Publisher Documentation

Last modified 02/02/28 16:11:59 cw

Ton Roosendaal

Carsten Wartmann

Reevan McKay

V1.05 for Blender Publisher 2.25 Edition

Publisher Documentation: Last modified 02/02/28 16:11:59 cw
by Ton Roosendaal, Carsten Wartmann, and Reevan McKay

V1.05 for Blender Publisher 2.25 Edition

Table of Contents

List of Tables

List of Figures

Chapter 1. Overview of Publisher products

As a special version of the Blender Creation Suite, the Blender Publisher will support professionals to publish content to the web and other media such as CDROM, utilizing the Blender stand-alone player or the Blender 3D Plug-in.

The 3D plug-in will be published by NaN for free download. However, usage of this program, or of the stand-alone Player, will be limited both by functionality and by the license. Commercial and professional usage will require purchasing the Publisher product.

Blender Publisher gives you access to the products described briefly in the next sections.

1.1. The Key

This is a small file that has to be installed on each system the owner is licensed to. Without this Key file, the special Blender Publisher features won't work. A Key file contains information about the owner and compression and encryption codes. Therefore, it is important to fill in the correct ownership name (company or individual) when you purchase the Publisher. Ownership cannot be transferred to others. For site licenses, only one key is generated by NaN. This is to enable groups working together, with the same encryption/decryption codes and identical signatures. The key will be valid for all Blender Publisher 2.xx releases.

1.2. Blender Publisher

This is a special version of Blender Creator with additional features such as compression, locking, signing and run-time generation. The Blender Publisher needs a Key file installed in order to access these features. The Blender Publisher will be available via a separate download section on www.blender3d.com, accessible with the e-shop password only. All current OS versions of Blender Creator will be available as Publisher as well.

1.3. Blender 3D Plug-in

Files written with Blender Publisher are optimised for usage in the Blender 3D plug-in. The plug-in then will not display the Blender logo and URL as overlay, and allow for a custom 'loading' animation. For internet usage, the plug-in is available for free download via the Blender website. For other appliances, such as using the Plug-in with ActiveX control to integrate with PowerPoint, Word, Visual Basic, or Macromedia Director, the Blender Publisher license grants the owner the right to redistribute the 3D plug-in.

1.4. Stand-alone Player

To be able to deliver interactive 3D media to clients for example on CDROM, the stand-alone player allows distributing Blender files with additional graphical options such as full-screen mode.

The stand-alone Player will only play back compressed, signed or locked files that are created with Blender Publisher. It will not play back files created with Blender Creator.

1.5. Publishing License

Blender Publisher owners have the right to redistribute or resell products which include the Blender stand-alone Player and/or the Blender 3D Plug-in via CDROM, Internet, or bundling in with hardware.

Depending on the license purchased, a limit to the maximum number of copies may apply per media production using Blender player technology.

1.6. Extras & updates

In the coming months, additional features and support for Blender Publisher have been planned, which will be available as free updates:

- new optimised, lossy file format; for smaller files and faster loading.
- Installer & help product; to help resolving h/w related playback problems
- Apple OS 8/9 players

As a separate project, for a major upgrade, research will be started on:

- Streaming technology in Publisher files: for geometry, texture maps and audio files.
- Engine support for external streaming media such as video and mp3
- Multi-player environments

Information on the status of this new development is expected to be available Q1 2002.

1.7. Support & documentation

Via a closed section on the www.blender3d.com site, all relevant support systems, documentation and software updates will be made available. For Publisher related questions, personal email support is available at mailto:publishersupport@blender3d.com. It is NaN's aim to listen carefully to feedback of Publisher clients via these channels; to improve the product, get bugs solved and new features implemented.

Chapter 2. Installation

The installation consists of a few easy steps, covered in detail in the next sections.

2.1. Get the Key file

After you got payment confirmation for the Publisher, you can download the Key file from the link given you by the e-shop.

Please note a few important issues first:

- The first time you download the Key, you will see a web-page displaying the license agreement. By loading the Key, you will agree to the license.
- When you do not agree on the license, do not load the Key but send us an email instead (mailto:shop@blender.nl). We then can cancel the order and you get the money refunded.
- The Key file will be generated only once, and then stored in a secure database. You can get a Key file several times, but it will remain the same Key.
- Only the original buyer has access to the Key that way, but it implies that you keep the shop password and login a secret.

2.2. Get Blender Publisher

At the Publisher download page, you can load the Blender Publisher versions you wish to use. Since all features of the free Blender Creator are also available in the Publisher, there's no need to get the freeware version as well.

2.3. Unpack and install Blender Publisher

As with the Creator releases, Blender Publisher is compressed and packed in a way that suits the OS version. The Windows version contains an auto-installer, assigning the appropriate file types.

2.4. Install the Key

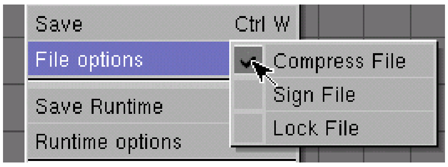
Start up the Blender Publisher. In the File menu (top bar, left) you can find a new option: "Install Key". This invokes the typical Blender file-selecting window, browse to the location where you saved the Key, select the Key file with LeftMouse, and press the button "LOAD". This action will make Blender copying the Key to its default help file directory.

When the install went right, you get a new splash-screen, a new ’P’ icon in the top header, and extra options related to the Publishers functionality.

Chapter 3. Blender Publisher

3.1. File features

Figure 3-1. File features



Blender Publisher supports a new file format, ready for future expansion to streaming and optimising. Currently, the following new options are available:

Compression

- The entire file is compressed. This will compress regular Blender files down to as low as 25% of its original size.
- Please note that files which include packed jpeg images will compress less efficient.
- Compressed files are readable by all Blender Creators and players. So you can send a compressed file to someone who does not own a Publisher (Blender V2.22 or later)
- Compressed files are especially useful for playback in the 3D Plug-in or with the stand-alone Player.

