-based object, you should start here to define the object's attributes before you begin to create your script.



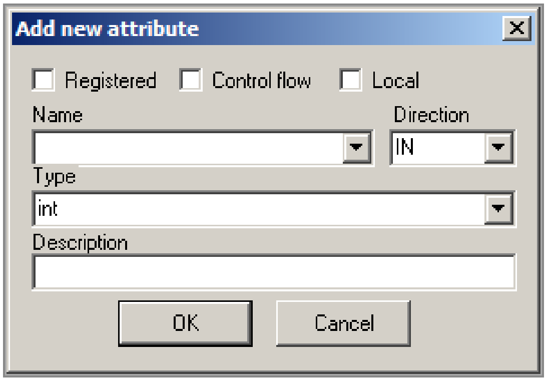
Name	ConeMesh
Description	
Type	Object
Language	Script Object Package/jScript language
Attributes	in longitude : int 'No description'
	in radius : number 'No description'
	in height : number 'No description'
	out outMesh : 'Space 3D Package/Mesh Data' 'No description'

The Script Attributes view

You can change "Name" and "Description" to fit your object's purpose. You can also add, edit, and remove attributes using the buttons at the left of the view.

Adding attributes

Press the **Add attr** button to add a new attribute to your object. This will open the Add New Attribute dialog box to fill-in a name for the attribute, choose its type and direction, and enter a description. Check 'registered' to pick an existing registered attribute (like mesh or matrix). Check ”control” to make this attribute part of the control flow. Check "Local" to ensure this attribute is available locally only.



Add new attribute

☐ Registered ☐ Control flow ☐ Local

Name

Direction
IN

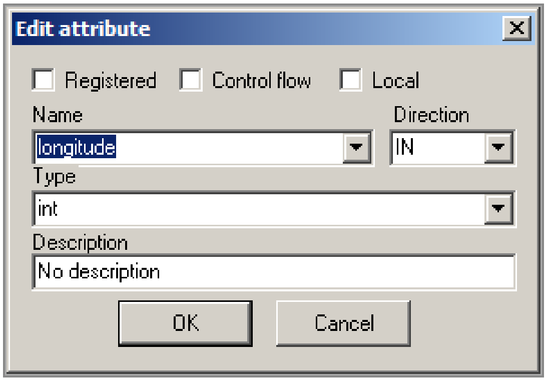
Type
int

Description

Add New Attribute dialog box

Editing Attributes:

Select an existing attribute from the list and press the **Edit attr** button to edit the details of that attribute. Remember that, if you edit an attribute, you may need to edit your script accordingly. This is especially true if you change the attribute's direction or type.



Edit attribute

☐ Registered ☐ Control flow ☐ Local

Name
longitude

Direction
IN

Type
int

Description
No description

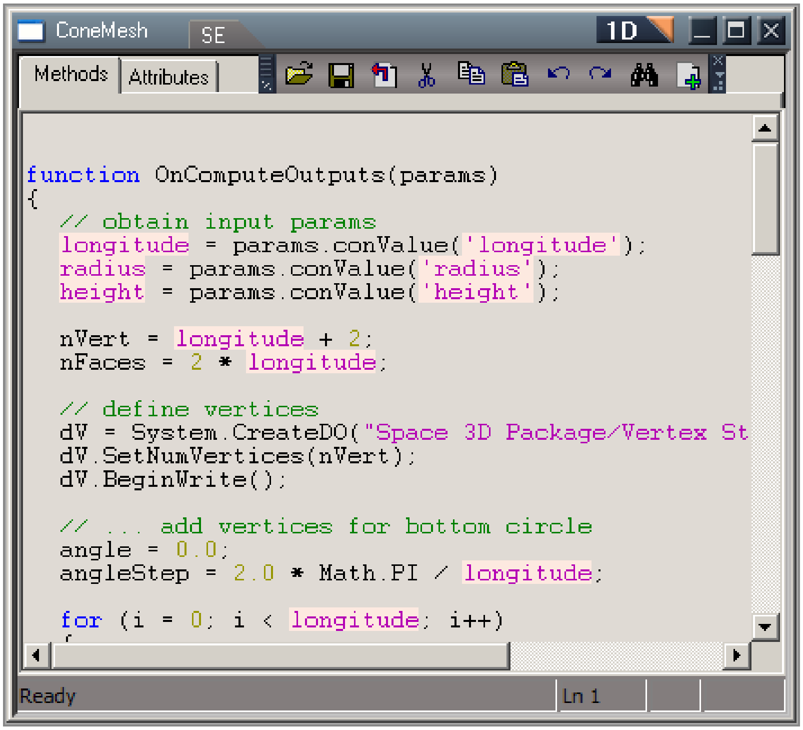
Edit existing attributes using Edit attribute panel

Removing Attributes:

Select an existing attribute from the list and press the **Remove attr** button to delete the attribute. Remember to ensure your script no longer references the deleted attribute!

Script Method view

Select the Methods tab in the Script Editor to show the Script Methods view. This is where you will write and modify the script itself. When creating a new script, this will contain a pre-generated template based on the type of object and attributes you have already defined in the Attributes view. This is why it is usually best to define an object's attributes before writing its script.



```
function OnComputeOutputs(params)
{
   // obtain input params
   longitude = params.conValue('longitude');
   radius = params.conValue('radius');
   height = params.conValue('height');

   nVert = longitude + 2;
   nFaces = 2 * longitude;

   // define vertices
   dV = System.CreateDO("Space 3D Package/Vertex St
   dV.SetNumVertices(nVert);
   dV.BeginWrite();

   // ... add vertices for bottom circle
   angle = 0.0;
   angleStep = 2.0 * Math.PI / longitude;

   for (i = 0; i < longitude; i++)
```

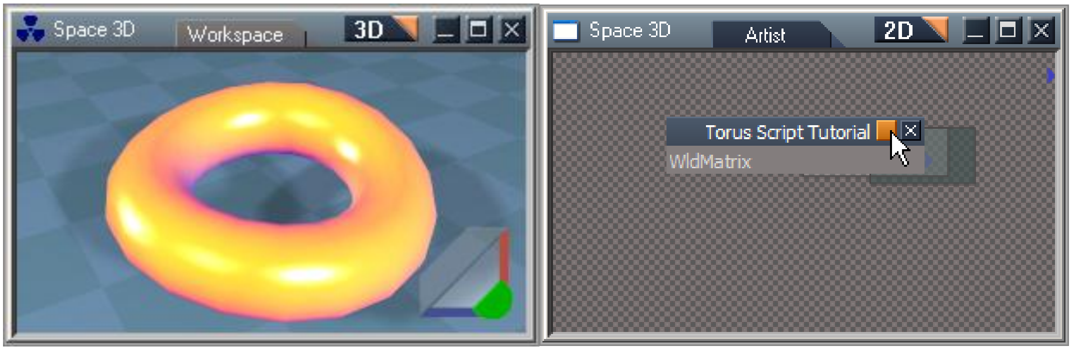
The Script Methods view

Tutorial: Modifying a Script

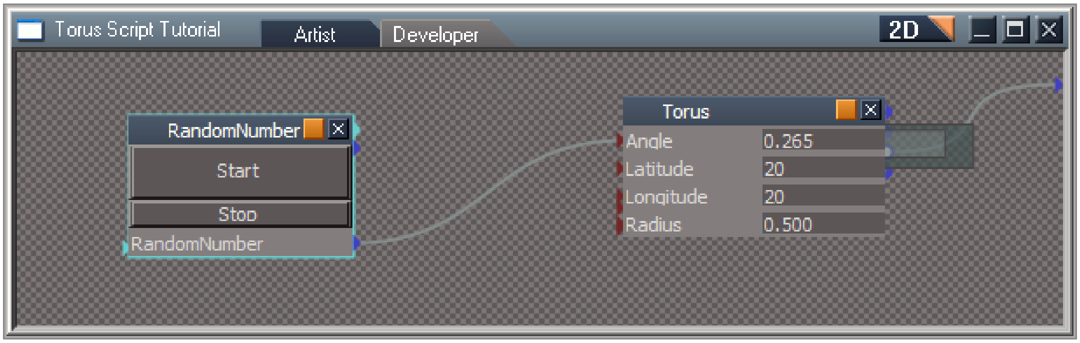
The best way to get started working with scripts in trueSpace is to modify an existing script and watch the results. In this example you will change the way the torus, in the Torus Script Tutorial object, is modified by altering its script.

Open the **Objects-Tutorial Objects** library and double-left-click on the **Torus Script Tutorial** object to load it into trueSpace.

Enter the Torus Script Tutorial object by clicking on the orange enter button, on the upper-right side of the object's panel.



The Torus Script Tutorial object as it appears in Workspace and the Link Editor's Space 3D scene level

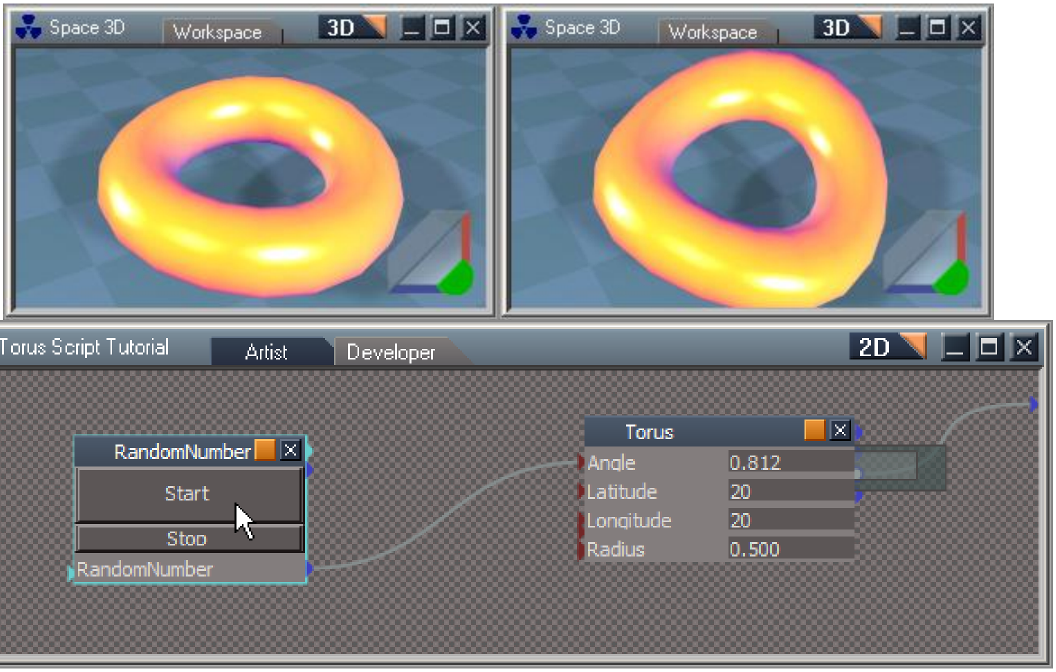


Angle 0.265
Latitude 20
Longitude 20
Radius 0.500

The Link Editor view inside Torus Script Tutorial object

The Torus Script Tutorial object actually encapsulates/contains two sub-objects. One of these is called Torus and it is responsible for creating the torus mesh/material that is seen in the Workspace view. The Radius, Longitude, Latitude and the Angle can be directly controlled by typing values into the text fields on the object's front panel.

Angle, is connected to the second object, a jScript Command called Random Number. This is the object which we will modify. Before we advance further, it is important for you to set the script in motion for the first time. You instantiate by clicking on the RandomNumber's Start button. A random number is sent across the link to the Torus' *Angle* attribute. You should see a change in the shape of the torus object. Click the Start button a few times and watch the results. Usually the torus will change by just a small amount. Review the values for *Angle* on Torus to better understand how it works.

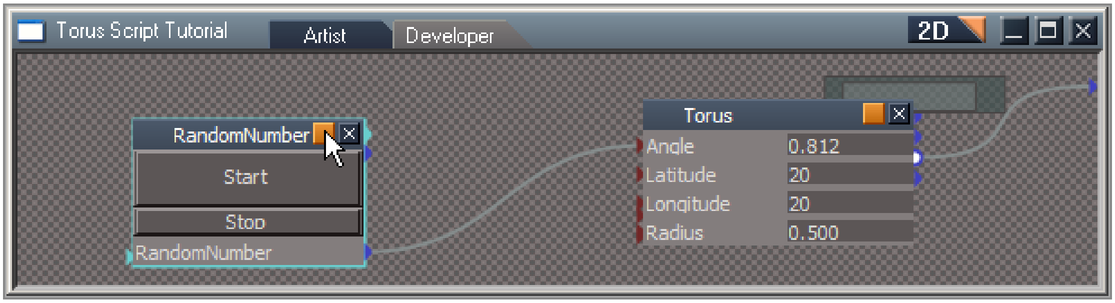


RandomNumber

Angle 0.812
Latitude 20
Longitude 20
Radius 0.500

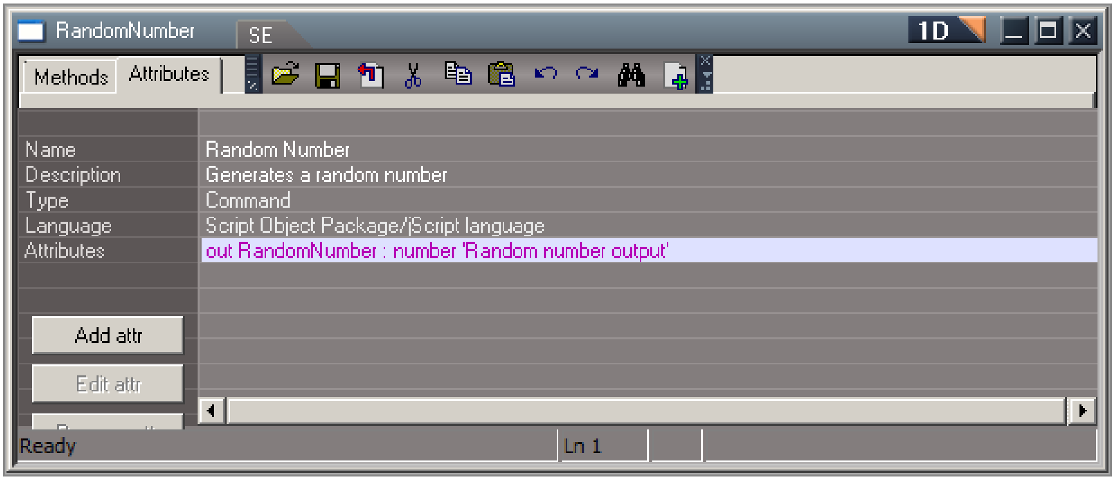
Clicking Start bends the torus randomly

Enter RandomNumber by clicking on the orange triangle on its panel. This will open the Script Editor. Click on the Attributes tab to see the Script Attributes view. *RandomNumber* is shown as an output attribute. This connects to *Angle* on the Torus object, which controls how much the torus bends.



RandomNumber

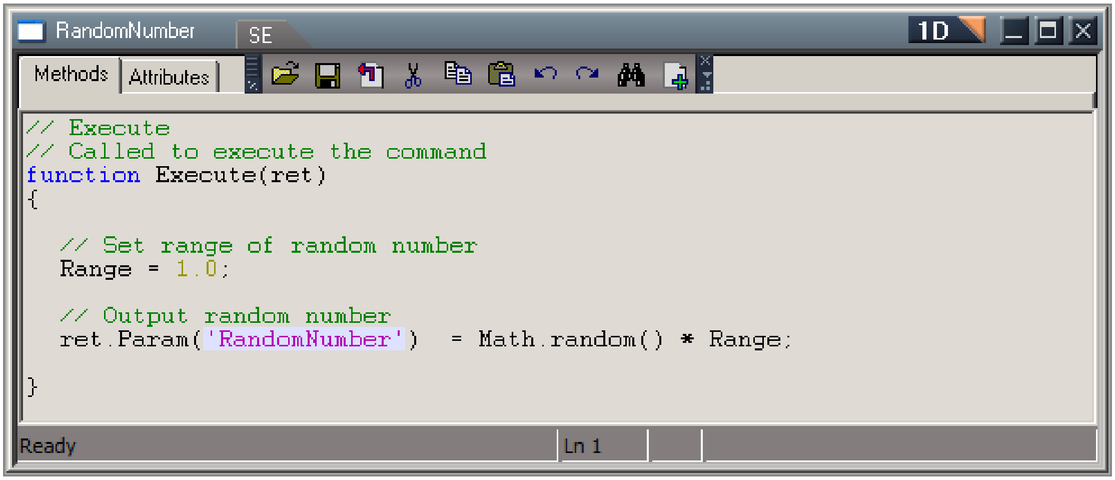
Angle 0.812
Latitude 20
Longitude 20
Radius 0.500



Name	Random Number
Description	Generates a random number
Type	Command
Language	Script Object Package/jScript language
Attributes	out RandomNumber : number 'Random number output'

RandomNumber's attributes as seen in the Script Attributes view

Click the Methods tab to open the Script Methods view. This will give you a view of the script in RandomNumber.



```
// Execute
// Called to execute the command
function Execute(ret)
{

   // Set range of random number
   Range = 1.0;

   // Output random number
   ret.Param('RandomNumber')  = Math.random() * Range;

}
```

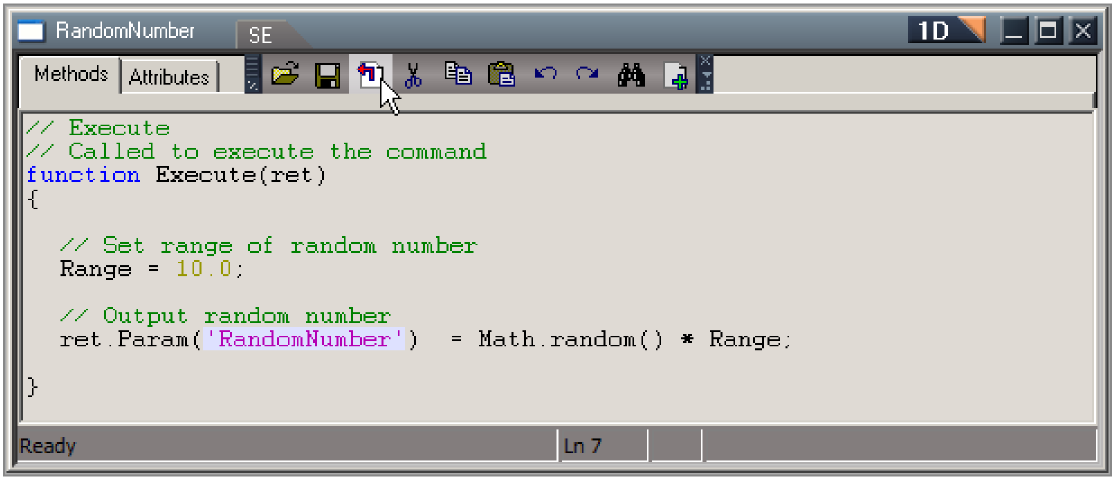
The RandomNumber script in the Script Methods view

There is one function contained in this script. The Execute method, is called each time you press the Start button on the object's outer panel. The purpose of this function is to generate a random number and then pass that out through the *RandomNumber* output attribute, to whatever object(s) are attached to it. Here is the function:

```
function Execute(ret)
{
   // Set range of random numbers
   Range = 1.0;
   // Output random number
   ret.Param('RandomNumber') = Math.random() * Range;
}
```

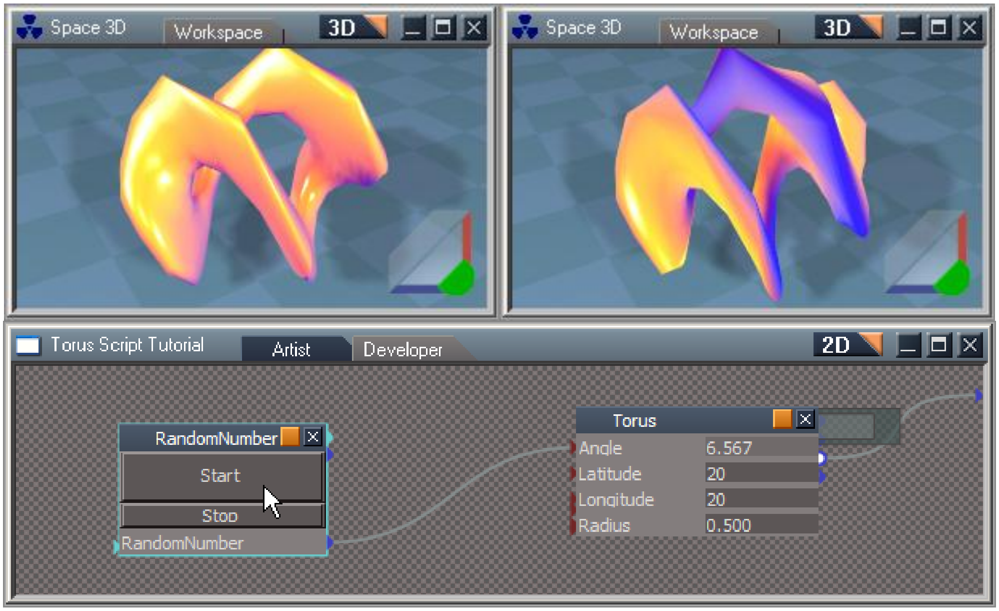
The pre-defined JScript function Math.random() creates a random number in the range of 0 and 1. This value is multiplied by the variable Range to expand that range of random values. By changing Range to 10, you will get random values that range from 1 to 10. Change the value of Range by highlighting ”1.0” and typing ”10.0.” then commit the changes.

```
// Set range of random numbers
Range = 10.0;
```



```
// Execute
// Called to execute the command
function Execute(ret)
{

   // Set range of random number
   Range = 10.0;

   // Output random number
   ret.Param('RandomNumber')  = Math.random() * Range;

}
```

Click the orange triangle to exit out of the Script Editor, and press the Start button on RandomNumber a few more times. You should see something like the image below. The torus has the potential to be much more ”twisted” than before because you have expanded the range of random numbers that are available, for delivery, to the *Angle* attribute.