Locking

- Locked files are encrypted using the owner's Key file.
- Locked files can be read by any Blender 3D plug-in or stand-alone player
- As additional feature, locked files can still be read back by the original owner only
- Technical details about locking can be found in Chapter 4

Signature

- adding a signature to a file will prove ownership of a file. It then includes the owner information as stored in the Key.

- Signed files can be read by all Blender versions V2.22 or later. Reading a signed file will pop up a splash screen displaying owner information
- on save-over the signature disappears (or is replaced with another signature).

3D Plug-in support

- Publisher files will be displayed without the default overlay logos in the plug-in
- A Publisher file can be used to create a custom ’loading’ animation.
- This implies that you use (one of) the new options compression/locking/signing, otherwise Blender Publisher will write a regular Blender Creator file.

Stand-alone player support

- The publisher installation contains a stand-alone player. This player only accepts files written with Publisher using one of the new file features

For one of the next releases, a new file type will be supported that will gain an even further compression and is optimised for fast loading in a Blender player. This format is nicked the ’.blip’ format. It will not contain any information that is required for editing a file, so it cannot be read back by any Creator or Publisher.

Table 3-1. Overview of file formats: (R=read, W=write, X=not supported)

Blender		Creator	Player	Plug-in	Publisher
xxx.blend	Compressed	R	R	R	R/W
Signed	R	R	R	R/W	
Locked	X	R	R	R*/W	
Regular file	R/W	X	R**/W	R/W	
xxx.blip	Optimized, lossy compression	X	R	R	W

*Read only for the owner who wrote it.

**The Plug-in reads regular Blender files with a default loading animation and overlay logos.

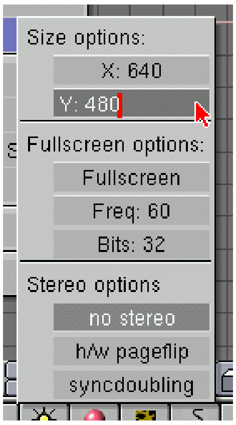
3.2. Run-times

Blender Publisher can write a run-time executable file. In fact it is the stand-alone player with a Blender file included. Versions for Mac OS8/9 are being worked on.

Run-times are primary aimed at distribution via CDROM or bundling in with applications. Sending run-times via e-mail is strongly discouraged, since most receivers will assume it's a virus.

You can write static or dynamic linked run-times. Static run-times will contain all needed to be run, for dynamic linked run-times you need to provide the python.dll, enabling you to deliver customized Python libraries with functions or modules you need.

Figure 3-2. Runtime options



X: 640
Y: 480
Freq: 60
Bits: 32

Size options:

- X, Y: size of the window or screen in pixels

Fullscreen options:

- Fullscreen: makes the run-time fullscreen, if not selected, a window will be opened
- Freq: Monitor refresh frequency for Fullscreen
- Bits: color depth for fullscreen display

Stereo options (see also Section 3.3):

- no stereo: disable stereo view
- h/w pageflip: use the "h/w pageflip" method to display stereo
- syncdoubling: use the "syncdoubling" method to display stereo

The Windows runtime comes in two versions. The normal runtime will be distributable as a single file which will run out of the box. The dynamic runtime will enable you to add your own Python extensions but you'll need to distribute the Python DLL as well. Use the dynamic runtime if you want to use aditional Python extensions but it requires that your distribution comes on a CD or with a custom installer.

3.3. Stereo for the Blender Player

With the Blenderplayer it is possible to get a stereo view (using LCD-shutterglasses).

From the users perspective all he or she needs to do is to enable stereo from within the publisher file menu (see Section 3.2. This setting, along the the type of stereo which will be described later, will then be invoked when the file is saved as a runtime stand-alone application. Also, one can enable or disable stereo for the stand-alone player by specifying the desired stereo option on the command-line.

Two types of stereo are now implemented:

"h/w pageflip" stereo which only works if the driver (windows and Unix) supports this type of stereo rendering. This is usually called quad buffered stereo. Of course your glasses need to support this type of stereo.

"Sync doubling" stereo which is a cross platform solution for stereo. Again, of course your glasses need to support this type of stereo.

❗ Note: before trying this read the remarks below, it may damage your monitor!

Both stereo techniques have their advantages and disadvantages. To name just a few:

Window vs fullscreen

- h/w page flip makes it possible to have stereo in a window
- Sync doubling only works for fullscreen applications

Screen resolution and refresh

- h/w page flip doesn’t mess with your screen resolution and refresh rate. Everything is handled by the driver of the graphics system.
- Sync doubling doubles, as the name says, the vertical frequency. At the same time the vertical resolution is divided by 2 which has of course visual effects. Doubling the frequency might not be healthy for your monitor but most modern monitors will handle this (just like for instance setting a too high frequency with MS Windows). It is a matter of finding the boundaries of your monitor. But you are warned and NaN Technologies will not take any responsibility if things go wrong.

Using your card driver to get stereo

Another way to get a stereo view for GeForce cards is to use the Geforce Tweak Utility (http://www.geforcetweak.com/). Support for other drivers (i.e. the NVIDIA stereo drivers) and operating systems will follow.

Steps to get stereo

- get recent drivers for your nvidia card from the NVIDIA drivers page (http://www.nvidia.com/view.asp?PAGE=driver) and install them
- get the Geforce Tweak Utility (http://www.geforcetweak.com/) and enable stereo in the tweak utility. Be sure that you enable and set the 'StereoConvergence' and 'StereoSeparation' in the Geforce Tweak Utility to some reasonable value.
- Make a fullscreen stand-alone with Blender Publisher

General links about stereo view

- 3D Stereo guide from NVIDIA (http://www.nvidia.com/view.asp?IO=IO_20010614_4380)
- http://www.stereo3d.com
- http://www.stereovision.net/
- Discussion board about the GeforceTweak utility (http://www.guru3d.com/forum/list_forum.php?forumid=ubb10)

3.4. Documentation of extra functionality and updates in Blender Publisher

Blender Publisher contains some advanced funcionality only avaible for Publisher Key owners. These functionality is described in this section.

3.4.1. Local axis rotation

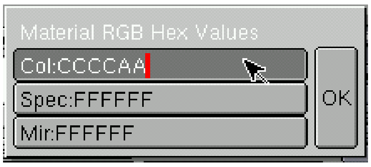
When you enter rotation mode by pressing **RKEY** Blender rotates the objec by default around the views Z-axis (). Pressing **XKEY YKEY** or **ZKEY** switches to rotation around the world axes.

Double tapping **XKEY YKEY** or **ZKEY** in rotate mode allows rotation around an object's local axes.

3.4.2. HTML compatible color input

NKEY in LampButtons, MaterialButtons, WorldButtons or PaintFaceButtons allows setting of the RGB colors using a HTML-compatible six-digit hex value.

Figure 3-3. HTML color input



Material RGB Hex Values

Col:CCCCAA

Spec:FFFFFF

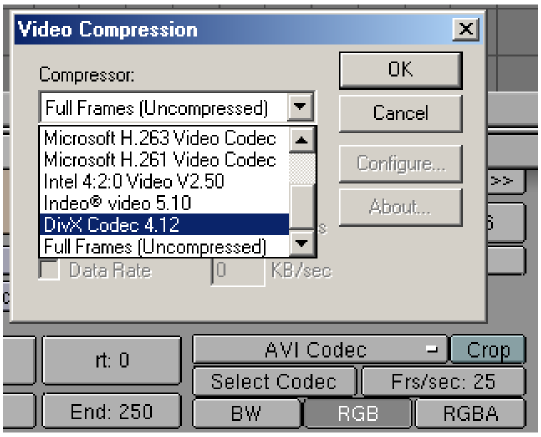
Mir:FFFFFF

3.4.3. Support for installed AVI codecs under windows

Blender Publisher on Windows now supports encoding and decoding AVI files using the localy installed AVI codecs. You can render directly to an AVI file, play back an AVI file or use an AVI file as an animated texture in the renderer.

If you want to render to an AVI file, go to the render buttons (press F10 key) and select ’AVI Codec’ from the image type menu. If this is the first time you selected ’AVI Codec’ in this Blender file, a window will pop up where you can select which codec you want to use and, if the codec allows it, set additional variables.

Figure 3-4. Choosing a installed AVI codec



Video Compression

Compressor: Full Frames (Uncompressed)

Data Rate 0 KB/sec

rt: 0

End: 250

AVI Codec

Frs/sec: 25

Once you selected the codec, the ’Quality’ button just below the image type menu, changes into a ’Select Codec’ button. Press this button if you want to select another codec or readjust the codec settings later on.

If you want to use an AVI file as a texture for rendering, go to the texture buttons (press F6 key) and add a new texture. Now first press the ’Image’ and then the ’Movie’ button. Next click on the ’Load Image’ button, browse to the AVI you want to use as a texture and select it. Finally enter the number of frames you want to use in the ’Frames’ button.

Current limitations

You can only use AVI files that are stored on your local hard drive. At this moment it is not possible to use a packed AVI file (a file that's stored inside the Blender file itself) as a texture.

If you want to render your Blender file on another Windows system, you will have to make sure that the correct AVI codecs are installed on that system to enable reading and / or writing the AVI files.

If you render a Blender file that tries to ouput images using a Windows codec on another platform (Linux, SGI, etc.), Blender will print a warning message and will render an uncompressed AVI instead.

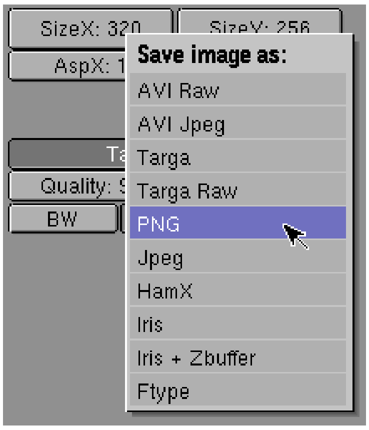
You can't rerender a part of an AVI. If a couple of frames are wrong in your animation you will have to rerender the entire AVI or use an external tool like virtualdub to edit the AVI's.

You can't use an AVI as an UV texture in the Blender real-time engine. The Blender interface allows you to use an image from an allready opened AVI as a UV texture, but the next time you open the Blender file the AVI will fail to load.

3.4.4. PNG file format

The Portable Network Graphics (http://www.libpng.org/pub/png/) (PNG) format was designed to replace the older and simpler GIF format and, to some extent, the much more complex TIFF format. It has two major uses: the World Wide Web (WWW) and image-editing.

Figure 3-5. PNG filetype



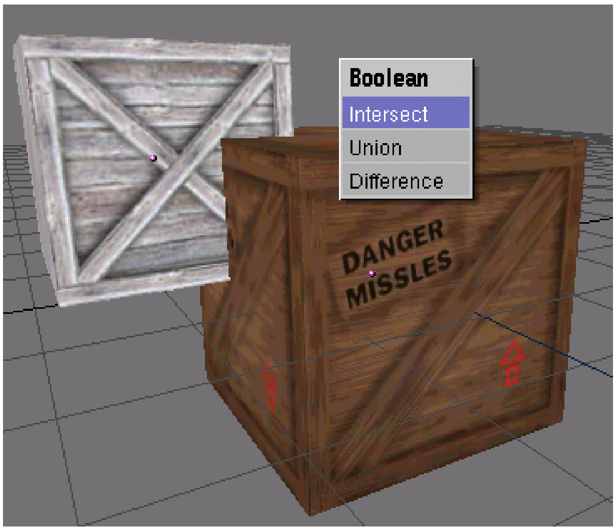
To choose the PNG format go to the DisplayButtons **F10** and choose "PNG" from the drop-down. To have a rendered Alpha-Channel in the image, select also "RGBA" in the DisplayButtons.