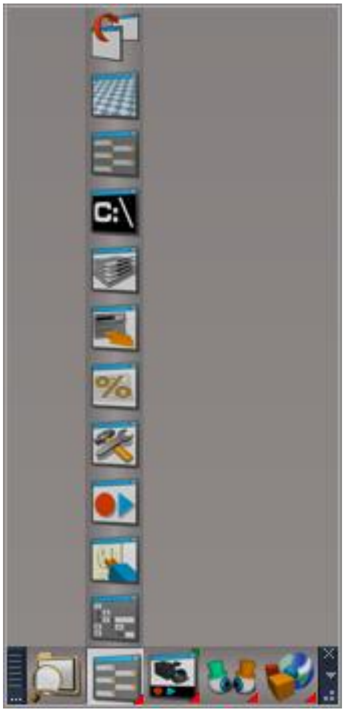


Angle 6.567
Latitude 20
Longitude 20
Radius 0.500

Now the torus is even more ”twisted”

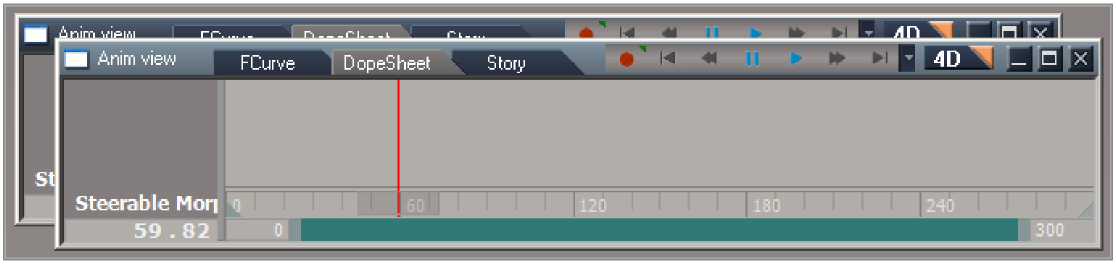
Try changing the range even further to see what happens. If you break the script simply get a new one from the library and try again.

2.12 Other Views

 ***Add New Windows***	Make copy of Window
	3D Window
	Link Editor
	Command Prompt
	Command History
	Output Console
	Status View
	Stack View
	Anim View
	Package Manager
	Scene View

Make copy of Window

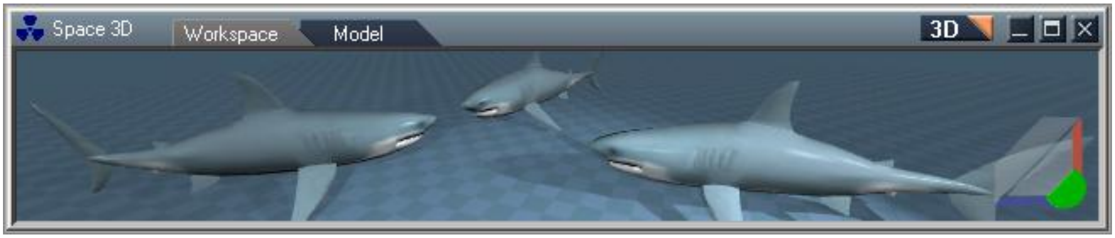
To make a copy of the current selected/active window, use this tool. Copy of a stack window is disabled for this tool.



Duplicated Anim Views

3D View

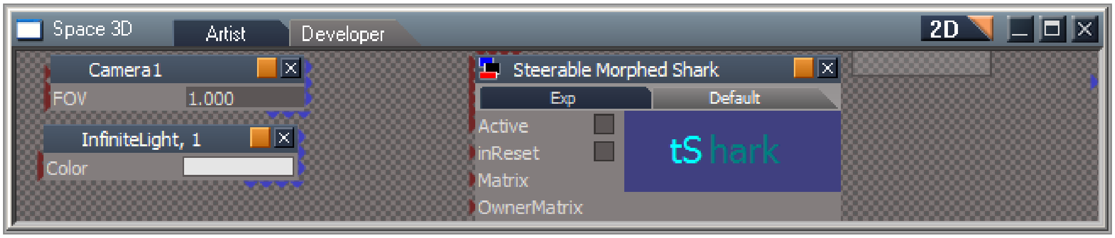
The 3D View is also referred to as 3D or WorkSpace. Most of your work is either created or viewed from this window.



3D View

Link Editor View

The Link Editor view is primary area for exploring structure of trueSpace. You are able to enter and explore some objects, to reveal additional levels of structure. You also interact with objects programmatically from within the Link Editor by setting values or connecting attributes.



LinkEditor View

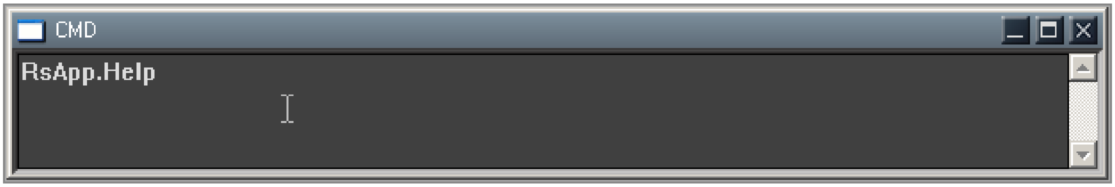
Command Prompt

The Command Prompt view allows you to directly input commands. Each command starts with a namespace (**Space**, **System**, **WinMng**, and so on), followed by a separator dot and a command.

Example command : which generates reference documents from within trueSpace.

1. Open up a Command window and type in.
2. RsApp.Help
3. press Return on the keyboard.

Executing this command generates 2 files called RsCmdDoc.xml , RsNodeDoc.xml in the \trueSpace7.*\tS folder, these can be opened in internet explorer or another viewer to read their contents and provide a useful source of object reference, the descriptions are also covered in the developers documentation and the SDK.



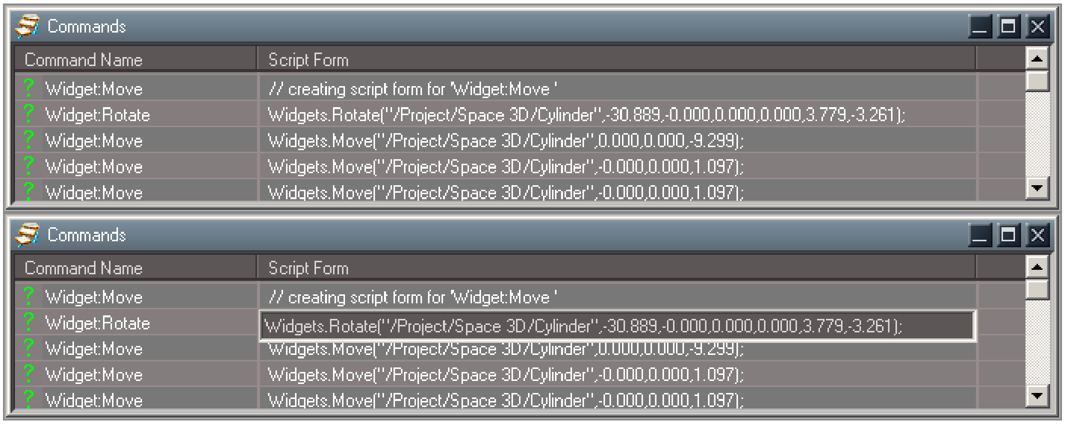
```
RsApp.Help
```

The Command Prompt view

Please consult the reference guide for more information on the namespaces and commands available to you via the Command Prompt view.

Command history

The Command history view displays commands that occur in trueSpace. These commands may originate from one of your custom script objects or perhaps from another script object at work in trueSpace. Double clicking in the command line and pressing enter will re-execute the command



Commands

Command Name	Script Form
Widget:Move	// creating script form for 'Widget:Move '
Widget:Rotate	Widgets.Rotate("/Project/Space 3D/Cylinder",-30.889,-0.000,0.000,0.000,3.779,-3.261);
Widget:Move	Widgets.Move("/Project/Space 3D/Cylinder",0.000,0.000,-9.299);
Widget:Move	Widgets.Move("/Project/Space 3D/Cylinder",-0.000,0.000,1.097);
Widget:Move	Widgets.Move("/Project/Space 3D/Cylinder",-0.000,0.000,1.097);

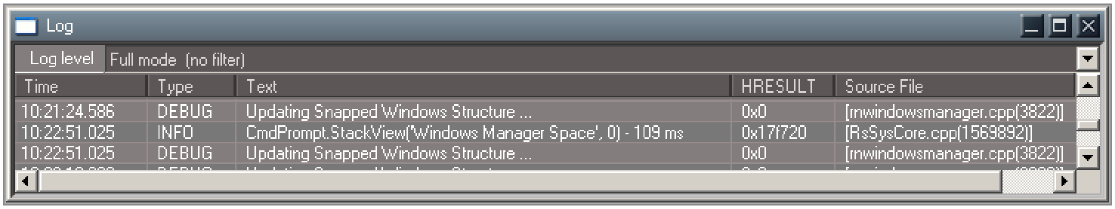
Commands

Command Name	Script Form
Widget:Move	// creating script form for 'Widget:Move '
Widget:Rotate	Widgets.Rotate("/Project/Space 3D/Cylinder",-30.889,-0.000,0.000,0.000,3.779,-3.261);
Widget:Move	Widgets.Move("/Project/Space 3D/Cylinder",0.000,0.000,-9.299);
Widget:Move	Widgets.Move("/Project/Space 3D/Cylinder",-0.000,0.000,1.097);
Widget:Move	Widgets.Move("/Project/Space 3D/Cylinder",-0.000,0.000,1.097);

Command History View

Output Console

The Output Console view shows a log of commands and results performed by trueSpace.



Log

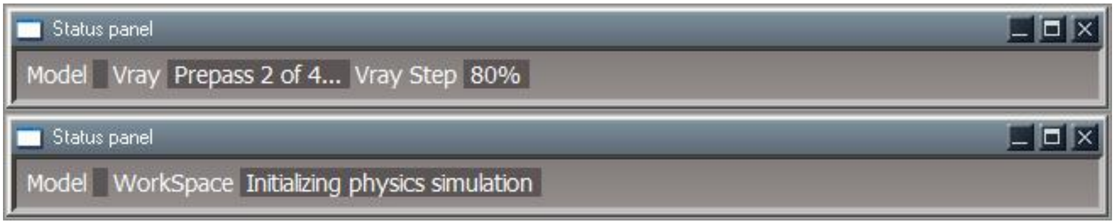
Log level: Full mode (no filter)

Time	Type	Text	HRESULT	Source File
10:21:24.586	DEBUG	Updating Snapped Windows Structure ...	0x0	[rnwindowsmanager.cpp(3822)]
10:22:51.025	INFO	CmdPrompt.StackView('Windows Manager Space', 0) - 109 ms	0x17f720	[RsSysCore.cpp(1569892)]
10:22:51.025	DEBUG	Updating Snapped Windows Structure ...	0x0	[rnwindowsmanager.cpp(3822)]

Output Console View

Status View

The Status View displays information on the status of render engine and gives feedback for other processes .



Status panel

Model Vray Prepass 2 of 4... Vray Step 80%

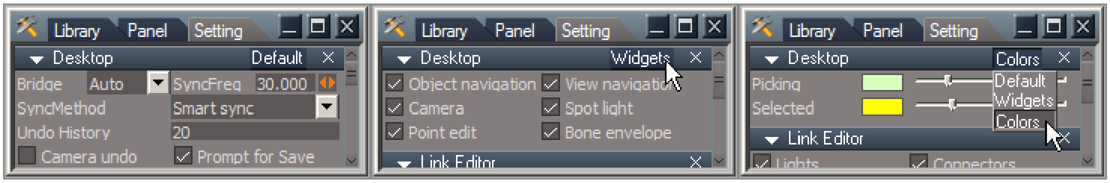
Status panel

Model WorkSpace Initializing physics simulation

Status View

Stack View

The Stack View displays Preferences, Objects and Libraries . The images below illustrates Preferences view of the Stack.



Bridge Auto
SyncFreq 30.000
SyncMethod Smart sync
Undo History 20
□ Camera undo
☑ Prompt for Save

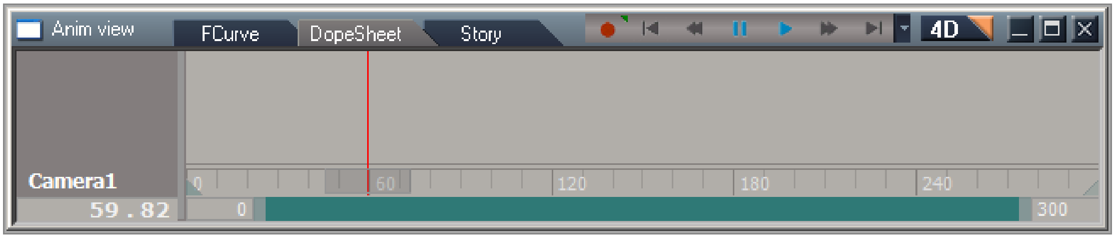
☑ Object navigation
☑ View navigation
☑ Camera
☑ Spot light
☑ Point edit
☑ Bone envelope

Picking 🟩
Selected 🟨

Stack View

Anim View

The Anim View is also referred to as 4D. It displays and allows you to edit animation keyframes .



Anim View

Package Manager

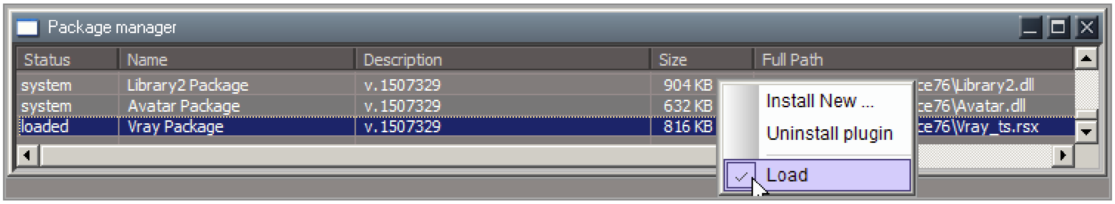
The Package manager is used to install new packages/ plugins (*.dll ,*.rsx) and also to load and unload existing ones from a r-click menu.

It displays the Status, Name , Description of version number, Size and Full Path of the extension.

To load a new extension , choose install new and browse to the package (*.dll ,*.rsx) once the package is in the List then it can be loaded and unloaded.

A package / plugin is removed by the menu Uninstall plugin.

Non valid menu choices are disabled for system packages, e.g.; System packages cannot be uninstalled.



Status	Name	Description	Size	Full Path
system	Library2 Package	v.1507329	904 KB	ce76\Library2.dll
system	Avatar Package	v.1507329	632 KB	ce76\Avatar.dll
loaded	Vray Package	v.1507329	816 KB	ce76\Vray_ts.rsx

Package Manager

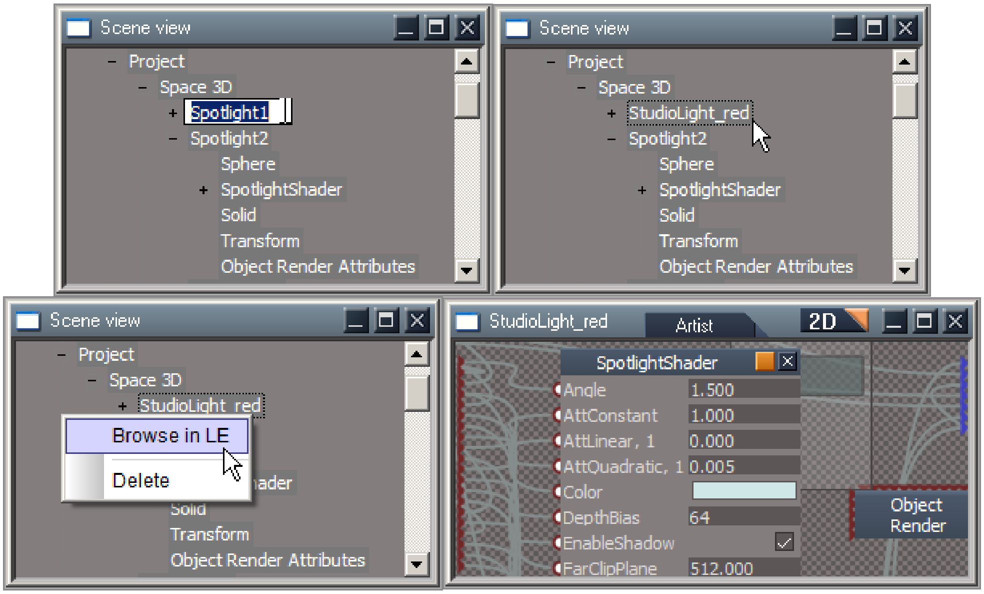
Scene View

The Scene View provides a tree based view of all the objects in the current project. Object structures can be expanded or folded by using the + and - signs to the left of the name to allow navigation into their sub-parts. The entire project graph can be browsed.

Rename: L-click select object and L-click again gives a rename option for the selected object.

Right-click menu gives an option to browse the selected object in an LE window or choose to Delete the object.

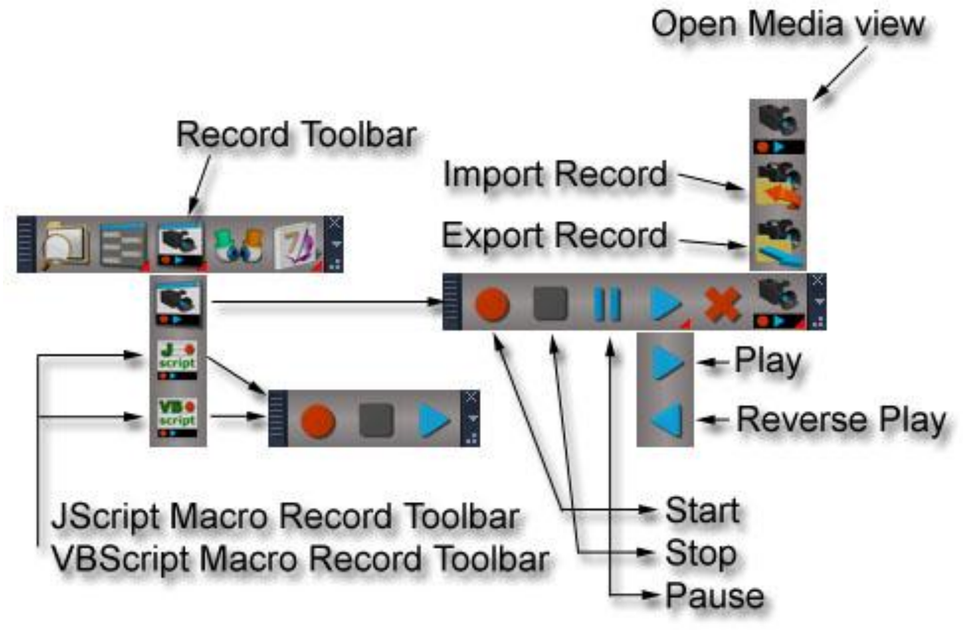
Note- For safety of the project graph integrity deleting objects that exist outside of the "Space3D" is disabled. (Space3D can be the name of your currently opened scene).