3.4.5. Boolean operations (WKEY)

The boolean operations will work for all objects but is really intended for use with solid closed objects with a well defined interior and exterior region. In the case of open objects the interior and is defined in a rather mathematical way by extending the boundary faces of the object off into infinity. So results may be unexpected for these objects. A boolean operation never affects the original operands, the result is always a new blender object.

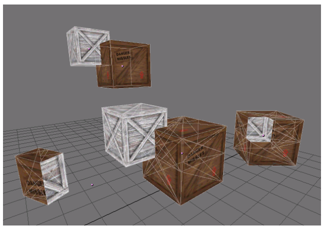
The boolean operations also take Materials and UV-Textures into account, producing objects with material indices or multi UV-mapped objects.

Figure 3-6. Options for boolean operations



For the "Difference" operation the order of selection is important. The active object (light purple in wire-frame view) is subtracted by the selected object.

Figure 3-7. Resulting objects, original top, intersect, union, difference



This functionality is currently under heavy development. To make it possible for you to use this functions for your work we list here the current limitations:

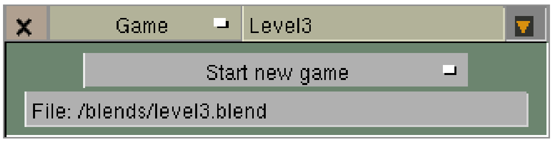
Known problems of the Boolean Operations

- The number of polygons generated can be very large compared to the original meshes. This is especially true for complex concave objects
- Output polygons can be of generally poor quality, meaning they can be very long and thing and sometimes very small, you can try the Mesh Decimator (EditButtons **F9**) to fix this
- Vertices in the resulting mesh falling on the boundary of the 2 oirignal objects do not match up
- Boundary vertices are duplicated. This is good in some respects because it means you can select parts of the original meshes by selecting one vertex in the result and hitting the select linked button (**LKEY**) in Blender. Handy if you want to assign materials etc to the result. To get rid of the doubled vertices use the "Remove Doubles" button in the EditButtons **F9**.
- The boolean operation can fail, a message is popped up saying ("An internal error occurred -- sorry"). Try to move or rotate the objects just a very small amount.
- Operations are between two objects only, there is no way to perform operations on more than 2 operands, such as intersect all these selected objects with the active object.

3.4.6. Game Actuator

The Game Actuator that allows you to quit or restart a game, or start a game from a different file from within the 3D real-time engine, the stand-alone player or the 3D Plug-in

Figure 3-8. Game Actuator



Game

Level3

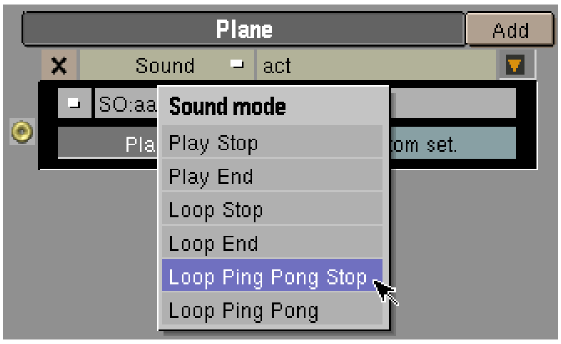
Start new game

File: /blends/level3.blend

3.4.7. Enhanced audio support

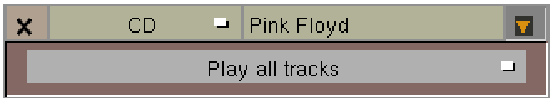
The Windows version of the Blender Publisher and the ActiveX control now uses the fmod library (http://www.fmod.org/). This enables you to use Ogg Vorbis audio (a free very good compressing format) and a wider variety of sound formats. Beside that the library is very fast (meaning less CPU cycles used).

Figure 3-9. Sound Actuator play modes



The Sound Actuator (see Figure 3-9) now supports two new play modes for playing sound in a Ping-Pong mode. This plays the sound first forward, then backwards, which is great for environment sounds.

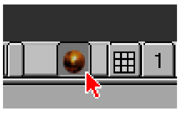
Figure 3-10. CD Actuator



The CD Actuator that enables you to play back tracks from a CD on Windows sytems. It includes control of the CD volume and pausing/resuming the playback.

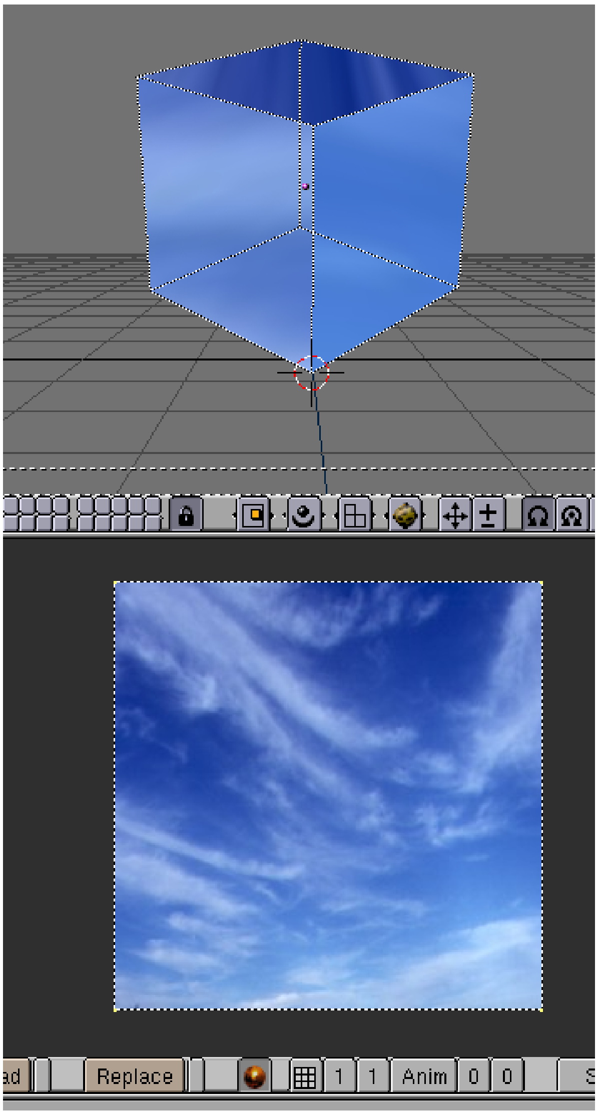
3.4.8. Reflection mapping in the realtime engine

The reflection mapping is enabled with the sphere icon in the ImageWindow (UV Editor).