SpotlightShader	
Angle	1.500
AttConstant	1.000
AttLinear, 1	0.000
AttQuadratic, 1	0.005
Color	
DepthBias	64
EnableShadow	✓
FarClipPlane	512.000

Scene View

2.13 Macro Recording and 3D Recording



2.13.1 Macro Record

Macro record is a powerful tool that can save you time during intensive work with trueSpace, repeating similar operations for various objects, etc. It also allows you to share your results/actions with other users.

We will distinguish between 2 types of macro record: **Scene** (plain) record, and **Selection** based record.

With **Scene** recording, you can record actions like putting several objects into the scene, relocate them to form some pattern, etc. It produces a plain macro that could be played back to recreate the exact same pattern in another scene.

With **Selection** based recording, on the other hand, you can record some actions for one particular selected object, and then apply the captured macro to another object or selection of objects.

Note: There is no UI difference between recording or playback of those two types. The macro recorder automatically decides which type is used and adjusts the final script to the first or second type.

Comparison with 3D record

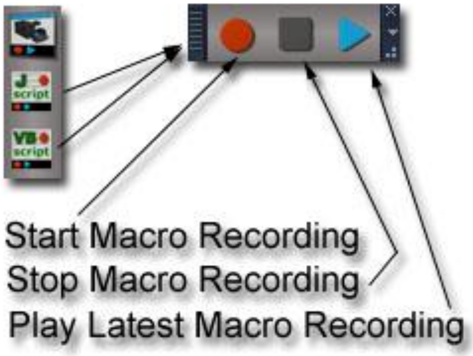
Advantages:

- Macro record produces much smaller records as it uses high-level script notations of actions.
- A macro recorded for one object can be applied to another object(s).

Disadvantages:

- Macro record cannot reproduce the flow of actions (e.g. object moving). It creates the final state with no animations.
- Macro record may fail to record or playback more complex scenarios.

Macro record toolbar



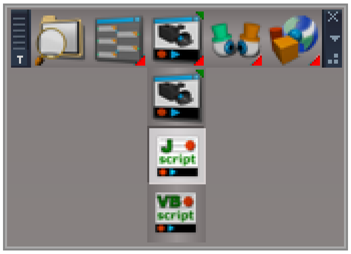
Macro Record - jScript	**Macro Record - VBScript**

The macro record toolbar can be invoked from the main toolbar, next to the Record Toolbar icon. You can choose between two languages for macros: jScript or VBScript (features are the same). The macro record toolbar contains 3 icons: **Start Macro Recording**, **Stop Macro Recording**, and

Play Latest Macro Recording. Normally you would press them in that listed order to capture and playback a macro.

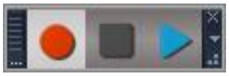
> *Note: You can playback a macro also by pressing the Start button on recorded macro object that appears in the scene after recording is over. This is the preferred way.*

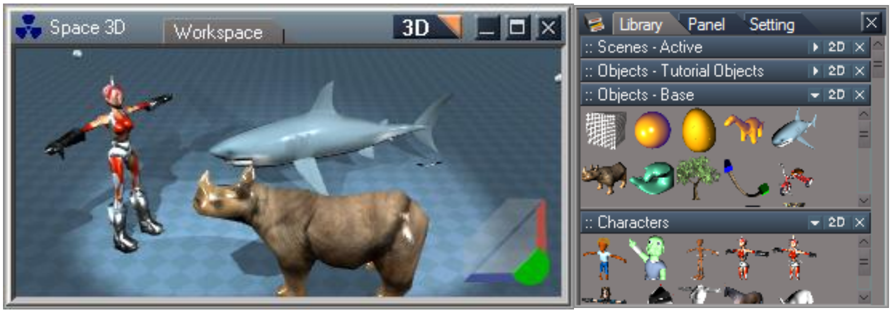
2.13.2 Macro Record Tutorial 1: Scene Macro Record

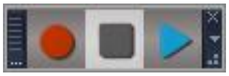


Macro Recorder

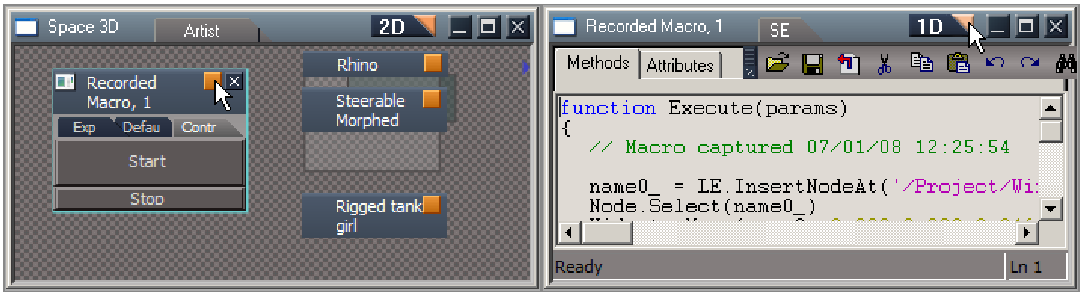
Follow these steps:

1. Start with a new scene.
2. Invoke the macro record toolbar.
3.  Press the **Start Macro Record** icon (red circle).
4. Now, add some objects to the scene.



5.  Finish recording by pressing the **Stop Macro Record** in the toolbar.

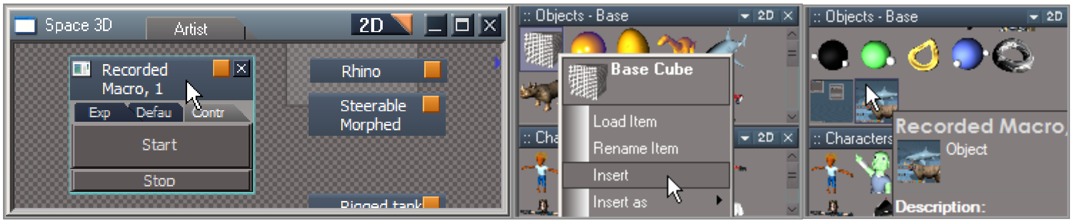
6. A “Recorded macro” object appears in the scene, containing the captured script. You can enter it and check how simple the final script looks. Exit back to the scene level using the orange triangle in the scripts title-bar



```
function Execute(params)
{
  // Macro captured 07/01/08 12:25:54

  name0_ = LE.InsertNodeAt('/Project/Wi:
  Node.Select(name0_)
```

Jscript MacroRecording

7. You can save the macro to a library and use it later to load the same objects from the libraries in another scene to automate the loading process for regular object you want to use.

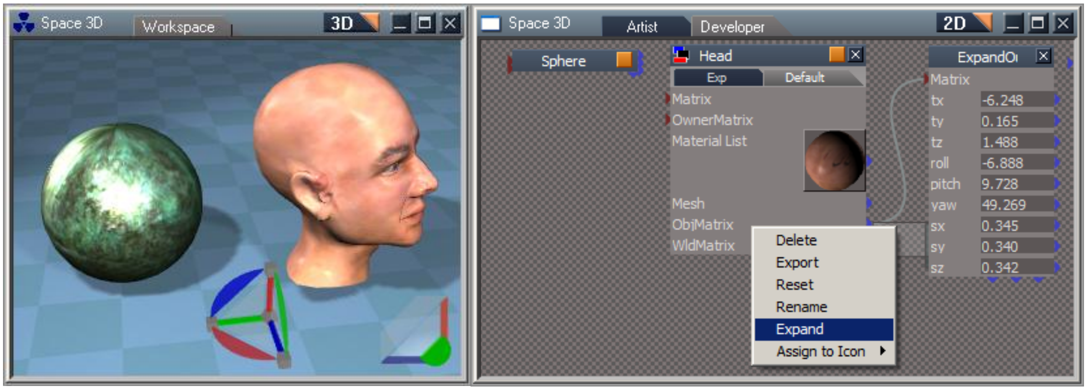


8. The macro will remain in the toolbar until trueSpace is exited.
   Clear the scene using the “new scene” icon, and press the **Playback** macro record icon The objects will appear.

2.13.3 Macro Record Tutorial 2: Selection-based Macro Record

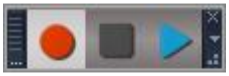
Follow these steps:

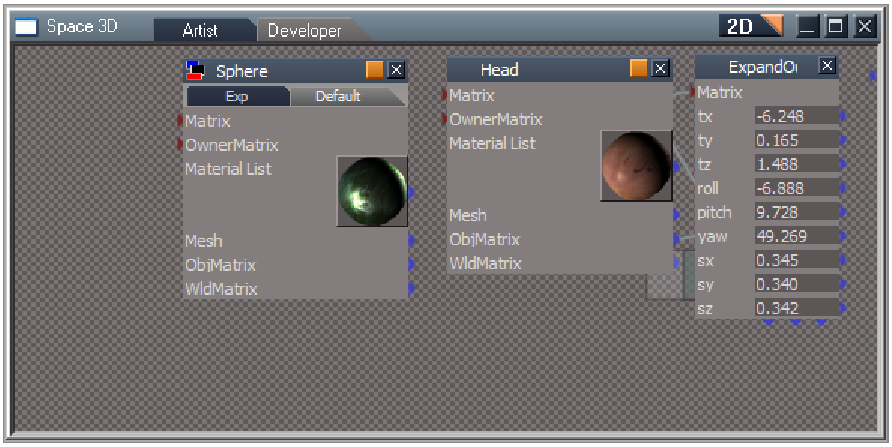
1. Start with a new scene.
2. Add the Sphere object and Head object into the scene, and expand Head’s *ObjMatrix* attribute.



tx -6.248
ty 0.165
tz 1.488
roll -6.888
pitch 9.728
yaw 49.269
sx 0.345
sy 0.340
sz 0.342

Expand Heads objMatrix

3.  Now select the Sphere object and start macro recording.

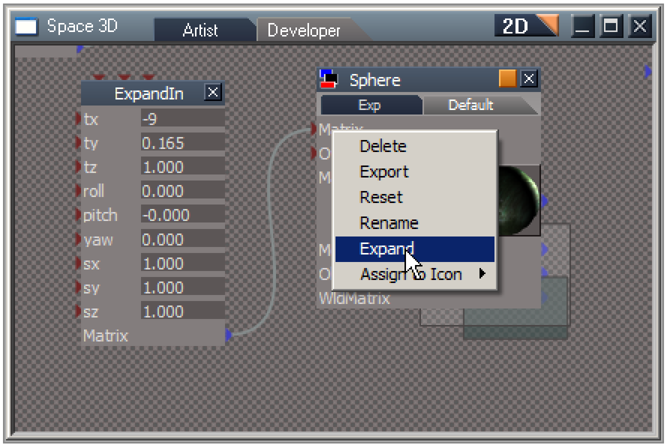


tx -6.248
ty 0.165
tz 1.488
roll -6.888
pitch 9.728
yaw 49.269
sx 0.345
sy 0.340
sz 0.342

Sphere selected

Note: Ensure that the Sphere is selected, otherwise the macro will not work properly!

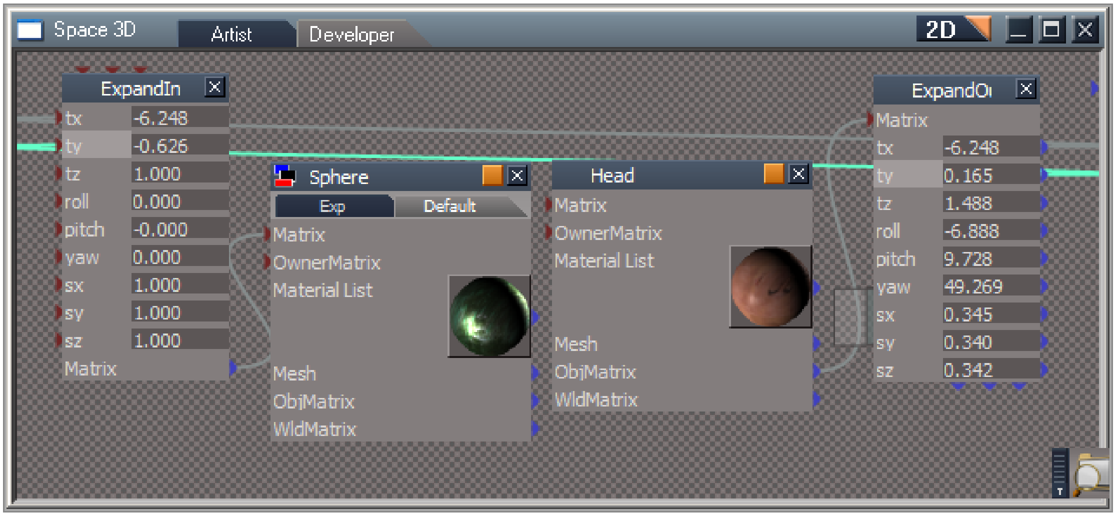
4. Expand Sphere object’s *Matrix* attribute.



ExpandIn
tx -9
ty 0.165
tz 1.000
roll 0.000
pitch -0.000
yaw 0.000
sx 1.000
sy 1.000
sz 1.000
Matrix

Expand Sphere object's* Matrix *attribute

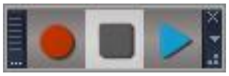
5. Switch both Expand objects to the Exp aspect, and link the *tx* and *tz* attributes from Head's expanded matrix to *tx* and *tz* in Sphere's expanded matrix. This will constrain the z and x positions between the two objects, and the Sphere will jump to its new position next to the Head.



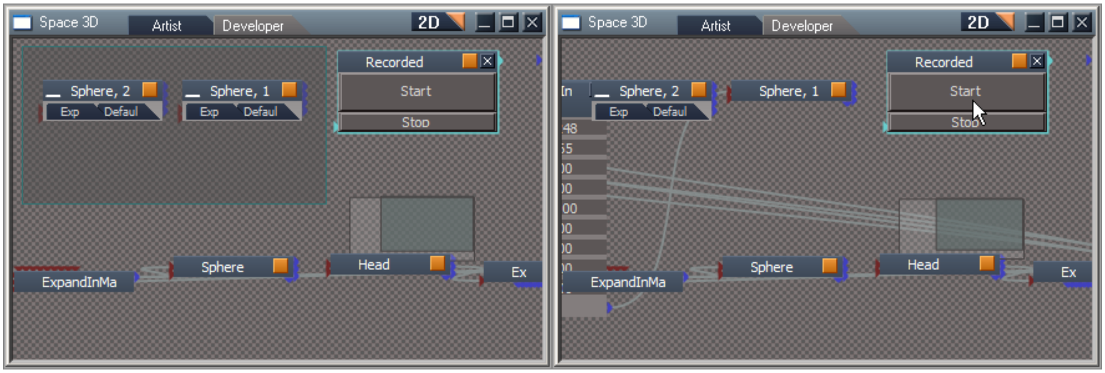
ExpandIn
tx -6.248
ty -0.626
tz 1.000
roll 0.000
pitch -0.000
yaw 0.000
sx 1.000
sy 1.000
sz 1.000
Matrix

ExpandOu
Matrix
tx -6.248
ty 0.165
tz 1.488
roll -6.888
pitch 9.728
yaw 49.269
sx 0.345
sy 0.340
sz 0.342

link the* tx *and* tz *attributes from Head's expanded matrix to* tx *and* tz *in Sphere's expanded matrix

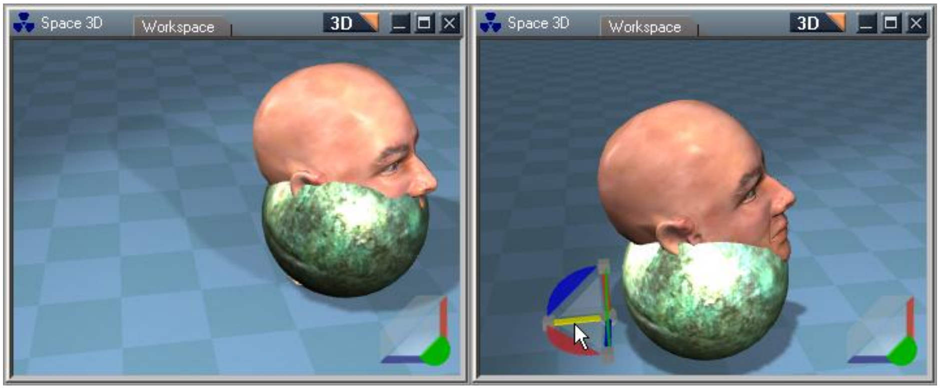
6.  Finish macro recording by pressing the **Stop Macro Record** button.

7. Now add two more spheres to the scene, select both of them in the LE, and playback the recorded macro. The recorded actions will be applied to the whole selection, so both spheres jump to their constrained positions.



add two more spheres, select both of them, playback the recorded macro

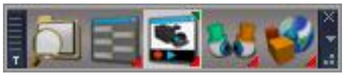
8. Play with the result. Move the head in various directions and watch the constrained spheres move.



Spheres are constrained to head

9. Add several more objects and apply the macro to them as well.

2.13.4 Record Toolbar

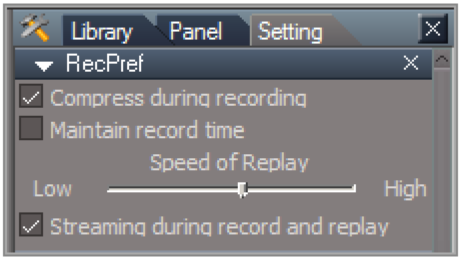


Record Toolbar

3D record is content-based recording. Unlike script recording, which processes actions, 3D record works only with the content of recorded space. This technique allows the detection of all changes caused by user action, activity, or simulation and then replays them, even in reversed order. Because of this, 3D record can be a powerful tool for recording the history of events, creating animations, presentations, training courses, and tutorials. It can also be useful as **unlimited undo** support during scene and object editing.

3D record implements full **streaming** support for recording and playing, so the size of recordings is limited only by available disk space.

Record Toolbar Preferences



☑ Compress during recording

□ Maintain record time

Speed of Replay

Low — High

☑ Streaming during record and replay

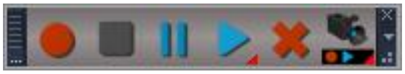
Right-click Record Toolbar to access preferences

- **Compress during recording**: Enables or disables compression during recording. It significantly reduces the size of the record file.
- **Maintain record time**: Allows applying events with constant or real recorded delays. When this option is enabled, replay will take the same amount of time as recording; each event is applied with the same delay as it was recorded. If disabled, constant 10ms delays are used for all events.
- **Speed of replay**: Controls the speed of playback. The speed value directly alters the delays between events during replay. A higher speed value results in faster replay, while a lower speed value causes slower replay. This option may also be specified when replaying records.
- **Streaming during record and replay**: When enabled, events are saved to the file during recording, reducing memory usage. Streaming also improves speed of initialization and processing. You cannot use reverse playback while streaming is enabled.

3D record toolbar

The main 3D record functionality is accessible from the 3D record toolbar.

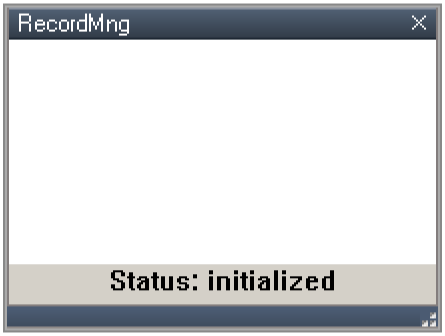
The 3D record toolbar opens when you left-click on the **Open Record Toolbar** button located in the main toolbar.