The effect can be seen in the textured views and in the game engine.

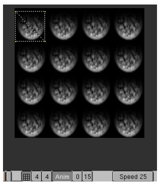
Figure 3-11. ImageWindow with texture and the textured view



3.4.9. Animated Textures

With Blender Publisher 2.25 it is possible to use simple animated textures as UV textures.

Figure 3-12. Animated texture in ImageWindow



This feature uses tiled images to hold the animation frames. Activate the tiles-icon in the ImageWindow to tell Blender you want to use this feature, then define with the NumberButtons the dimensions and press "Anim". The next NumberButtons define from which to which tile the animation will run. "Speed" defines the playback speed in frames per second.

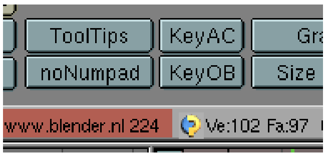
3.5. Character Animation Tools Documentation by Reevan McKay

3.5.1. General Tools

Auto-key

The auto-key feature can be found in the info bar. When it is enabled, blender will automatically set keyframes when you move objects. This is helpful for people who are not used to explicitly inserting keyframes with **IKEY**. There are two separate toggles for auto-keying: one for object mode and one for pose mode. These two options can be set independent of one another.

Figure 3-13. Auto key options



For Objects

“KeyOB” will set keyframes for objects that are moved in object mode. Users who are familiar with the blender interface will likely want to leave this option disabled.

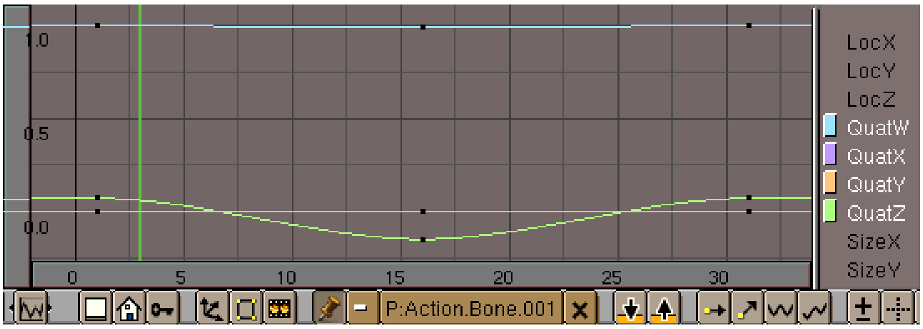
For Actions

“KeyAC” sets keyframes for transformations done in pose mode. This ensures that you will not lose a pose by forgetting to insert keyframes. Even users who are familiar with the blender interface may find this to be a useful feature.

Ipo/Action Pinning

It is now possible to display different ipos in different windows. This is especially valuable while editing actions, which have a different ipo for each bone.

Figure 3-14. Pinned Action IpoWindow



You can “pin” an ipo or action (lock it to the current window) by pressing the pin icon in the header of the window. The contents of the window will stay there, even when the object is deselected, or another object is selected. Note that the color of the ipo block menu will change, along with the background color of the ipo window. These serve as reminders that the window is not necessarily displaying the ipo of the currently selected object.

Browsing while pinned

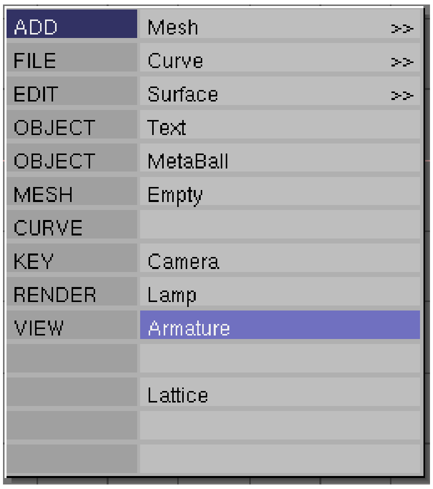
The browse menu is still available while a window is pinned. In this case however, changing the current data will not affect the current object; it merely changes which data is displayed.

3.5.2. Armature Object

Creating

A single armature will contain many bones. Consider an armature to be like a skeleton for a living creature. The arms, legs, spine and head are all part of the same skeleton object.

Figure 3-15. Adding an Armature



To create a new armature, select "ADD->Armature" from the toolbox. A new bone will appear with its root at the location of the 3d cursor. As you move the mouse, the bone will resize accordingly. **LMB** will finalize the bone and start a new one that is the child of the previous one. In this way you can make a complete chain. Pressing **ESC** will cancel the addition of the bone.

Adding Bones

You can add another bone to an armature while it is in edit mode by selecting "ADD->Armature" from the toolbox again. This will start the bone-adding mode again, and the new bones you create will be a part of the current armature.

Extruding Bones

You can also extrude bones from existing bones by selecting a bone joint and pressing **EKEY**. The newly created bone will be a child of the bone it is extruded from.

Editing

While in edit mode, you can perform the following operations to the bones in an armature.

Adjusting

Select one or more bone joints and use any of the standard transformation operations to adjust the position or orientation of any bones in the armature. Note that IK chains cannot have any gaps between their bones and as such moving the end point of a bone will move the start point of its child.

You can select an entire IK chain at once by moving the mouse cursor over a joint in the chain and pressing **LKEY**. You can also use the boundary select tool (**BKEY**).

Deleting

You can delete one or more bones by selecting its start and end points. When you do this you will notice the bone itself will be drawn in a highlighted color. Pressing **XKEY** will remove the highlighted bones. Note that selecting a single point is insufficient to delete a bone.

Point Snapping

It is possible to snap bone joints to the grid or to the cursor by using the snap menu accessible with **SHIFT+S**.

Numeric Mode

For more precise editing, pressing **NKEY** will bring up the numeric entry box. Here you can adjust the position of the start and end points as well as the bone's roll around its own axis.

An easy way to automatically orient the z-axis handles of all selected bones (necessary for proper use of the pose-flipped option) is to press **CTRL+N**. Remember to do this before starting to create any animation for the armature.

Undo

While in edit mode, you can cancel the changes you have made in the current editing session by pressing **UKEY**. The armature will revert to the state it was in before editing began.

Joining

It is possible to join two armatures together into a single object. To do this, ensure you are in object mode, select both armatures and press **CTRL+J**.

Renaming

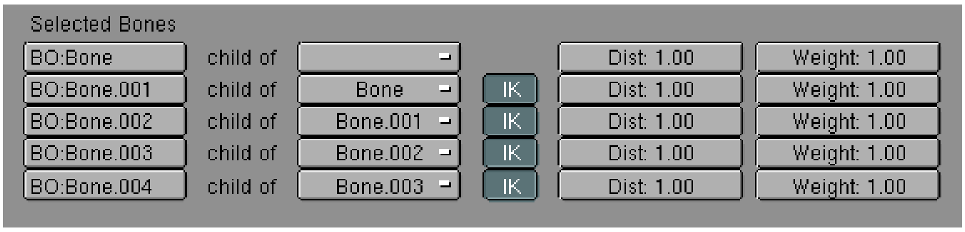
Assigning meaningful names the bones in your armatures is important for several reasons. Firstly it will make your life easier when editing actions in the action window. Secondly, the bone names are used to associate action channels with bones when you are attempting to re-use actions, and thirdly the names are used when taking advantage of the automatic pose-flipping feature.

Note that bone names need only be unique within a given armature. You can have several bone called "Head" so long as they are all in different armatures.

Basic Naming

To change the names of one or more bones, select the bones in edit mode and switch to the edit buttons with **F9**. A list of all the selected bones should appear.

Figure 3-16. EditButtons for an Armature



Selected Bones

BO:Bone	child of			Dist: 1.00	Weight: 1.00
BO:Bone.001	child of	Bone	IK	Dist: 1.00	Weight: 1.00
BO:Bone.002	child of	Bone.001	IK	Dist: 1.00	Weight: 1.00
BO:Bone.003	child of	Bone.002	IK	Dist: 1.00	Weight: 1.00
BO:Bone.004	child of	Bone.003	IK	Dist: 1.00	Weight: 1.00

Change a bone’s name by **SHIFT-LMB** in the bone’s name box and typing a new name.

It is easier to name the bones by either only editing one bone at a time, or by making sure the “Draw-Names” option is enabled in the EditButtons **F9** (see).

Pose Flipping Conventions

Character armatures are typically axially symmetrical. This means that many elements are found in pairs, one on the left and one on the right. If you name them correctly, Blender can flip a given pose around the axis of symmetry, making animation of walk-cycles much easier.

For every bone that is paired, suffix the names for the left and right with either “.L” and “.R” or “.Left” and “.Right”. Bones that lie along the axis of symmetry or that have no twin need no suffix. Note that the part of the name preceding the suffix should be identical for both sides. So if there are two hands, they should be named “Hand.R” and “Hand.L”.

Basic Parenting

To change parenting relationships within the armature, select the bone that should be the CHILD and switch to the edit buttons window. Next to the bone there should be a menu button labeled “Child Of”. To make the bone become the child of another bone, pick the appropriate parent from the list. Note that this is much easier if the bones have been correctly named. To dissolve a parenting relationship, choose the first (blank) entry in the list.

Note that the parenting menu only contains the names of valid parents. Bones that cannot be parents (such as children of the current bone) will not be displayed.

IK Relationship

The IK toggle next to each bone with a parent is used to determine if the IK solver should propagate its effects across this joint. If the IK button is active, the child’s start point will be moved to math its parent’s end point. This is to satisfy the requirement that there are no gaps in an IK chain. Deactivating the IK button will not restore the child’s start point to its previous location, but moving the point will no longer affect the parent’s end point.

Setting Local Axes

To get the best results while animating, it is necessary to ensure that the local axes of each bone are consistent throughout the armature. This should be done before any animation takes place.

Clearing Transforms

It is necessary that the when the armature object is in its untransformed orientation in object mode, that the front of the armature is visible in the front view, the left side is visible in the left view and so on. You can ensure this by orienting the armature so that the appropriate views are aligned and pressing **CTRL+A** to apply size and rotation. Again, this should be done before any animation takes place.

Adjusting Roll Handles

The orientation of the bones’ roll handles is important to getting good results from the animation system.

You can adjust the roll angle of a bone by selecting it and pressing **NKEY**. The roll angle is the item at the bottom. The exact number that must be entered here depends on the orientation of the bone.

The Z-axis of each bone should point in a consistent direction for paired bones. A good solution is to have the Z-axes point upwards (or forwards, when the bone is vertically oriented).

This task is much easier if the “Draw Axes” option is enabled in the edit buttons window.

Setting Weights (DEPRECIATED)

The Weight and Dist settings are only used by the automatic skinning which is a depreciated feature.

Object Mode Parenting

When making a child of an armature, several options are presented.

Parent to Bone

In this case, the a popup menu appears allowing you to choose which bone should be the parent of the child(ren) objects.