3D record toolbar

- Record: Starts recording. First, the current state of recorded space is saved, and then all subsequent changes in content, activities, or simulations are recorded until the stop button is pressed.
- Stop: Stops recording or replaying the recording. When replaying it resets the position to the beginning of the recording.
- Pause: Interrupts the replaying of the recording. Unlike the stop command, it allows you to continue replaying from the last interrupted event. The pause command has no effect when recording.
- Play: Starts replay of recorded or imported events. The speed of replaying can be controlled in record preferences panel. 3D recorder uses a separate thread for replaying events, which greatly improves user interaction with the scene during replay.
- Reverse Play: Starts reversed replay of recorded or imported events.
- Clear: Erases the current record from the record manager.
- Open media control: Opens control for attached media preview file. Record manager allows the synchronized replay of 3D record and 2D video media files. Because 3D record does not record UI changes, the media preview file provides a convenient way to include 3D recorded video fully synchronized with 3D recording. For example, you could watch a 2D Video course, stop it at any time and continue where video left off by direct interaction with

3D scene. You can find more information about creating and attaching media preview in the reference documentation.



RecordMng

Status: initialized

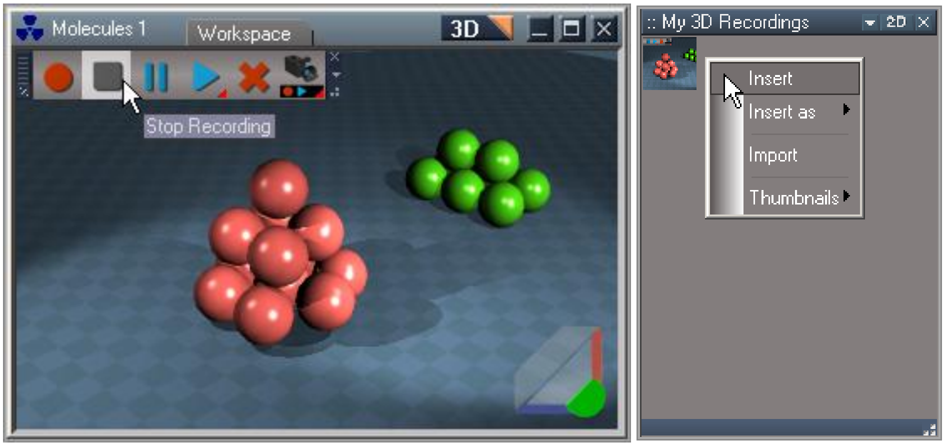
Opened Video Preview panel

- Export: Saves recorded events to the specified file. You may also use the record library for exporting and sharing recordings among users.
- Import: Loads recorded events from the specified file or library. When loaded, it replaces the current scene with the initial state of the recorded space. It also opens a control for attached media if available.

2.13.5 3D Record Tutorial 1: Recording Physical Simulation and User Interaction

Follow these steps:

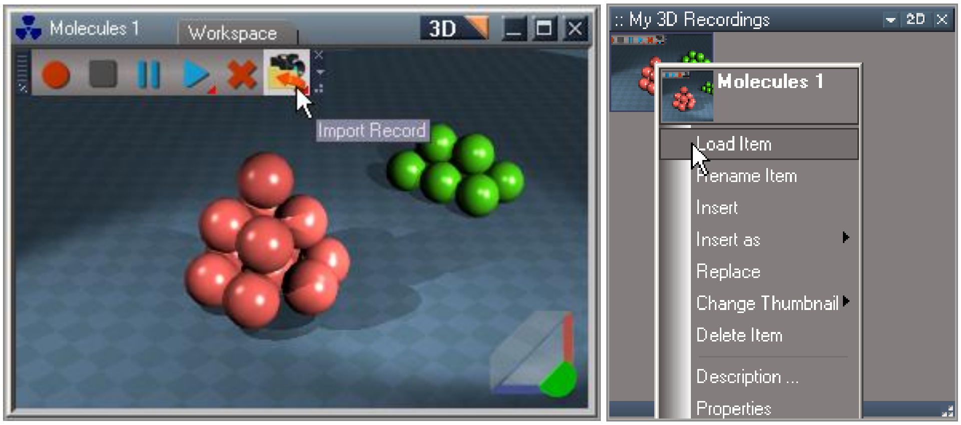
- Switch to the full layout and generate a new space.
- Select the Workspace and open the “molecules1” scene from the Base scene library.
- Open the record toolbar and start recording by pressing the record button.
- Start the simulation using the **Start/Stop** tool and then select the **PhysMove** tool.
- Use the **PhysMove** tool and interact with molecules for 30 seconds during the simulation.
- Stop recording by pressing the **Stop** button in the record toolbar.
- Stop the physical simulation.
- Store the recording in the record library, or export it directly using the record toolbar.



2.13.6 3D Record Tutorial 2: Playing a Recording

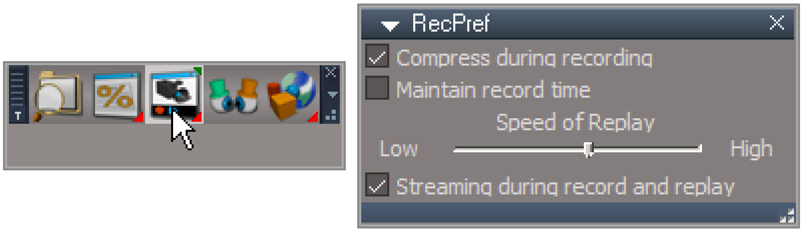
Follow the steps in the previous tutorial to create a recording of physical simulation with user interaction.

- Open a recording from the record library by single-clicking on the recording. You can also import records by using the Import button. Initial record state will display in Workspace when successfully loaded.



Load or import a record

- Right-click on the **Open Record Toolbar** button in the main toolbar to bring up record preferences panel.



RecPref

☑ Compress during recording

□ Maintain record time

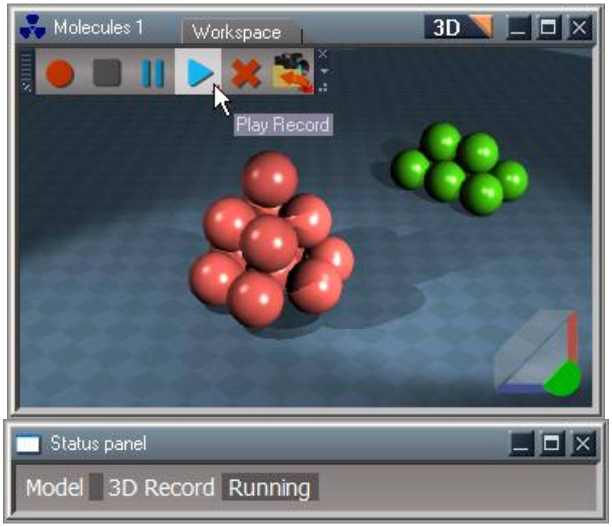
Speed of Replay

Low — High

☑ Streaming during record and replay

R-click the open record tool button to show the RecPref panel

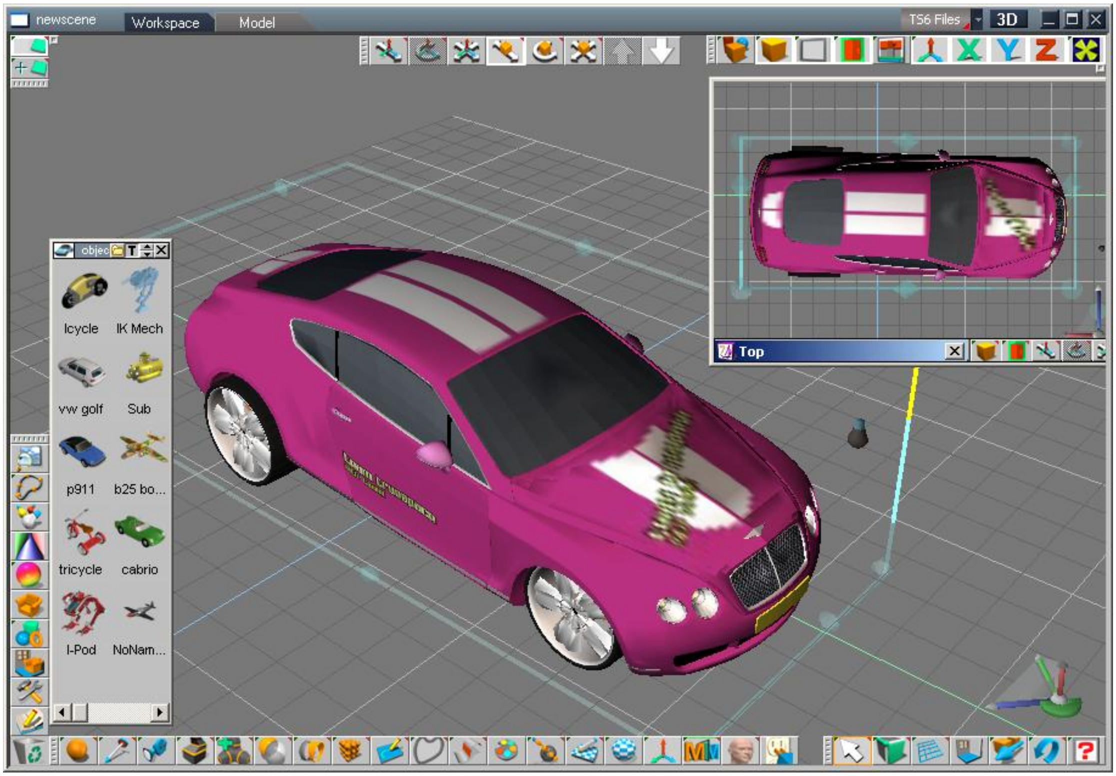
- Uncheck the *Maintain Record Time* option in the preferences panel, and press the **Play** button located in the record toolbar to start replay.
- You may control the speed of replay by using the slider in the preferences panel.
- You can also navigate in the view using the view widget during replay.
- Press the pause button to interrupt replay, and press it again to continue in replaying.
- Use the **Stop** button to finish record replay.



Model 3D Record Running

Record playing

2.14 3D View – Model



Model View

Model View of trueSpace will be familiar to users of previous generations of trueSpace. There are literally hundreds of modeling and animation tools available in the older, pre-trueSpace7 format, and we decided to keep this part in its existing format with only minor changes such as newly redesigned icons.

In some cases keeping the older modeling tools with their interfaces intact means that you will see duplicates of some control elements between the older legacy Model View, and the newer Workspace View. Over time, as we transfer all modeling tools to the Workspace View, the duplicates will be eliminated. For now, however, you can enjoy all the benefits and comfort of the existing tools, which are integrated with the new Workspace tools via a real-time bidirectional **bridge**. This bridge has many benefits. For example, you can use the existing key frame editor and watch as well as edit the animation in the new photorealistic Workspace View. We hope that it is worth putting up with the inconvenience of a few duplicate icons and toolbars.

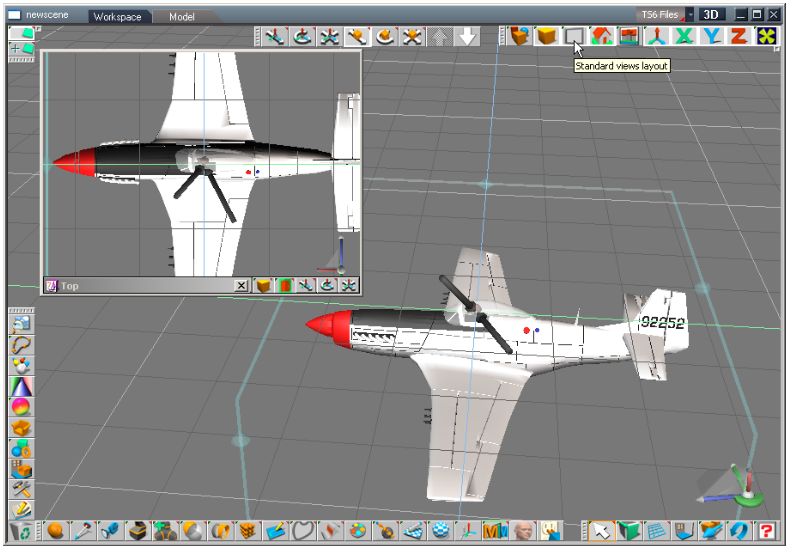
2.14.1 Views

The view above is the default view for Model. The title bar and menus are along the bottom of the screen, and strips of icons arranged in logical groups surround the main work area or workspace. The grid that crosses through the workspace runs along the X and Y axes and is used as a visual reference as you work, but it will not appear in your final rendering.

There are two view modes in trueSpace: **standard** and **4-view**.

Standard View Layout

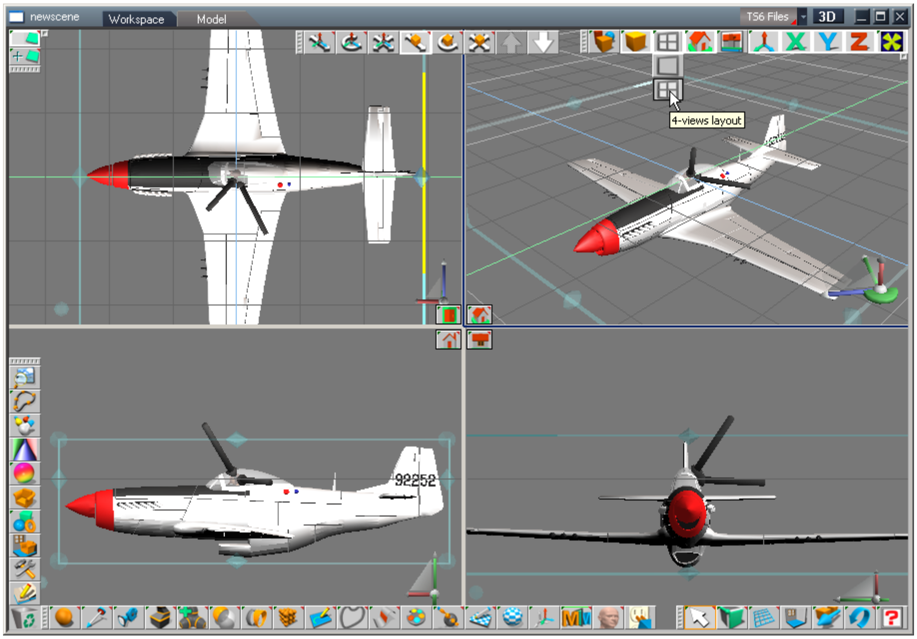
The default interface for Model View, this presents you with a perspective view, and up to three auxiliary windows of different views. Each auxiliary window can be moved and sized independently, or in the case of dual monitor support, can be moved to the second monitor's screen. To access the 4-views option, either click and drag on this icon, or just hover the mouse pointer over the icon with the button depressed, and a flyout menu will appear.



Standard View Layout

4-Views Layout

This layout splits the screen into 4 equal parts of top, front, left and perspective views. These view windows can be resized by dragging on the division line between the views, and changed using the **View Select** icon (see below). Dual monitor support has no effect in this view.



4 View Layout

Camera View

Although in Model View any object may be used as a camera, special non-rendering camera objects are also available. One or more such cameras may be placed in a scene, each one with its own settings and animations. The camera's point of view can be zoomed in our out by altering the Z-scale parameter in the **Object Properties Panel**, accessed by right-clicking the Object Tool. (See **ARTIST GUIDE APPENDIX B: PREFERENCES**, for more information about this panel.)

To place a camera object in a scene:

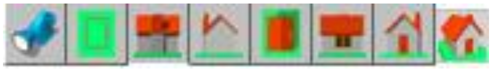
1. Activate the **Camera** (or **Panoramic Camera**) tool. The mouse cursor will change to a pointer with a small camera icon attached to it.

2. Click and hold the left mouse button to place a camera object in the scene.
3. Drag while holding down the left mouse button to move the newly placed camera object parallel to the ground (XY plane).
4. While still holding down the left mouse button, press and hold the right mouse button to move the camera vertically along the Z axis.
5. Hold down the CTRL key to rotate the camera. The camera will follow the cursor's movement.

Tip: The Camera View is most often used with Camera objects, but it is also useful for "aiming" other types of objects. For example, use the Camera View on a spotlight to pinpoint the light's center of focus, or use it on a character's eyes to see exactly where they are looking.

New View

This icon group enables you to open up to three new view windows of any view type while in Standard Views Layout: top, bottom, left, right, perspective, front, or back .

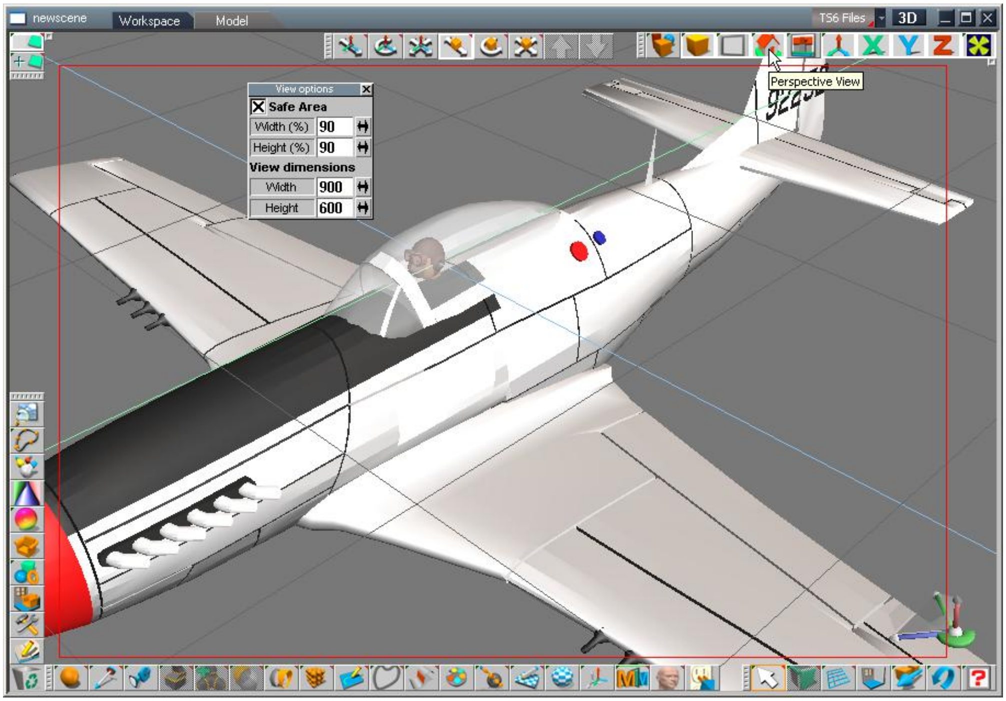


New View options

Video Safe Area

Although computers display the entire working area of the screen, video displays and TV sets are usually set up with a portion of the display hidden behind the frame, or bezel, which surrounds the screen. Therefore all video imagery which is intended for video playback with an amount of bleed area should be designed in such a way so that the important visual information is kept within the video safe area.

User-specifiable safe area lines are provided by trueSpace, showing the relevant part of the view, which is guaranteed to be visible in the video output. These lines can be set on a per-view basis.



View options

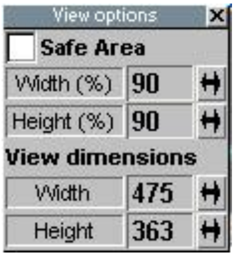
☒ Safe Area

Width (%)	90
Height (%)	90

View dimensions

Width	900
Height	600

The **View Options panel** (available by right-clicking the view type icon) contains the interface for enabling the safe area lines.



View options

□ Safe Area

Width (%)	90
Height (%)	90

View dimensions

Width	475
Height	363

- **Safe area:** Enable safe area lines in the active view.
- **Width:** The width (in percent of the complete view width) of the safe area.
- **Height:** The height (in percent of the complete view height) of the safe area.

2.14.2 Display Modes

D3D Mode

D3D is short for **Direct3D** and is the default mode for trueSpace. This allows you to see a preview of your objects with lighting and textures in real time as you make changes without having to take the time to render. Most 3D accelerator cards support Direct3D.

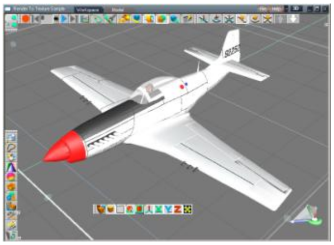
Right-clicking on the D3D icon gives you a properties menu where you can adjust the display settings. This is covered in detail in **APPENDIX B: PREFERENCES**. One feature of immediate interest is the transparency setting:

Transparency: This sets the level of transparency used in real time rendering for objects that are being edited.

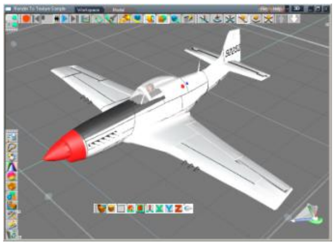
- Level 1: This setting will give you the best performance, but the lowest quality of transparency. In some cases you will not see what is behind foreground objects.
- Level 2: At this transparency level, all objects are fully transparent, but they are highly saturated with light. In this level the color blending is the same between 2 or more objects, so it becomes difficult to distinguish between overlapping objects.
- Level 3: This is the highest quality transparency, but also the slowest to render.

Wire Frame Mode

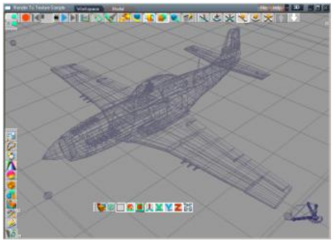
Wire frame mode renders only the vertices and meshes that make up your objects and scenes. This is the quickest interface if your graphics card cannot do D3D or OpenGL as there is much less calculating and rendering to be done, but it is also the least informative: you cannot see previews of how your textures will look or how the lighting strikes your models as you work with them.



D3D Solid Mode



OpenGL Solid Mode



Wireframe Mode

2.14.3 Draw Modes

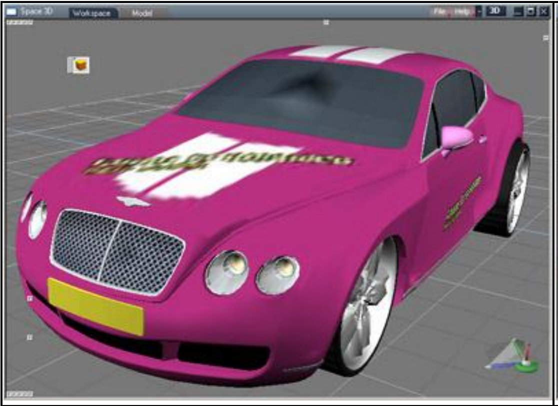
With this icon group you can set the real-time render modes.

Draw as Solid

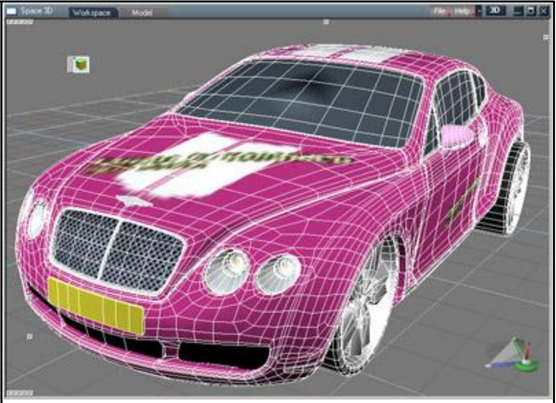
trueSpace will render all objects in your scene as completely opaque, shaded, and preview rendered.

Draw as Solid Outline

trueSpace will render the objects just as in solid mode and also outline the edges and vertices of the objects.



Solid Mode



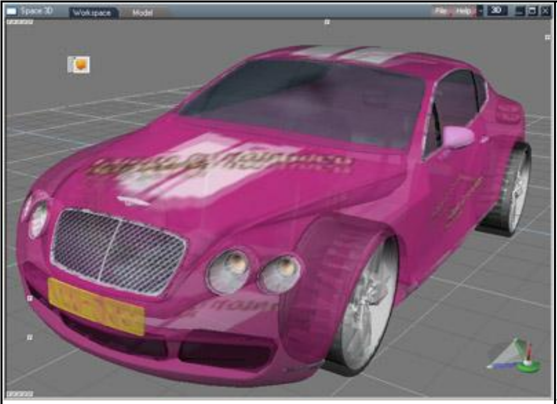
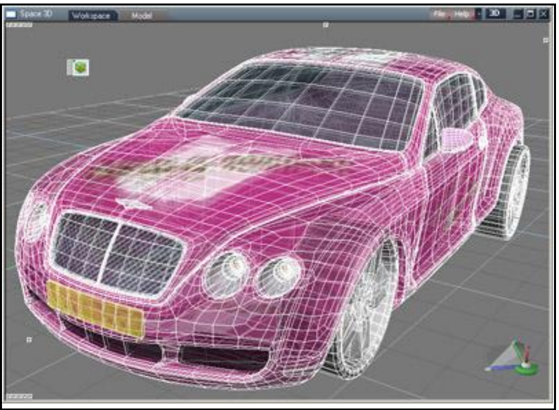
Solid Outline Mode

Draw as Transparent

This will allow you to partially see through objects.

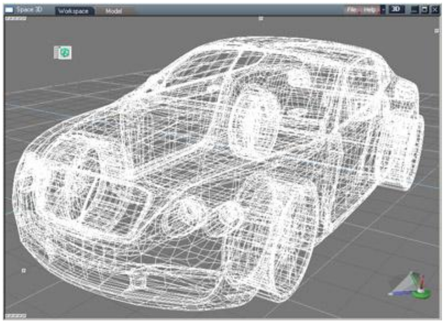
Draw Transparent Outline

This is similar to **Draw as Transparent**, with the added function of highlighting all vertices and edges.

	
Transparent	***Transparent outline***

Draw as Wireframe

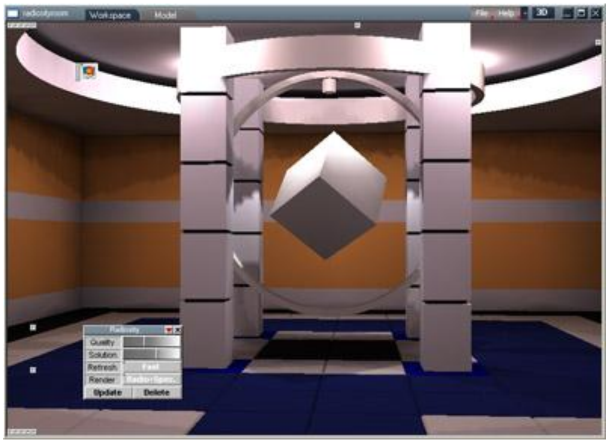
This will not give you a preview render of what your object will look like in the final render, but it will speed up navigation in the interface because there is less information on screen for the video card to update.



Wireframe

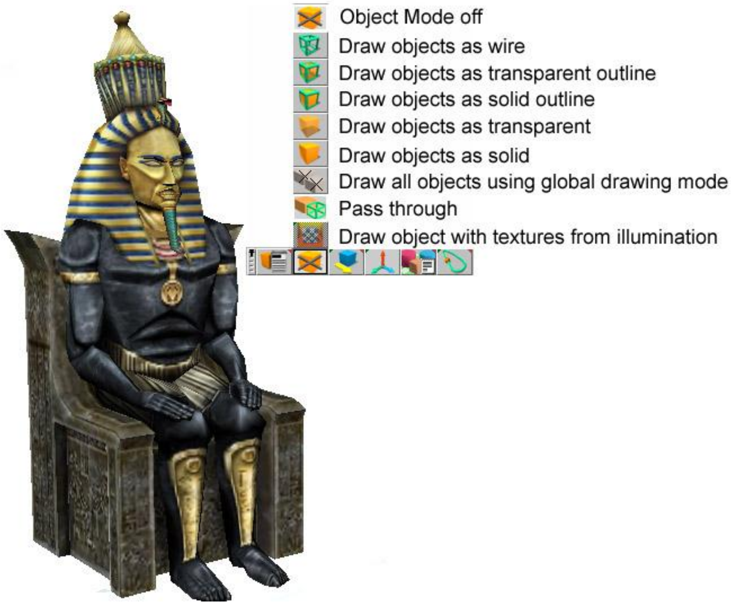
Draw as Radiosity

trueSpace will create a real time radiosity solution. You can navigate in real-time or create a non real-time photorealistic render.



Draw as Radiosity

Per-object Draw Modes


Object Mode off
Draw objects as wire
Draw objects as transparent outline
Draw objects as solid outline
Draw objects as transparent
Draw objects as solid
Draw all objects using global drawing mode
Pass through
Draw object with textures from illumination

In addition to setting the global **Draw Mode**, you can also set per-object draw modes. This allows you to do things such as manipulate a wireframe mesh using a solid mesh or texture painted on a backdrop as reference.

The per-object **Draw Modes** are accessed via the object's **Selector Context Toolbar**, described later in this section. The available per-object draw modes are **Solid**, **Solid Outline**, **Transparent**, **Transparent Outline**, **Wireframe**, and three additional modes: **Pass through**, **Textures from Illumination**, and **Object Mode Off**.

To enable or disable per-object draw modes for all objects in the scene, use the *Pass Though* toggle on the Display Options panel, accessed by right-clicking any of the global draw modes. See **APPENDIX B: PREFERENCES** for details on this setting.

Note: Holding down the ALT key while selecting a global draw mode will set all objects' draw modes to the same selection. Holding down the ALT key while selecting *Pass Through* will switch off all per-object draw modes.

Draw Object with Textures from Illumination

If textures from illumination have been generated for this object (see **ARTIST GUIDE CHAPTER 5: LIGHTING AND RENDERING**), enabling this mode will cause the current object to be drawn as solid with textures from illumination. This mode will not be available if the current object does not have textures generated from illumination.

Object Mode Off

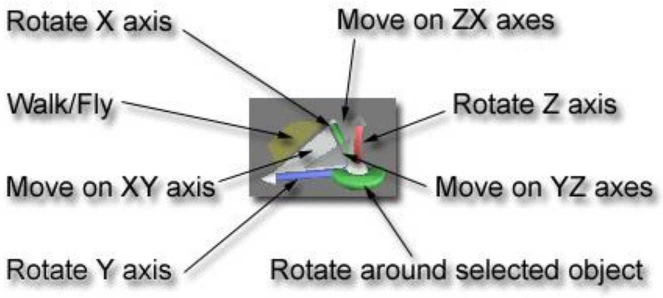
When enabled, this turns off the custom per-object drawing mode for the current object until it is either disabled, or another drawing mode is chosen for the current object.

Pass through

If Pass through is turned off, global settings are used. If Pass through is turned on, then screen settings and local flags are combined and used.

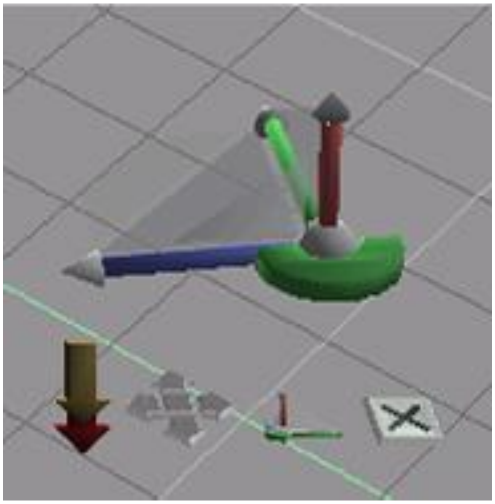
2.7.4 Navigation and Selectors

Moving around your world in trueSpace can be done through two different tools: the **View Navigation Tool**, and the **View Control**. Each of these will move the view around your scene, rotating, moving or zooming.

	
View Navigation Tools	***View Control***

The **View Control** is in every view window for quick and easy access. It enables you to move and rotate as needed, all from one location. The **View Control** consists of control points: rods that run parallel to their axes of influence for rotations, and planes that are used for moving.

Each control point also has a right drag function that performs the same action (rotate or move) but on the other axes. In addition, the green base of the control is used for rotating around the currently selected object, and the gray sphere at the center has two functions: left-drag to move the view control, and right-click to open up the options panel for the view control.



The control options, left to right as pictured:

Walk / Fly: Toggle control between **Walk** and **Fly** modes of the navigation tab on the **View Control**. Left-click and drag or hover over the downward arrow to get the flyout list. There are two modes of moving in trueSpace: **Walk** and **Fly**. The difference is subtle and all in the mouse buttons, but it can be significant. In **Walk** mode, your view moves when you move your mouse. Hold your left button down and moving your mouse left or right will move your view in that direction. Hold down the right mouse button and moving your mouse forward or back will move your view up or down. The view only moves as far as you move your mouse. In **Fly** mode there is no need to keep moving the mouse; the view will move as long as you still hold the button down. For instance, in fly mode if you click the left button and move the mouse to the left slightly, your view will continue to move as long as you keep the mouse button depressed.

Size control: Drag on this option to resize your **View Control**.

Delete View Control: Clicking this will cause the View Control to disappear. If you wish to bring the View Control back, you can use the **Global Panel** or the Preferences panel. In the Preferences panel, simply toggle the "3D controls" checkbox. The **Global Panel** is a 3D interface that you can use to adjust many of the settings in trueSpace. To bring back that View Control:

1. Open the **Global Panel .**
2. Select Menu Settings.
3. Select "Enable View Controls creating" to turn it off.
4. Select "Enable View Controls creating" to turn it back on.
5. Close the global panel by clicking the "x" on its interface.

The View Control will be automatically recreated when the creating option is enabled. The Global Panel is covered extensively in **APPENDIX B: PREFERENCES**.

Close properties: Closes this list of control properties.

Navigation Icons

In addition to maneuvering around your environment with the **View Control**, you can use the **Eye Tools**.

Eye Move

After selecting this tool, a left drag in the workspace will move your view on the XY axes, and right drag will move you on the Z axis. In the **World** system, clicking the left mouse button and dragging causes the eye to move parallel to the XY reference grid. Clicking the right mouse button and dragging causes the eye to change altitude over the XY reference grid.

Eye Rotate

Although the coordinate system controls for eye rotation appear the same as for other navigation tools, they have different meanings. Regardless of which coordinate system is chosen, eye rotation (turning about its own axis) and revolution (orbiting) are based on the **Screen** coordinate system.

By selecting the **World** coordinate system, the eye will orbit the world center, even if the eye does not directly point at it.

By selecting the **Object** coordinate system, the eye will orbit the active object’s center (see below) even if the eye does not directly point at it. If no object is active, the eye will orbit the world center.

Eye Zoom

Zooming in a perspective or camera view is analogous to changing the focal length of a lens. You can zoom from **fish-eye** to **super-telephoto**. The photographic effects of the eye’s real-world counterparts are apparent in trueSpace. For example, when the eye is zoomed to telephoto ranges, objects will appear to be crowded together. Zooming in an orthographic view moves the eye in and out without visual distortion.

Eye Zoom is independent of any coordinate system; it moves along an axis perpendicular to the screen. Within a perspective view, the amount of zoom on the eye is limited. To “pull out” of a scene or to get close to very small objects, use the **Eye Move** tool in the **Screen** coordinate system.

Mouse Wheel Support

The mouse wheel allows easy eye manipulation without the need to switch tools:

- Turning the mouse wheel forward moves the eye/camera toward the center of the coordinate system.
- Turning the mouse wheel backward moves the eye/camera away from the center of the coordinate system.
- Dragging the mouse while the mouse wheel is pressed-in rotates the eye around the selected object. (Note that this feature does not work in the orthographic views.)

Look at Current Object

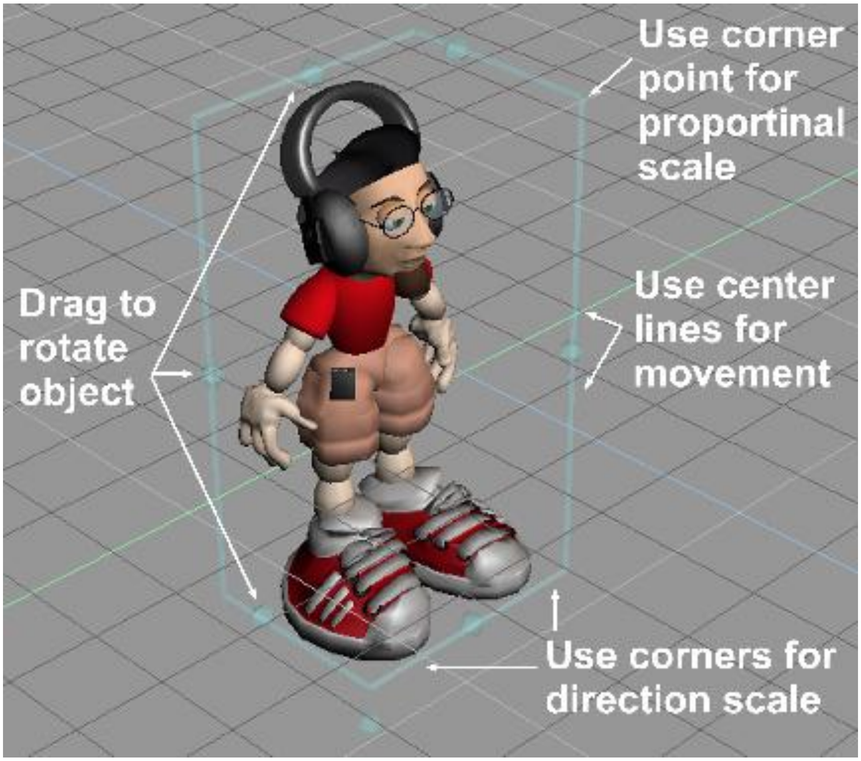
Use this tool to center the current view on the currently selected object. This moves your point of view, not the object.

Reset View

This tool will return the current view back to the default angle and location. This can be very useful for normalizing the view of your scene after you have changed the view often to focus on specific objects.

2.14.5 Object Selector Box

The **Object Selector Box** is a **bounding box** that surrounds the currently selected object. It will appear as a light blue box with conical rotation handles at the center of each side. Using this bounding box you can move, rotate or scale the object. The conical handles at the center are for rotating, the center portion of the edges is used for moving the object, and the outer portions of the edges are used for scaling the object. In addition, the joint between two edges is a control point for proportional (3D) scaling.


Use corner point for proportinal scale
Drag to rotate object
Use center lines for movement
Use corners for direction scale

The axis of movement, rotation or scaling depends on the edge that is grabbed as well as the initial direction of dragging. For instance, if you drag on the center edge at the bottom right of an object and are dragging in a rightwards motion, your object will move along the ground on the Y axis. However, if you grab the same edge and drag your mouse toward you, the object will move downward along the Z axis. Once you are moving in one direction, you can change the direction by depressing the right mouse button while still holding the left button. Similarly, to go back to manipulating on the initial axis, release the right mouse button. This method allows you to easily move along separate axes without needing to reorient your selection.

To temporarily hide the **Object Selector**, hold down the CTRL key. This is very useful when trying to select objects or portions of objects that are small enough to be obstructed by the selector, such as **point edit mode**.

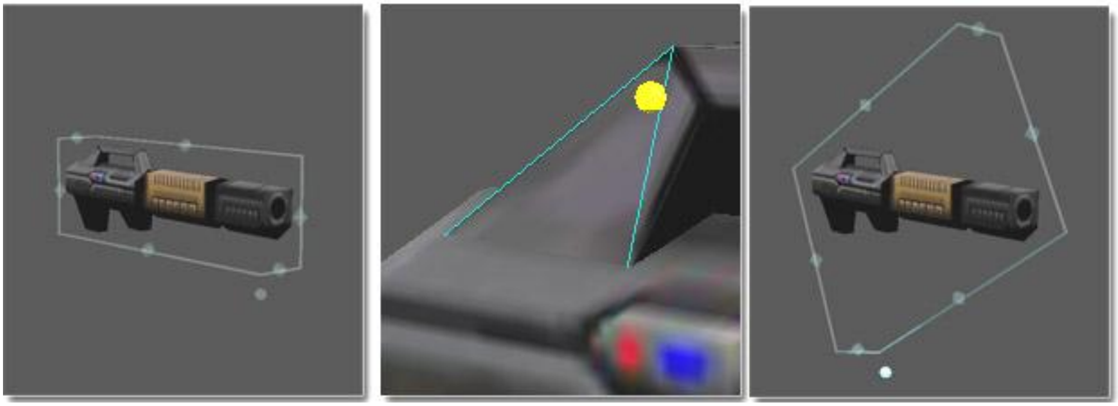
The **rotary diamonds** in the center of the box edges are used for rotating the object. (If they are not visible, you may need to toggle them on by clicking the selector orientation sphere.) Simply drag on one of the diamonds to rotate the selected object around the corresponding axis. You can also move the object's center by right dragging on the diamonds. To see this in effect, display the axes for your object, then drag on one of the rotary diamonds. You will see the axes move, and when you release the mouse button the diamond will stay parallel to where you moved the axes to give you visual cues as to where the object's center lies.

2.14.6 Object Selector Sphere

The sphere selector serves a number of different functions, including

- Re-orienting the object selector box.
- Selecting sub-objects within a hierarchy.
- Toggling display of the **Rotation Control**.
- Accessing a context toolbar containing common tools.