Parent to Armature

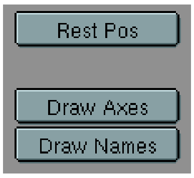
Choosing this option will deform the child(ren) mesh(es) according to their vertex groups. If the child meshes don’t have any vertex groups, they will be subject to automatic skinning. This is very slow, so it is advised to create vertex groups instead.

Parent to Armature Object

Choosing this option will cause the child(ren) to consider the armature to be an Empty for all intents and purposes.

Toggle Buttons for Armatures in the EditButtons F9

Figure 3-17. Draw options for Armatures



Rest Position Button

When this toggle is activated, the armature will be displayed in its rest position. This is useful if it becomes necessary to edit the mesh associated with an armature after some posing or animation has been done. Note that the actions and poses are still there, but they are temporarily disabled while this button is pressed.

Draw Axes Button

When this toggle is activated, the local axes of each bone will be displayed in the 3d view.

Draw Names Button

When this toggle is activated, the names of each bone will be displayed in the 3d view.

3.5.3. Skinning

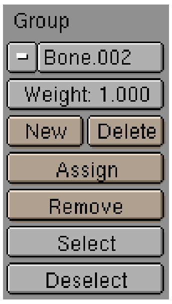
Skinning is a technique for creating smooth mesh deformations with an armature. Essentially the skinning is the relationship between the vertices in a mesh and the bones of an armature, and how the transformations of each bone will affect the position of the mesh vertices.

Automatic (DEPRECIATED)

If a mesh does not have any vertex groups, and it is made the armature-child of an armature, Blender will attempt to calculate deformation information on the fly. This is very slow and is not recommended. It is advisable to create and use vertex groups instead.

Vertex Weights

Figure 3-18. Vertex Groups



Vertex groups are necessary to define which bones deform which vertices. A vertex can be a member of several groups, in which case its deformation will be a weighted average of the deformations of the bones it is assigned to. In this way it is possible to create smooth joints.

Creating

To add a new vertex group to a mesh, you must be in edit mode. Create a new vertex group by clicking on the “New” button in the mesh’s edit buttons.

A vertex group can subsequently be deleted by clicking on the “Delete” button.

Change the active group by choosing one from the pull-down group menu.

Naming

Vertex groups must have the same names as the bones that will manipulate them. Both spelling and capitalization matter. Rename a vertex group by **SHIFT-LMB** on the name button and typing a new name. Note that vertex group names must be unique within a given mesh.

Assigning

Vertices can be assigned to the active group by selecting them and clicking the “Assign” button. Depending on the setting of the “Weight” button, the vertices will receive more or less influence from the bone. This weighting is only important for vertices that are members of more than one bone. The weight setting is not an absolute value; rather it is a relative one. For each vertex, the system calculates the sum of the weights of all of the bones that affect the vertex. The transformations of each bone are then divided by this amount meaning that each vertex always receives exactly 100% deformation.

Assigning 0 weight to a vertex will effectively remove it from the active group.

Removing

Remove vertices from the current group by selecting them and clicking the “Remove” button.

Selection Tools

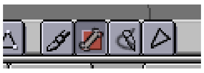
Pressing the “Select” button will add the vertices assigned to the current group to the selection set. Pressing the “Deselect” button will remove the vertices assigned to the current group from the selection set.

3.5.4. Weight Painting

Weight painting is an alternate technique for assigning vertices to vertex groups. The user can “paint” weights onto the model and see the results in real-time. This makes smooth joints easier to achieve.

Activating

To activate weight-painting mode, select a mesh with vertex groups and click on the weight paint icon.



The active mesh will be displayed in weight-color mode. In this mode dark blue represents areas with no weight from the current group and red represent areas with full weight.

Only one group can be visualized at a time. Changing the active vertex group in the edit buttons will change the weight painting display.

Painting

Weights are painted onto the mesh using techniques similar to those used for vertex painting, with a few exceptions.

The “color” is the weight value specified in the mesh’s edit-buttons. The “opacity” slider in the vertex paint buttons is used to modulate the weight.

“Erasing Weight”

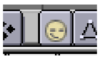
To erase weight from vertices, set the weight to “0” and start painting.

3.5.5. Posemode

To manipulate the bones in an armature, you must enter pose mode. In pose mode you can only select and manipulate the bones of the active armature. Unlike edit mode, you cannot add or delete bones in pose mode.

Entering

Enter pose mode by selecting an armature and pressing **CTRL+TAB**. Alternatively you can activate pose mode by selecting an armature and clicking on the pose mode icon in the header of the 3d window. You can leave pose mode by the same method, or by entering edit mode.



Editing

In pose mode, you can manipulate the bones in the armature by selecting them with **RMB** and using the standard transformation keys: **RKEY**, **SKEY** and **GKEY**. Note that you cannot “grab” (translate) bones that are IK children of another bone.

Press **IKEY** to insert keyframes for selected bones.

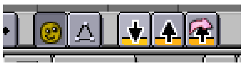
Clearing a pose

If you want to clear the posing for one or more bones, select the bones and press **ALT+R** to clear rotations, **ALT+S** to clear scaling and **ALT+G** to clear translations. Issuing these three commands will all bones selected will return the armature to its rest position.

Copy/Paste/Flipped

It is frequently convenient to copy poses from one armature to another, or from one action to a different point in the same action. This is where the pose copying tools come into play.

or best results, be sure to select all bones in editmode and press **CTRL+N** to auto-orient the bone handles before starting any animation.



To copy a pose, select one or more bones in pose mode, and click on the “Copy” button in the 3d window. The transformations of the selected bones are stored in the copy buffer until needed or until another copy operation is performed.

To paste a pose, simply click the “Paste” button. If “KeyAC” is active, keyframes will be inserted automatically.