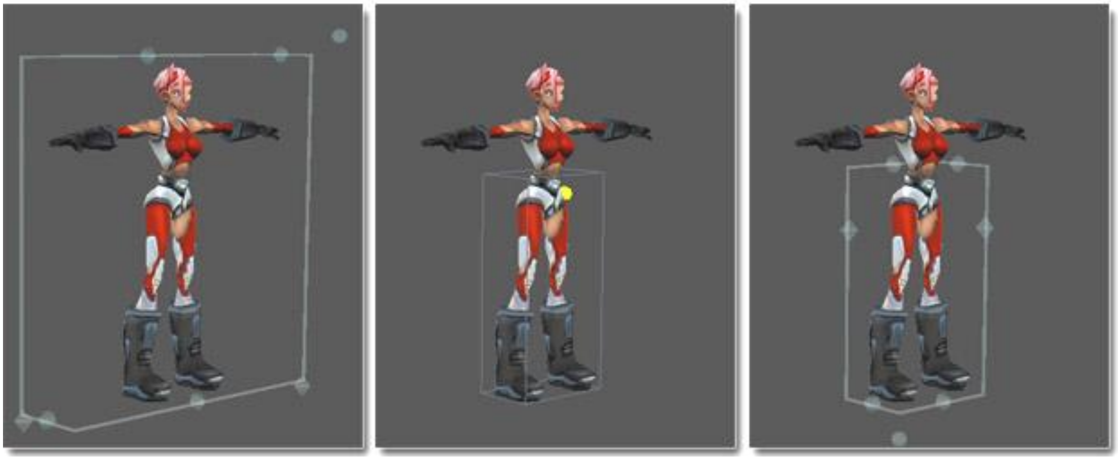
Using the selector sphere to re-orient the object selector box



Original bounding box orientation on left, dragging the selector sphere onto a face in middle and after releasing the sphere, the selector box is reoriented to the face's coordinate system at right

The sphere selector control can be used to define the selection or orientation further. For example, with a single object, you can drag the sphere and drop it on the face of an object to reorient the selector box to the face's coordinate system.

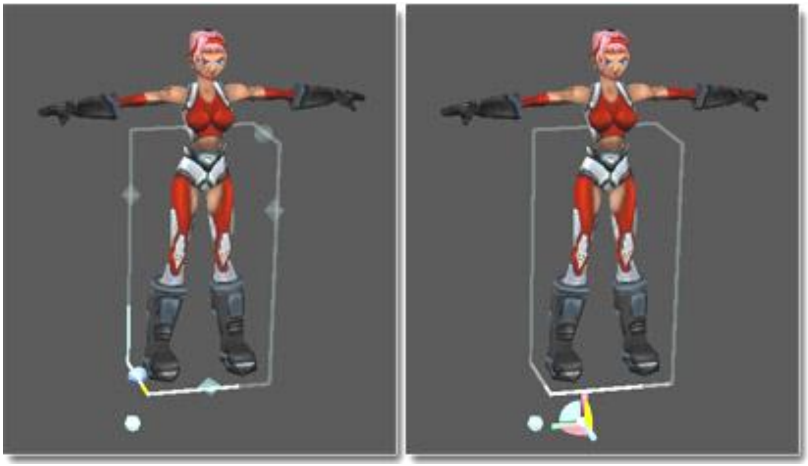
Using the selector sphere to change the current selection in a hierarchy



Original bounding box on left, move sphere around sub-objects in middle, while the right side image illustrates when you let go of the sphere and sub-obj shows bounding box

The sphere can also allow you to edit parts of an object within a hierarchy. For example, if two objects are glued together you can drop the sphere on one of the objects to select just that one. The glue relationship and the hierarchy of the objects is preserved as you edit the sub object:

Using the selector sphere to toggle display of the Rotation Control



Rotation diamond located on bounding box at left, while Rotation Control appears at right with diamonds now missing from bounding box

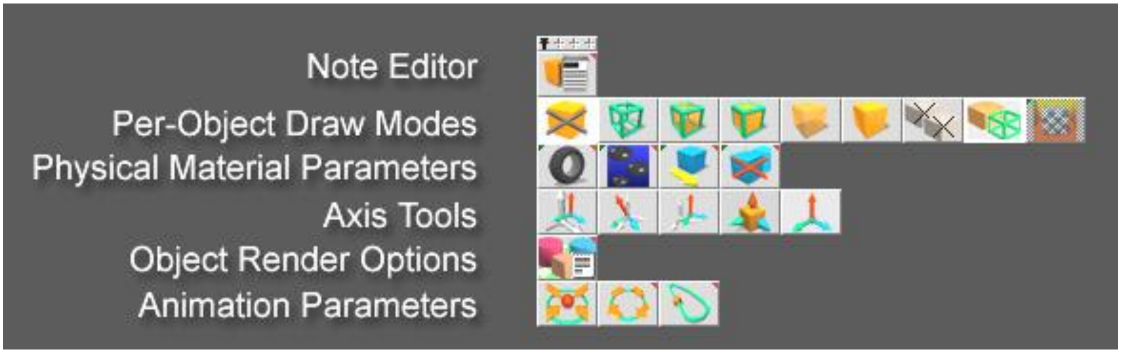
You can also use the sphere to toggle between displaying the rotation diamonds on the bounding box or the **Rotational Control**. Click once on the sphere control to toggle between these two modes.

The **Rotation Control** appears while the object is in an edit mode, such as point edit. The control can be used to rotate the selected object. This control looks similar to the **View Control**, and it functions much the same as well. However, the **Rotation Control** is used for manipulating an object, while the **View Control** manipulates a viewpoint. In addition, the right mouse button is used for rotating the object in fixed 90° increments.

The control temporarily disappears if mouse drag is detected to prevent the control from obscuring the view of the object itself, but it will reappear when hovered over. As with the **Object Selector Box**, the **Object Selector Sphere** is oriented by the selected coordinate system. The screen coordinates are not supported and if they are selected, the object's axes are used for orienting the control. Drag on the center of the rotation control to move it.

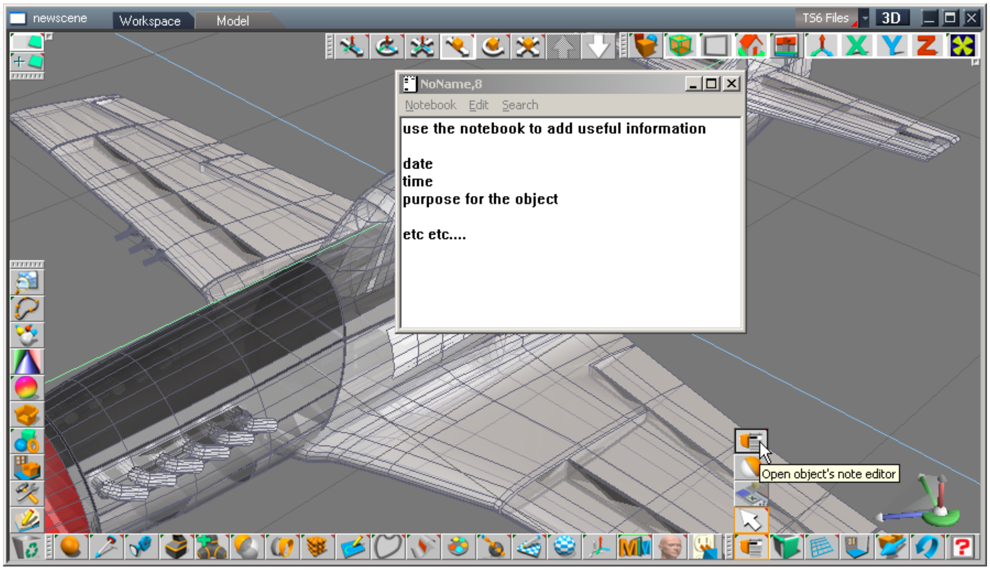
Selector Context Toolbar

Right-clicking the selector sphere will call up a context toolbar that allows you to create object notes and set per-object properties.



Note Editor

The notebook is a handy little tool that can be used to jot down object information, texturing ideas, shader settings, or anything you may wish to recall at a later time. Each object has only one notebook associated with it, and only one may be open at any time. While open, the notebook will always reflect the notes associated with the currently selected object.



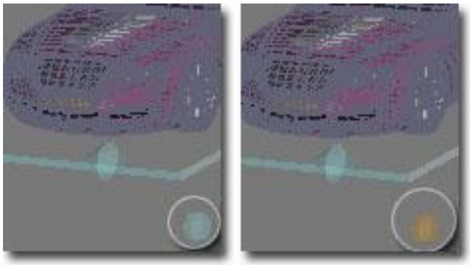
use the notebook to add useful information

date
time
purpose for the object

etc etc....

The **Notebook** icon is housed both on the selector context toolbar (accessed by right-clicking the object’s selector orientation sphere), and on the same toolbar as the Object tool. This Notebook is a toggle that applies to the currently selected object. When enabled, the notebook for the active object is displayed anytime this object is selected. Selecting a different object will cause the notebook to close (if the object’s Notebook icon is not enabled). The state of the Notebook toggle is saved in the .SCN or .COB files.

Changes to the notebook are saved automatically, and closing the notebook hides it from view. The notebook is not copied when the object is duplicated.



Blue indicates empty notebook. Orange indicates not empty

The selector orientation sphere will indicate whether or not the object’s notebook contains any information: blue for empty, and orange when a note is present.

Per-object Draw Modes

This toolbar allows you to set the draw mode (discussed previously in section **2.7.3**) on a per-object basis.

Physical Material Parameters

These tools are described in detail in **CHAPTER 9: ANIMATION**.

Axes Tools

This toolbar houses the axes-manipulation tools: normalize scale/rotation/location, move axes to center of object, show/hide axes. These tools are described in detail in section 2 of **CHAPTER 4: MODELING**.

Object Render and Radiosity Options

This button opens the object rendering options panel, described in detail in **APPENDIX B: PREFERENCES**.

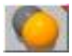
Animation Parameters

These tools are described in detail in **CHAPTER 9: ANIMATION**.

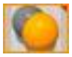
2.14.7 Toolbars

Icons

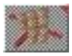
All Model icons are accessed primarily through left and right mouse clicks. Left-click activates the icon, and right-click will bring up an options or settings panel if the tool has one. There are three states that an icon may have: Active, Inactive, and Unavailable.



An **available** icon will look like a button with a gray background

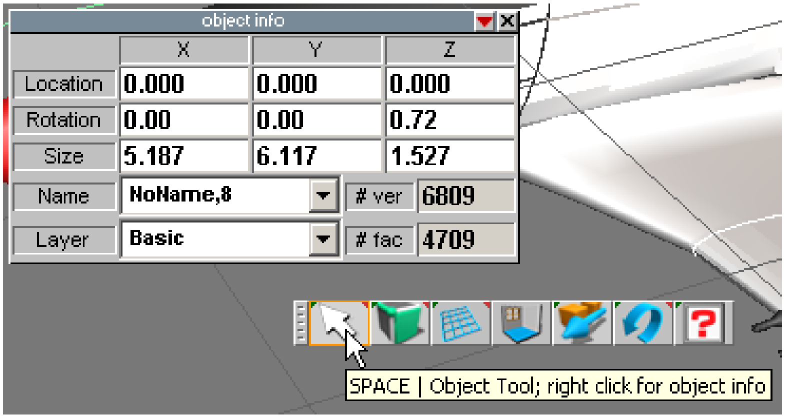


An **active** button will have a white background.



An **unavailable** button will be grayed out or ghosted. Most often a tool will only be unavailable if it does not make sense in the current context. For instance, if there are no objects in your scene yet, the **Object Union** button has nothing to work with so it is shown as unavailable. As soon as an object is created, this button becomes available.

The availability of a settings panel is indicated by tiny red triangle in the upper-right corner of the icon. To access the settings panel for a tool, right-click on its icon. (Some tools will not have a panel, and in this case right-clicking will do nothing.)



	X	Y	Z
Location	0.000	0.000	0.000
Rotation	0.00	0.00	0.72
Size	5.187	6.117	1.527

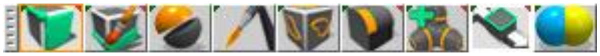
Name	NoName,8	# ver	6809
Layer	Basic	# fac	4709

SPACE | Object Tool; right click for object info

Settings for Object tool can be opened by right-clicking on Object Icon

In addition to the icons that are arranged in groups and are immediately visible, many icons are actually part of a family of tools that can be accessed through a pop-up list. The availability of a pop-up list is indicated by a tiny green triangle in the upper left corner of the icon. To activate the pop-up, left-click on the icon and drag or hover for a second.

Toolbars



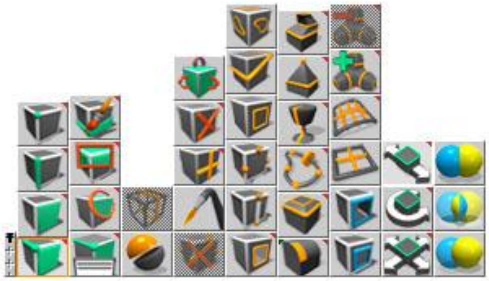
Model Point Edit Toolbar

While most Model toolbars were converted to the new format, some were kept in the older format which is slightly different. If you wish to re-arrange the tools on a toolbar or create new toolbars, you have the power to do so. There are two types of manipulation available: toolbar **sizing**, and **customization**. Toolbar sizing allows you to collapse a toolbar into a small square (effectively clearing as much work area as possible), to expand and show all icons on the toolbar strip, or to maximize and show all icons (including flyouts) on a tool bar.

A collapsed toolbar will appear as a small square. Hovering over the square will display the full toolbar, but it will collapse again as soon as the mouse moves off. Clicking on the handle of the toolbar once will expand it to show all icons on the toolbar strip, and it will not collapse when the mouse moves off.

Clicking on an expanded toolbar will maximize it. A maximized toolbar will show all icons in all groups on that toolbar:

Minimized Toolbar

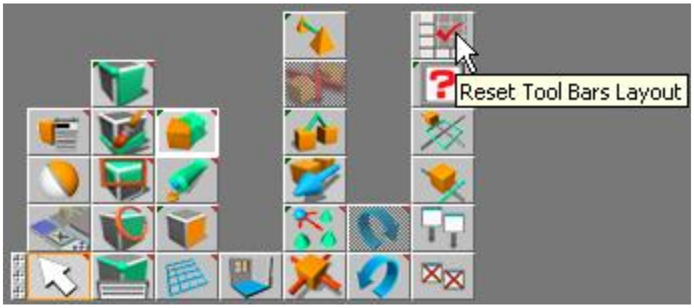


Maximized Toolbar

Clicking on the toolbar handle again will minimize it again, although the toolbar will still display until you move your mouse off of the handle.

Model toolbars can have several different location states: **docked**, **floating**, and **context**. A **floating** toolbar is one that has been dragged to an arbitrary place in your workspace. To **dock** a toolbar, simply drag it to an edge of the screen, and it will adhere to the side. A **context** toolbar is one that appears when a certain mode has been entered. For instance, when you right-click on an object, you enter edit mode, and you get a context sensitive toolbar that will sit below your object and will move with it. If you move the toolbar away from its parent object, it will be frozen at its new location. You can tell the toolbar is frozen by the small black nail in its handle bar (as shown in the image above). Moving it back over the parent object re-attaches it to the object.

Reset Toolbars Layout

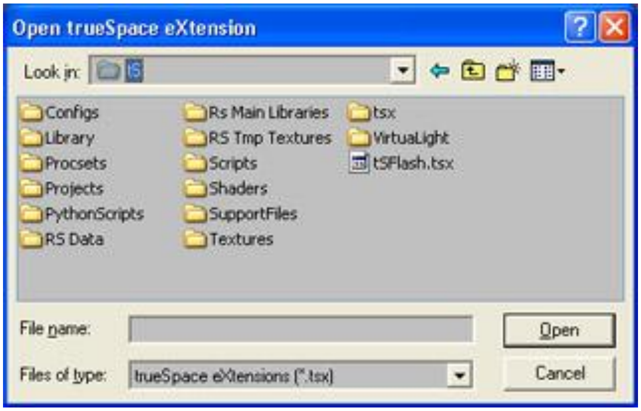


Reset Toolbar Layout location in trueSpace

After working with trueSpace for a period of time, you may find that you have changed your toolbars and icons around. If you require all toolbars to be at a default state, the Reset Toolbars Layout tool will do just that. It restores your toolbars for Model view to a default state.

3rd Party eXtensions

3rd party trueSpace extensions can be seamlessly integrated into the standard trueSpace toolbars. To load a 3rd party trueSpace eXtension, click the 3D plugin icon. You will be presented with the open extension dialog:



Files of type: trueSpace eXtensions (*.tsx)

Open trueSpace Extension dialog

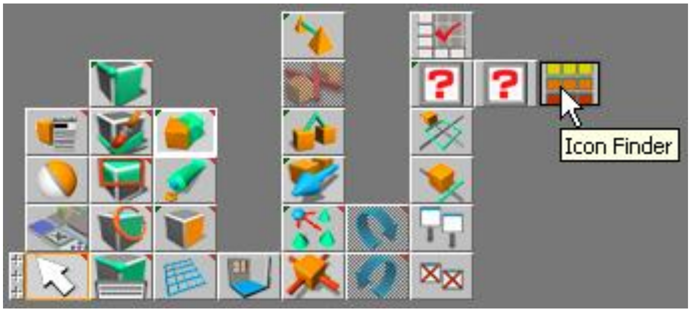
To load a particular plugin, just select the .tsx file in the dialog and click **Open**. The icon for the plugin will be added to the modeling toolbar to the left of the 3D plugin icon. You can use the trueSpace toolbar customization options described earlier to place the icon into any toolbar in trueSpace interface, or even to create a new toolbar for the plugin. To unload the plugin, just drag the plugin's icon to the **Recycle Bin**. Note that the information about plugins loaded into the trueSpace workspace as well as their position in toolbars is stored in the project/configuration library.

Delete Icons

In addition to building toolbars, you can also destroy them; shift-dragging a toolbar or icon to the trashcan on the layer toolbar will delete it.

Icon Finder

Clicking on the **Icon Finder** tool will bring up a repository of all tools available in Model. (**Note:** Use of the **Icon Finder** and **Icon Helper** is currently limited to the legacy Model toolbars, such as the object context toolbars.) This repository is a toolbar unto itself and thus will react the same as any other toolbar. Clicking on the tool again will hide the icon repository.

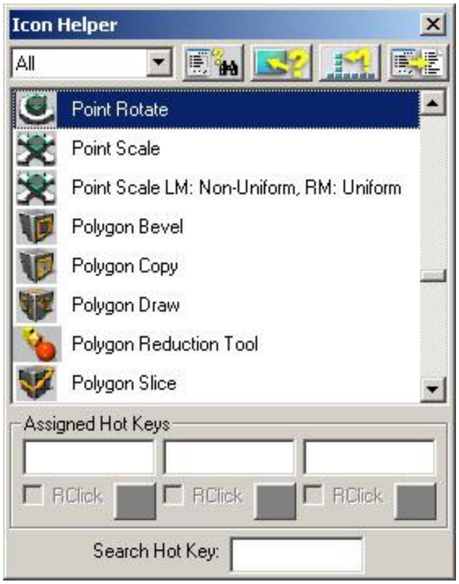


Icon Finder location in trueSpace

Icon Helper

The Icon Helper window allows you to easily find any tool or option within the trueSpace interface. (**Note:** Use of the **Icon Finder** and **Icon Helper** is currently limited to the legacy Model toolbars, such as the object context toolbars.) To open the Icon Helper, either click its icon or press CTRL+F1 while hovering the mouse cursor over a tool within the workspace. (**Note:** Using CTRL+F1 to call up the Icon Helper automatically opens the window with the selected tool highlighted.) You may scroll through the list of all usable tools (and their descriptions), or narrow the selection by using the Category drop-down box.

With it you can also find out to which tool a hot key is assigned by clicking within the “Search Hot Key” field and pressing a key. If the hot key is already used, the tool will become selected. When a tool is selected, the assigned hot key will be displayed in the “Assigned Hot Keys” boxes. These boxes may also be used to assign, change, or remove the selected tool’s hot keys by clicking inside one of the boxes and making the desired changes.



The Icon Helper window contains the following controls:

- **Category:** This drop-down box allows you to narrow the list to a category of tools. There are two special categories: Options and Hot Keys. The Options category includes all check boxes and radio buttons, including those found in options panels for tools. The Hot Keys category displays all tools and options with assigned Hot Keys.
- **Find Text :** Click to open a dialog box that allows you to search for a tool by specifying text.
- **Inspect Tool :** Click to start inspect mode. In inspect mode, the cursor changes, and you may click on any button in the interface. The selected button will be highlighted in the list box, allowing you to do things such as assign or change its hot keys. Right-click to stop inspect mode.
- **Find Tool :** This button allows you to locate a tool within interface. After clicking this

button, a short animation shows you where the highlighted tool or option button is housed. (**Note:** Do not move the cursor during the animation because the cursor's position is controlled by the program.)

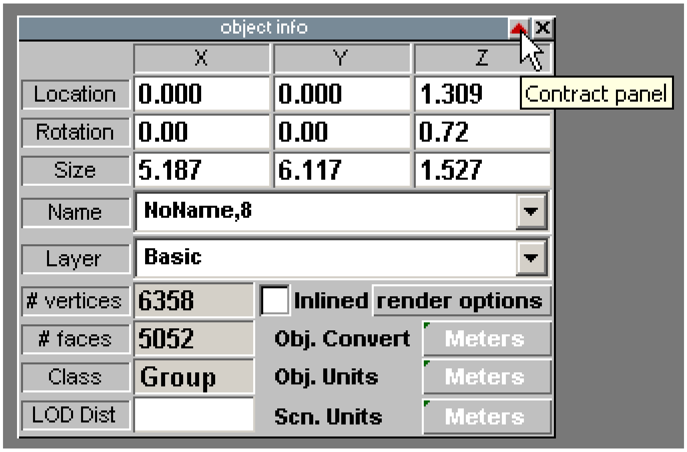
- **Save Hot Keys :** This button allows you to save a list of hot keys in text or HTML formats for reference. **Note:** When saving to HTML, the icons are saved into a subfolder of the chosen destination. The subfolder name is created from the HTML file name with suffix "images." For example, if you save the hot key list as *c:\test\keylist.htm*, the images would be stored in the *c:\test\keylist_images* folder.
- **Tool list:** This list box displays all the available tools and options.
- **Assigned Hot Keys:** Here you may assign, change, or remove a hot key. To add or change a hot key setting, click an edit field and press a key. (**Note:** You may use any edit field, but the key will be displayed in the first empty field.) To remove an existing hot key, click on the "Remove Hot Key" button. To make the hot key perform the "Right-click" functionality of the highlighted tool, check the "RClick" box. (For example, if you set CTRL+S as a hot key for the Scene Editor, and enable "RClick" for this hot key, then pressing CTRL+S would open the preferences panel for the Scene Editor as if you had right-clicked the SE icon.)
- **Search Hot Key:** You can find out to which tool a hot key is assigned by clicking within the "Search Hot Key" field and pressing a key. If the hot key is already used, the tool will become selected.

Note: Double clicking an entry runs the tool or toggles the option, as appropriate.

2.14.8 Object Tool and Panels

Object Tool

The object tool is used for selecting objects and manipulating them. When you select the object tool you will exit whatever other mode you may have been in, such as painting or point editing. A right-click on the object tool will bring up the object properties panel with the properties of the current object displayed.



object info

Contract panel

	X	Y	Z
Location	0.000	0.000	1.309
Rotation	0.00	0.00	0.72
Size	5.187	6.117	1.527

Name: NoName,8

Layer: Basic

# vertices	6358	□ Inlined	render options
# faces	5052	Obj. Convert	Meters
Class	Group	Obj. Units	Meters
LOD Dist		Scn. Units	Meters

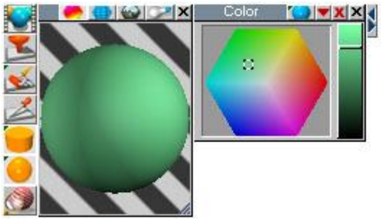
Expanded view of Object Information Panel

The red triangle in the upper right corner gives you access to a more extensive object information panel, and clicking on the red triangle again will collapse the object information panel back to its compact form.

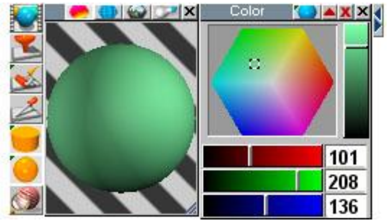
The properties displayed in the object information panel take two forms: information that is not changeable through this panel, and measurements that can be manipulated by typing directly into the fields in the panel. Any information in a white box is editable, while the information in gray boxes is not. For instance, you can change the location, rotation and size of an object by clicking in the box, typing a new number and hitting return, but you cannot change the class of the object through this panel. This panel is covered in detail **APPENDIX B: PREFERENCES**.

Panels

Panels open up and tile automatically from left to right along the bottom of the screen. However, you can move a panel to wherever you want by dragging its title bar. By default some panels will display a subset of its options to keep the panel small and manageable. However, when you need to adjust more details than are displayed by default, many panels will have a red triangle icon available on the menu bar that will display all the available settings.



Color panel minimized



Color panel expanded

After clicking the red triangle icon, it will change from pointing downwards to pointing upwards. Click the triangle again to roll the expanded options back into the panel, leaving the minimal panel display and an uncluttered desktop.

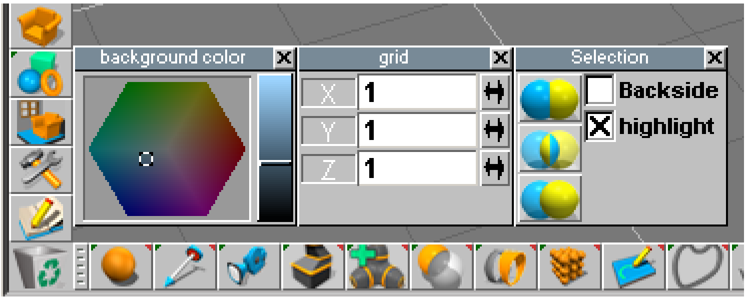
When working with many objects, materials and other tools, you may get a lot of panels open at one time and you may want a way to organize the clutter quickly. The following two tools will be invaluable in this case:

Close All Panels

This will close any panels that are currently open, leaving you with an unobstructed view of your workspace. If you would rather just hide all panels, press the TAB key. This will give you an unobstructed view of your workspace, and allow you to get all your panels back by pressing the TAB key again

Dock All Panels

Sometimes you can have a lot of panels open that you would just like to organize in a quick and logical way, and not close any of them. Use the Dock All Panels tool for this. Your panels will be lined up orderly along the bottom of your workspace.



All Model panels in docked state

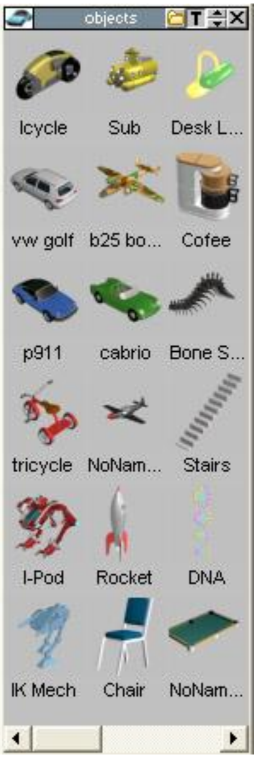
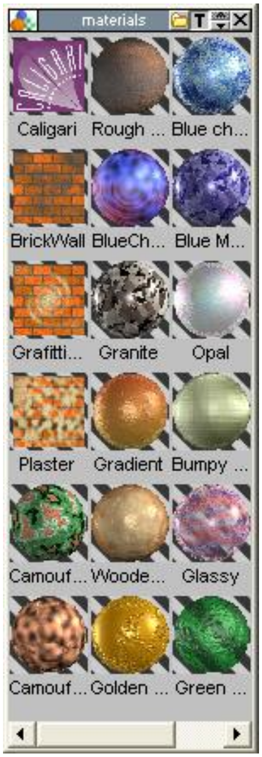
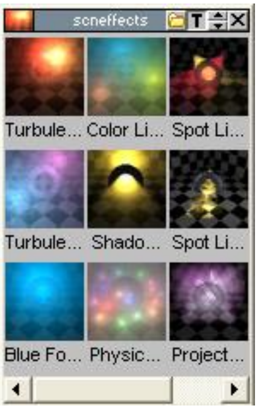
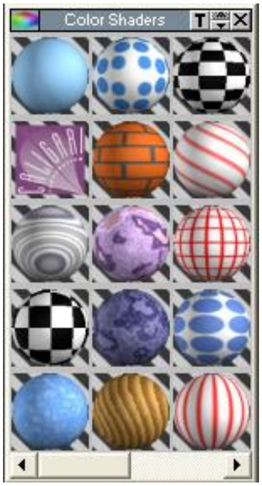
2.14.9 Projects, Libraries, Menus, and Files

Just like with toolbars, Model View has a duplicated library structure that in some ways parallels the newer Stack View Library structure. The reason for this again is to keep for existing users content created with previous generations of trueSpace in a familiar format. New trueSpace users may jump right in to new Stack View libraries. Over time more and more libraries will be moved to newer format.

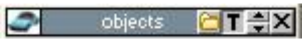
In most cases you can load/save objects to and from trueSpace scenes in either newer or older libraries. While you model and render it may be convenient to use the Model side libraries. Then, when you add some procedural behaviors or DX9 materials, you may decide to store the scene in the new format in Stack View library. What is great that even after you have added new additions to older scenes, it still can be saved in both formats! This is quite practical. For example, for an illustration project you can use the known and trusted older Model format, while for a game project you can save your content in the new Stack View format, and you still can move objects between both projects with ease.

Library Basics

Like Stack View libraries, **Model libraries** are visual representations of what is stored on your hard drive, so you no longer need to worry about the Windows file system. While you may still choose to load scenes and objects through a file list, you also have a visual method that is much quicker and less error prone. Each library and item has a file associated with it. When you create new libraries, trueSpace creates a new subdirectory off the *\libraries* directory. When you add items to the library, trueSpace adds files to the subdirectory. This is true for all user-created libraries, but some of trueSpace’s default libraries do not have directories associated with them because they are always a part of trueSpace.

 objects Icycle Sub Desk L... vw golf b25 bo... Cofee p911 cabrio Bone S... tricycle NoNam... Stairs I-Pod Rocket DNA IK Mech Chair NoNam...	 materials Caligari Rough ... Blue ch... BrickWall BlueCh... Blue M... Grafitti... Granite Opal Plaster Gradient Bumpy ... Camouf... Woode... Glassy Camouf... Golden ... Green ...	 sceneffects Turbule... Color Li... Spot Li... Turbule... Shado... Spot Li... Blue Fo... Physic... Project...	
Object Library	***Material Library***	***Scene Effects Library***	***Shader Library***

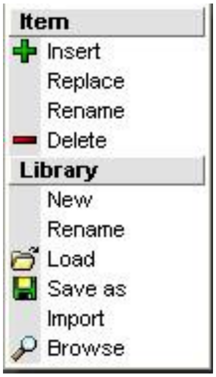
The library interface is a collection of panels that each contain items of a particular type, such as objects or materials. Each item is represented by an icon and a name. The panel can be moved by dragging the title bar, resized by dragging a corner or edge, and closed by clicking the corner 'x.' Each library panel also has three controls specific to libraries located on the title bar:



The up and down pointers increase and decrease of the size of the icons within the panel. The 'T' button will toggle item names on and off (the Shader library, above, does not display names). The small icons view can give you a more complete list but will not leave much room for item names. To select an item from a library, simply left-click the item to make it active, or drag and drop it directly into your scene.

In the case of objects, clicking on an item will create it at XY coordinates 0,0 (the center of your workspace).

Right-clicking within the library panel will call up a menu of actions used to control the contents of a library. This menu has several options depending on the context:



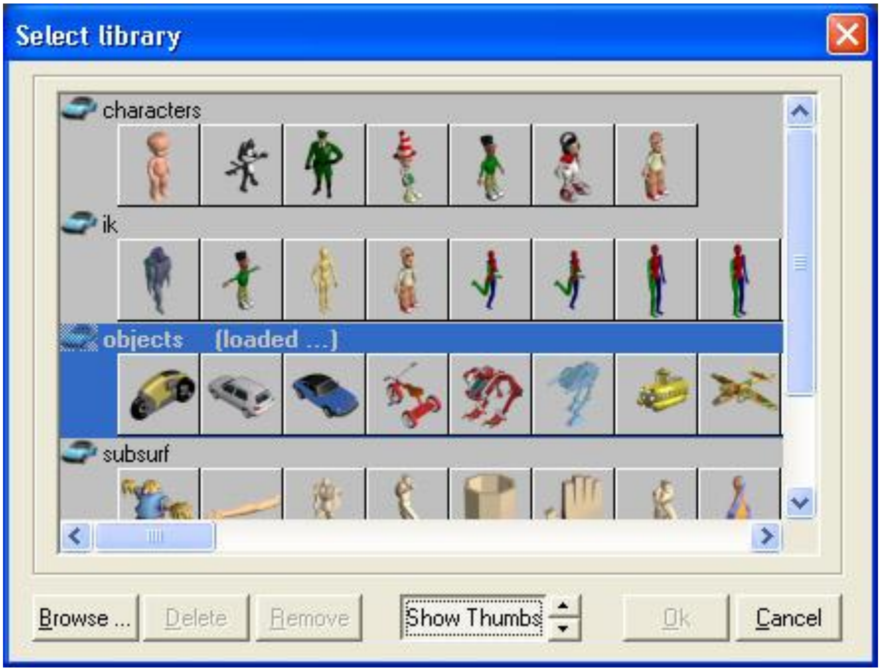
The menu is separated into two sections. The actions that affect the library's contents are under the Item heading. Actions that are specific to library file management are listed under the Library heading.

Item Menu Selections

- **Insert:** This will add the currently active item to the library and is context sensitive for the current library type. For example, if the currently active object in your scene is a sphere with a brick texture mapped to it, clicking Insert in the object library will add your sphere to the library along with the material map. On the other hand, clicking Insert in the material library will store only the brick texture.
- **Replace:** Replace is only available when you select an existing item within the library. This action will replace the library object with a copy of the currently selected object in your scene. The name of the object will be kept the same. This will also overwrite the object file in the library directory, so make sure you wish to replace your object before you choose this option.
- **Rename:** This will change the name of the selected item and name of the file stored in the library directory.
- **Delete:** This will remove the selected library item from the current library and delete the item from your hard drive.

Library Menu Selections

- **New:** New will create a new, empty library of the selected type, and a subdirectory under the libraries directory named new0. You may wish to rename this to something more suitable for the items you store in this new library.
- **Rename:** This renames the current library and its subdirectory in the Libraries directory on your hard drive.
- **Load:** The **Load** command will bring up a list of the libraries of the same type that have been imported or shared from other projects. Select a library and click Ok to load it as the active library. The Select Library dialog also contains the following features for working with loaded libraries:



  - **Browse:** Use this menu option to open and share a library from another project. Because the library is shared and not duplicated, any changes you make to a library loaded this way will affect the other projects that use the same library.
  - **Delete:** This command deletes the library from your hard drive.
  - **Remove:** This command deletes only the link to a shared library, without deleting the library from your hard drive.
  - **Show Thumbs:** This toggle displays or hides the thumbnails for the listed libraries.

- **Save:** This command saves changes made to the current library.
- **Save As:** This command saves a copy of the current library under a new name. A dialog box will appear asking you to name the copy.
- **Import:** Import copies a library from another project. Unlike sharing, which allows multiple projects to use and make changes to the same library, importing duplicates a library, copying it to the current project. (For information on sharing libraries, see the File:Load:Library command below.) Changes made to an imported library will not affect the project from which the library was imported.
- **Browse:** Use this menu option to open and share a library from another project. Because the library is shared and not duplicated, any changes you make to a library loaded this way will affect the other projects that use the same library.

Library types

Project Library

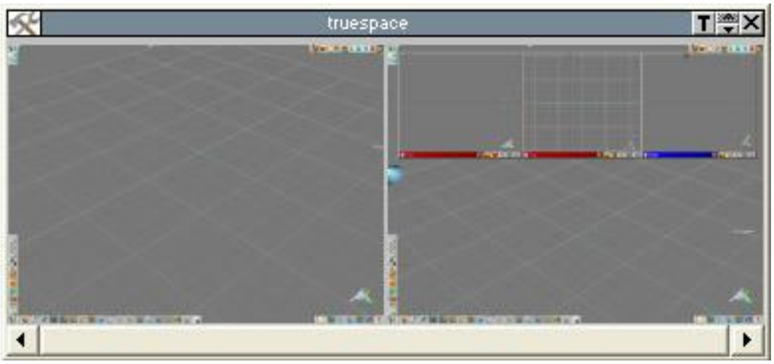
In real life you often work on a **project** basis. Projects in Model View consist of your workplace configuration, several libraries including artwork either imported or created, and your current scene. Your workspace state is automatically saved so that you find it the way you last left it when you start trueSpace again.

It is also very convenient to create new libraries while working on a project and drag objects, materials, etc. into them right from workplace. trueSpace will automatically name them for you so that you can continue working without interruption.

When you create a new project, trueSpace will create a new directory of the name that you specify under the \projects directory and store all information belonging to this project in subdirectories of that folder. You are also given the option to import the default libraries. As mentioned above, importing will create a copy of the library in the new project directory. This takes a bit more hard disk space than sharing, but greatly simplifies the maintenance of the artwork for your current project.

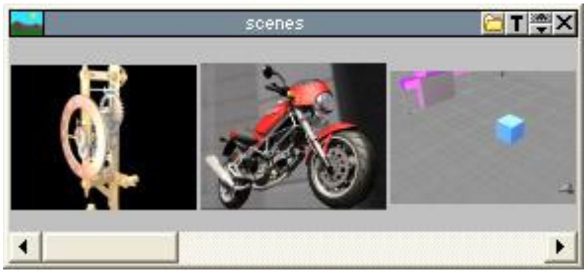
trueSpace Model View has a default project that is used as the working project if a new one if not created. This is called \truespac.tsp and is located in the root directory of the trueSpace installation.

Configuration Library



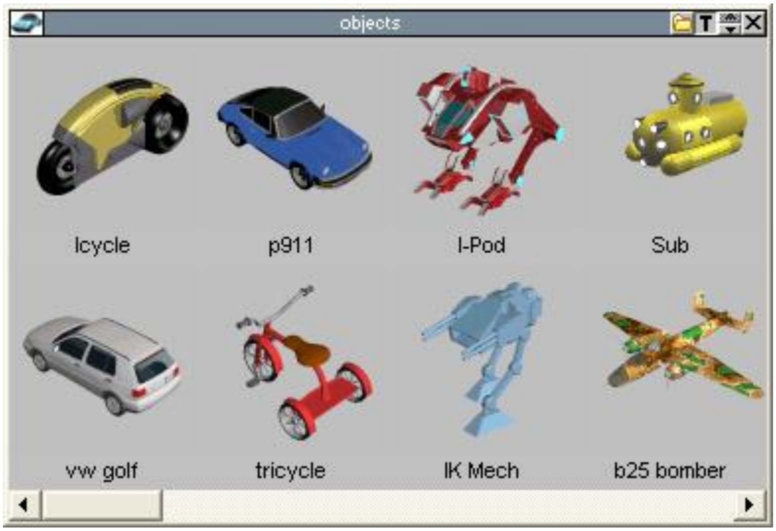
The configuration library stores toolbar arrangements, open panels, view windows and their positions, and real time render settings. This is useful for customizing your interface for the task at hand. An animation scene may have one layout, and a static image scene may have a completely different layout.

Scene Library



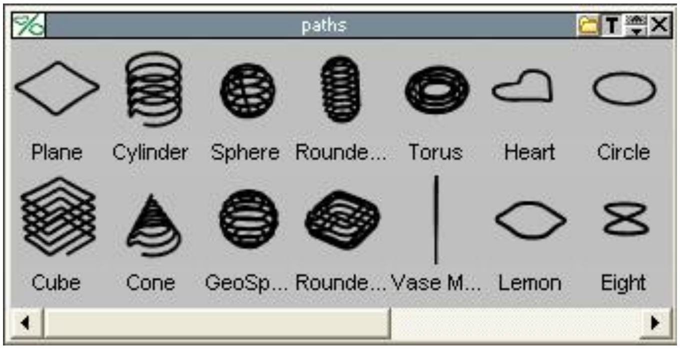
Scenes can also be stored in libraries. This allows you to browse your existing scenes using graphical representations. When you choose Insert from the library's pop-up menu, your current scene will be added to the library, and a preview render of it will be generated for use as an icon. Scenes saved via the file menu are not automatically inserted into this library. However, once you save a scene into the library, using the file menu to update your saved file will be reflected in the library. To save the scene automatically, see the preferences panel in **APPENDIX B: PREFERENCES**.

Object Library



The object library can be used to hold objects or backups of objects from your scene. Any object that you create or import can be stored in an object library.

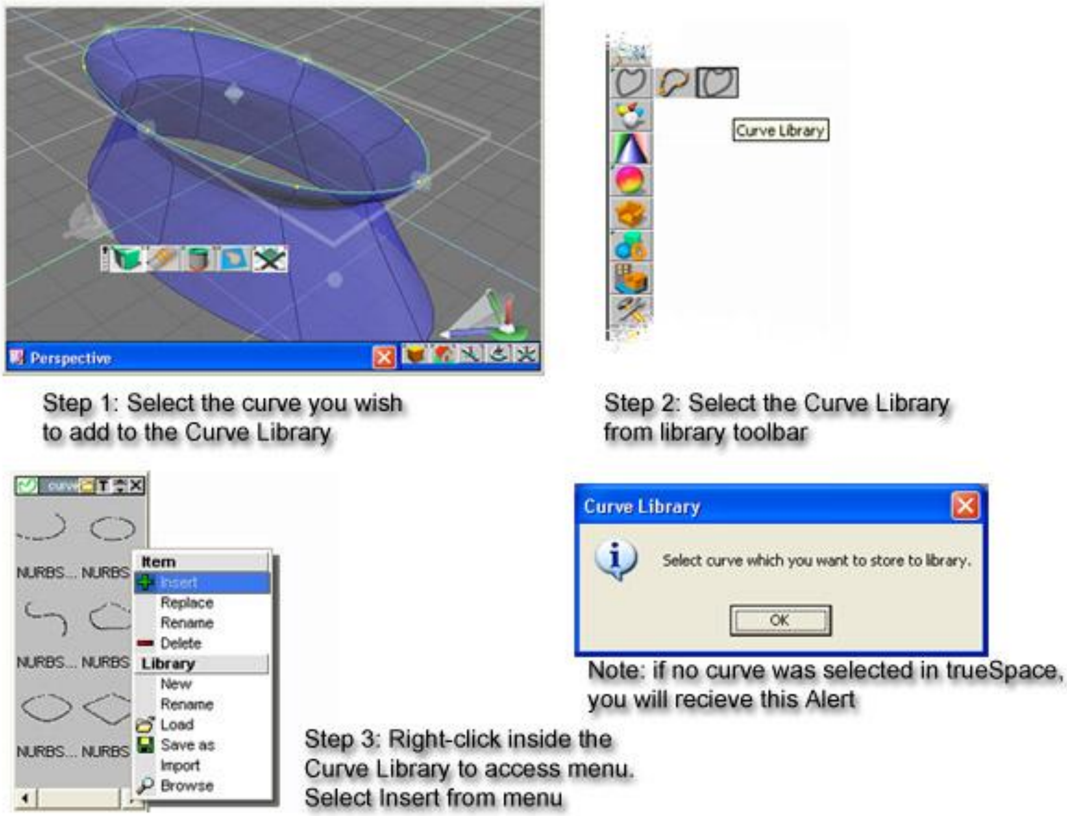
Paths



This library stores spline paths to be used for animation, sweeping, and other path-based functions. trueSpace4 path libraries imported into this library will be saved in the new format.

Curves

The curve library allows you to select a curve from the draw panel or from an existing object and save the curve to the library. The illustration below shows the steps involved in adding a curve to the new Curve Library.



Step 1: Select the curve you wish to add to the Curve Library

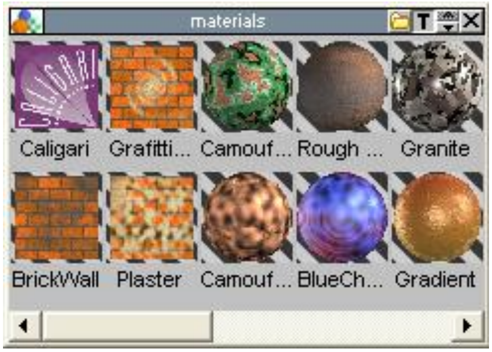
Step 2: Select the Curve Library from library toolbar

Select curve which you want to store to library.

Note: if no curve was selected in trueSpace, you will recieve this Alert

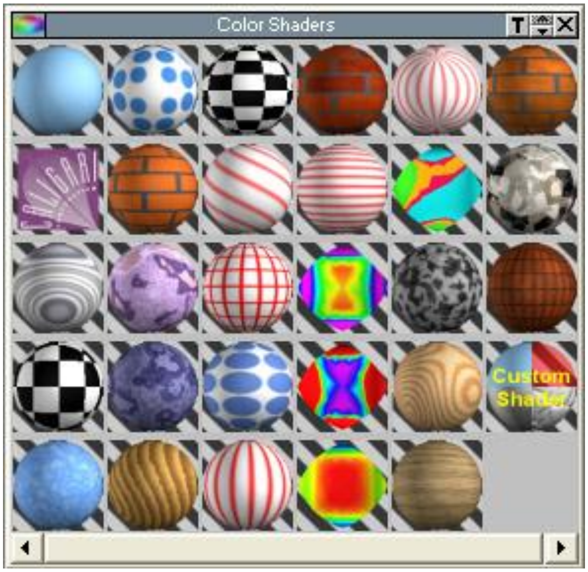
Step 3: Right-click inside the Curve Library to access menu. Select Insert from menu

LightWorks Materials



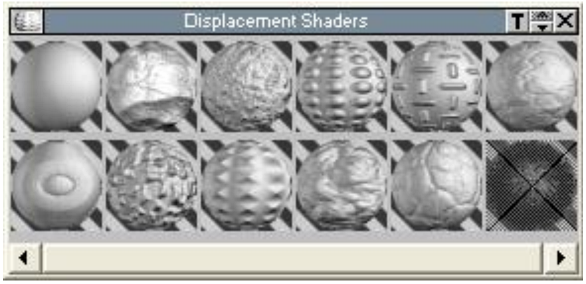
A LightWorks material is a combination of **color**, **transparency**, **reflectance**, and **bump** shaders into an overall "look." For example, you may define a brick material by using a brick image map in the color shader, a rough texture in the bump shader, and a low-gloss finish in the reflectance shader. This combination can be saved as a single item into the material library and recalled for later use. Materials can be saved into the library by either inserting them through the pop-up menu, or dragging and dropping from an object from the workspace. (Note that you can drag a material from any object in the scene, not limited to the selected object.) Material libraries from trueSpace4 or earlier (*.MLB files) will be imported and saved under the new library format when they are opened in trueSpace.

Color/Procedural Shaders



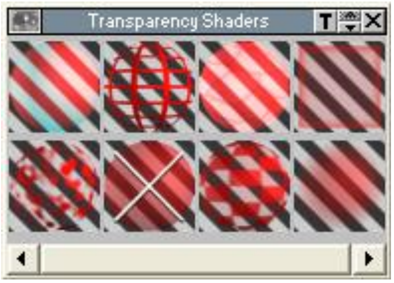
Color shaders are similar to textures except they are based on procedures and algorithms rather than existing images. This library stores color shaders and their settings. You cannot add items to this library through trueSpace, but new shaders that have been installed from outside of trueSpace will automatically show up here.

Displacement Shaders



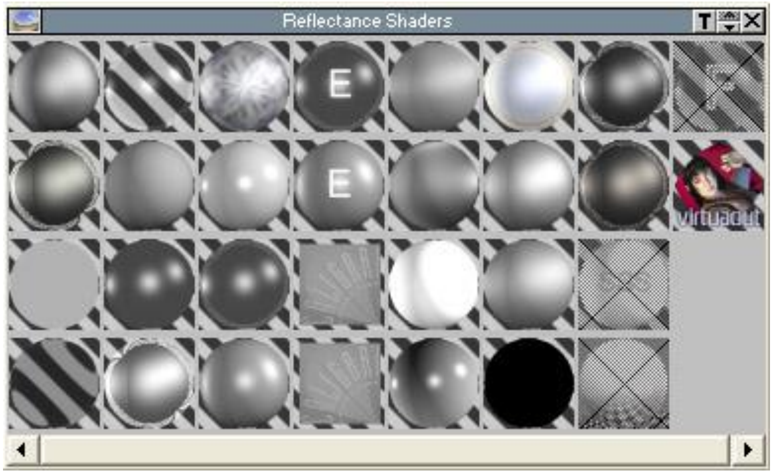
Displacement shaders define how uneven the surface of your material appears. As with the color shader library, you cannot add to this library from a scene, but externally installed shaders will show up here.

Transparency Shaders



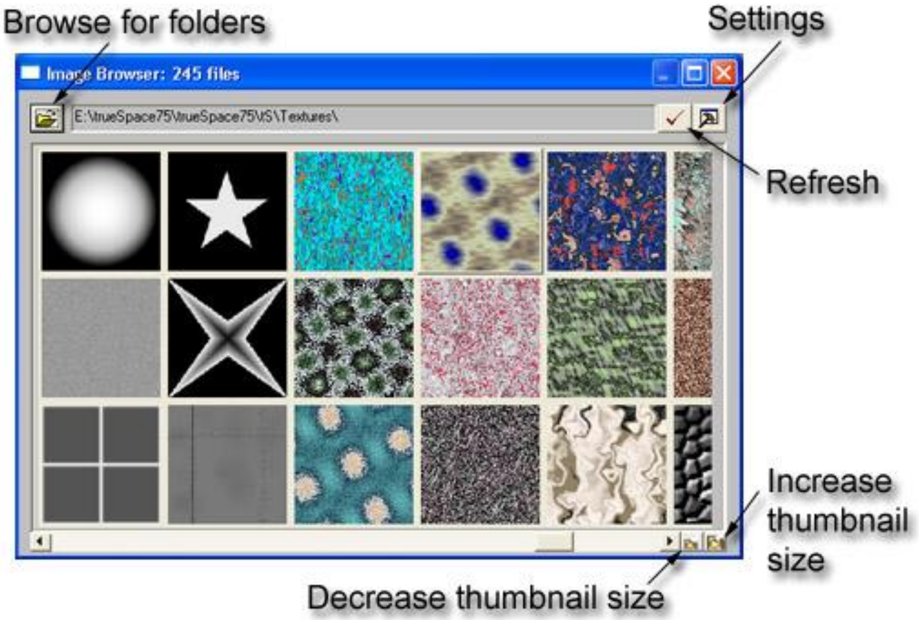
Transparency shaders define procedures that give your materials transparent qualities. trueSpace supplies an assortment of transparency shaders for you to use. As with the color shader library, you cannot add to this library from a scene, but externally installed shaders will show up here.

Reflectance Shaders



Reflectance shaders define procedures that simulate a range of reflective materials, from flat to mirror-like. As with the color shader library, you cannot add to this library from a scene, but externally installed shaders will show up here.

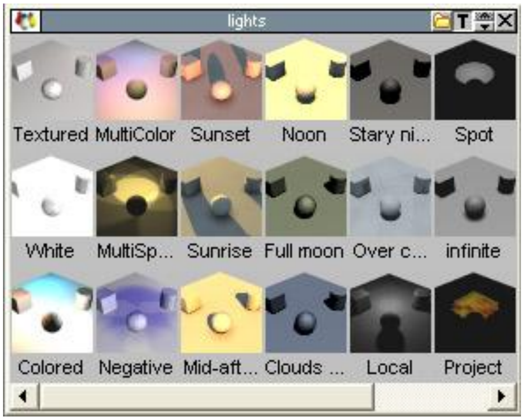
Image Browser



E:\trueSpace75\trueSpace75\tS\Textures\

The Image Browser allows you view directories of images using thumbnails instead of filenames. Images can be dragged and dropped directly onto main preview, color preview, or bump preview of the **Material Editor**. (See **CHAPTER 6: SURFACING** for more information on the Material Editor.) Many properties panels within trueSpace that contain the option to load an image allow you to call up the Image Browser by left-clicking the option's entry box. (Right-clicking the same box will allow you to use the standard Windows file browser instead.)

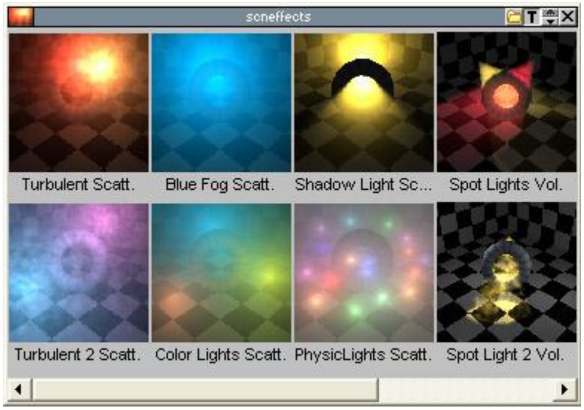
Lights



Custom light arrangements can be saved and reloaded at a later time. This lets you create specific light configurations that can be easily duplicated in many scenes, and also lets you experiment with lighting sets without risk of losing saved settings.

Note: Lights residing on locked or hidden layers do not get replaced when a new light setup is selected from the Lights library.

Scene Effects Library

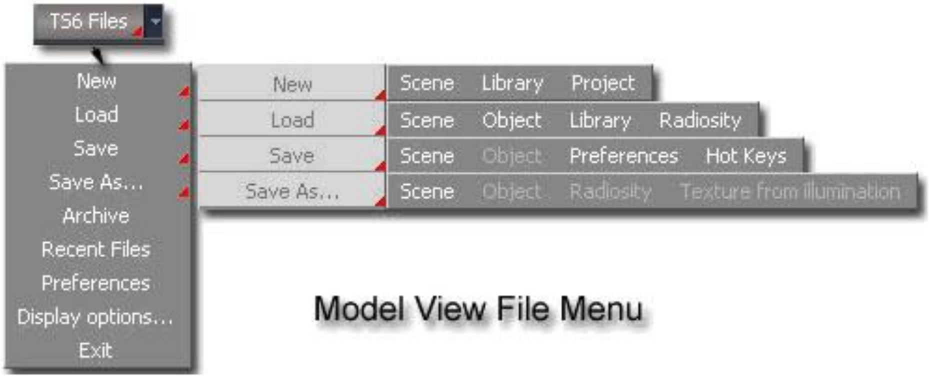


This library stores foreground and background shader settings. trueSpace comes with several sample scene effects, and you can add more as you create your own setups.

Menus and Files

Although most of the actions in trueSpace can be accomplished through icons, there are two menus available: File and Help. The File menu has options for loading and saving objects, scenes, and libraries, as well as access to the preferences panel, display options and optimizations for processor instruction sets. See **APPENDIX B: PREFERENCES** for more information on the preferences panel.

File Menu



New: Scene
This removes all objects from your Model View scene and starts you off with a fresh workspace. If you have not saved your current work, trueSpace will first prompt you to save your work.

New: Library
This will present you with a list of library types that you can create. Each of these already exists in the default trueSpace toolbars, and by using this menu option you can create empty libraries of your own. They will be attached to the library toolbar by default.

New: Project
This will present you with a dialog where you can name your new project. Once you have named it and clicked OK, trueSpace will create a new project folder for you in the \project subdirectory. A project will store all your settings from your trueSpace session.

Load: Scene
This will allow you to load a previously created scene. Loading a new scene will also unload the current scene you are working in. If you have not saved your work you will be prompted to do so. trueSpace can read .SCN, .WRL, and .WRZ files.

Load: Object

trueSpace can import many different formats of objects. The native trueSpace object format is .COB, but others are available. Opening an object adds it to your scene.

Load: Library

Use this menu option to open and share a library from another project. Because the library is shared and not duplicated, any changes you make to a library loaded this way will affect the other projects that use the same library. They will be attached to the library toolbar by default.

Load: Radiosity

Use this option to load a saved radiosity solution. For more information on radiosity, see **CHAPTER 7: LIGHTING AND RENDERING**.

Save: Scene

trueSpace stores a scene as a file with a .SCN extension by default. However, it can also save a scene in iSpace (.ISS) and VRML (.WRL) formats. If your scene has been previously saved and already has a name, you will not be prompted for a new name. If the scene has not been saved before, you will be prompted to give it a name and choose the format.

File: Save: Object

This will save your currently selected object to its own file. By default trueSpace saves to the .COB format, but many other formats are available for exporting. (See **DEVELOPER'S GUIDE CHAPTER 10: FILE FORMATS** for more information on the available file formats.)

Save: Preferences

This command saves any changes made to the Preferences during the current working session. Preferences are normally saved when exiting trueSpace.

Save: Hot Keys

This command saves the Hot Keys set using Icon Helper during the current working session. Hot keys are normally saved when exiting trueSpace.

Note: Hotkeys are stored in the trueSpace.key file. You can bring in your old hot keys by copying this file to your trueSpace directory. The file keylist.txt that is saved from the **Icon Helper** is a simple text file that can be printed out for reference, and is not used by the program to set hotkeys.

Save As: Scene or Object

Use the Save As menu item when you wish to save an already named scene or object to a new file, leaving the original one untouched, without any updates since you first loaded it, or last saved it.

Save As: Radiosity

Use this option to save a radiosity solution that has been calculated for the current scene. For more information on radiosity, see **CHAPTER 7: LIGHTING AND RENDERING**.

Save As: Texture from Illumination

If any objects in the scene have textures generated from lighting, this option can be used to export the scene as either a regular trueSpace scene or to web formats that support baked on textures, such as Viewpoint and Shockwave 3D. See **CHAPTER 7: LIGHT&RENDER** for more information on creating textures from illumination.

Archive

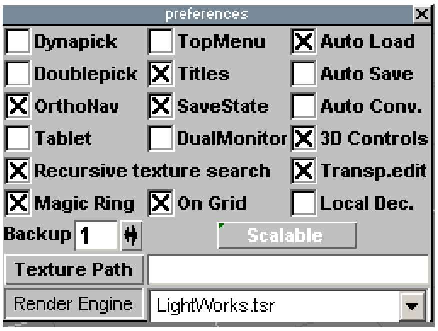
The archive function stores all resources in your scene, such as objects and materials, to its own directory under the *\archive* subdirectory. If the current scene has already been named, a folder inside the archive folder with the scene name will be created. Otherwise, the default name “newscene” is used for the scene and folder. This is an easy way to back up a scene and all its resources to one location.

Recent Files

This provides a list of the most recent files loaded or worked on for quick access.

Preferences

This opens the trueSpace Model View preferences panel:



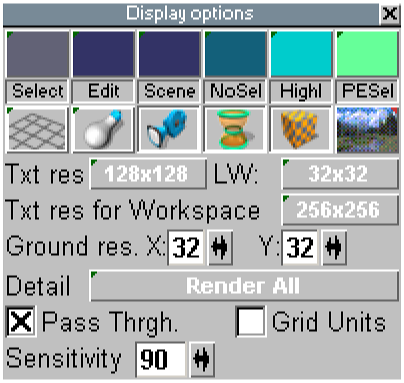
preferences

☐ Dynapick ☐ TopMenu ☒ Auto Load
☐ Doublepick ☒ Titles ☐ Auto Save
☒ OrthoNav ☒ SaveState ☐ Auto Conv.
☐ Tablet ☐ DualMonitor ☒ 3D Controls
☒ Recursive texture search ☒ Transp.edit
☒ Magic Ring ☒ On Grid ☐ Local Dec.
Backup 1 Scalable
Texture Path
Render Engine LightWorks.tsr

The options available on this panel affect the overall appearance and functionality of trueSpace. This panel and its options are covered in detail in **APPENDIX B: PREFERENCES**.

File: Display options

This brings up a panel for setting real-time display options. This panel is also available by right-clicking on the **Draw Objects** icon and is covered in detail in **APPENDIX B: PREFERENCES** .



Txt res 128x128 LW: 32x32

Txt res for Workspace 256x256

Ground res. X: 32 Y: 32

Detail Render All

☒ Pass Thrgh. □ Grid Units

Sensitivity 90

File: Exit

This will exit the program. If *AutoSave* in the preferences panel is enabled, trueSpace will automatically save your current project and the scene. Otherwise, it will prompt you to save if any changes have been made since the last save.