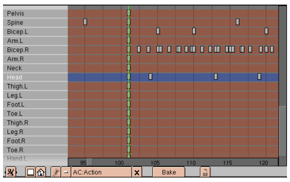
To paste a mirrored version of the pose (if the character was leaning left in the copied pose, the mirrored pose would have the character leaning right), click on the “Paste Flipped” button. Note that if the armature was not set up correctly, the paste flipped technique may not work as expected.

3.5.6. Action Window

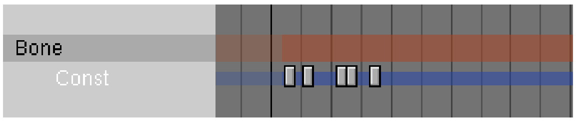
An action is made of one or more action channels. Each channel corresponds to one of the bones in the armature, and each channel has an Action Ipo associated with it. The action window provides a means to visualize and edit all of the Ipos associated with the action.

You can activate the action window with **SHIFT+F12**.

Figure 3-19. ActionWindow



For every key set in a given action ipo, a marker will be displayed at the appropriate frame in the action window. This is similar to the “Key” mode in the ipo window.For action channels with constraint ipos, the will be one or more additional constraint channels beneath each action channel. These channels can be selected independently of their owner channels.



Moving Action Keys

A block of action keys can be selected by either **RMB** on them or by using the boundary select tool (**BKEY**). Selected keys are highlighted in yellow. Once selected, the keys can be moved by pressing **GKEY>** and moving the mouse. Holding **CTRL** will lock the movement to whole-frame intervals. **LMB** will finalize the new location of the keys.

Scaling Action Keys

A block of action keys can be scaled horizontally (effectively speeding-up or slowing-down the action) by selecting number of keys and pressing **SKEY**. Moving the mouse horizontally will scale the block. **LMB** will finalize the operation.

Deleting Action Keys

Delete one or more selected action keys by pressing **XKEY** when the mouse cursor is over the keyframe area of the action window.

Duplicating Action Keys

A block of action keys can be duplicated and moved within the same action by selecting the desired keys and pressing **SHIFT+D**. This will immediately enter grab mode so that the new block of keys can be moved. Subsequently **LMB** will finalize the location of the new keys.

Deleting Channels

Delete one or more entire action or constraint channels (and all associated keys) by selecting the channels in the left-most portion of the action window (the selected channels will be highlighted in blue). With the mouse still over the left-hand portion of the window, press **XKEY** and confirm the deletion. Note that there is no undo so perform this operation with care. Also note that deleting an action channel that contains constraint channels will delete those constraint channels as well.

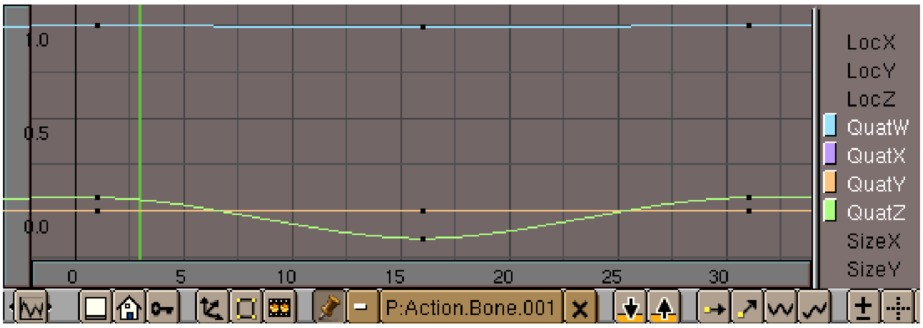
Baking Actions

If you have an animation that involves constraints and you would like to use it in the game engine (which does not evaluate constraints), you can bake the action by pressing the BAKE button in the Action Window headerbar. This will create a new action in which every frame is a keyframe. This action can be played in the game engine and should display correctly with all constraints removed. For best results, make sure that all constraint targets are located within the same armature.

Action IPO

The action ipo is a special ipo type that is only applicable to bones. Instead of using Euler angles to encode rotation, action ipos use quaternions, which provide better interpolation between poses.

Figure 3-20. ActionIpo



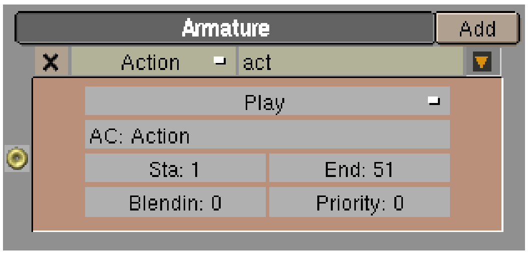
Quaternions

Instead of using a three-component Euler angle, quaternions use a four-component vector. It is generally difficult to describe the relationships of these quaternion channels to the resulting orientation, but it is often not necessary. It is best to generate quaternion keyframes by manipulating the bones directly, only editing the specific curves to adjust lead-in and lead-out transitions.

3.5.7. Action Actuator

The action actuator provides an interface for controlling action playback in the game engine. Action actuators can only be created on armature objects.

Figure 3-21. Action Actuator



Armature

Add

Action

act

Play

AC: Action

Sta: 1

End: 51

Blendin: 0

Priority: 0

Play Modes

Play

Once triggered, the action will play all the way to the end, regardless of other signals is receives.

Flipper

When it receives a positive signal, the action will play to the end. When it no longer receives a positive signal it will play from its current frame back to the start.

Loop Stop

Once triggered, the action will loop so long as it does not receive a negative signal. When it does receive a negative signal it will stop immediately.

Loop End

Once triggered, the action will loop so long as it does not receive a negative signal. When it does receive a negative signal it will stop only once it has reached the end of the loop.

Property

The action will display the frame specified in the property field. Will only update when it receives a positive pulse.

Blending

By editing the “Blendin” field you can request that Blender generates smooth transitions between actions. Blender will create a transition that lasts as many frames as the number specified in the Blendin field.

Priority

In situations where two action actuators are active on the same frame and they specify conflicting poses, the priority field can be used to resolve the conflict. The action with the lowest numbered priority will override actions with higher numbers. So priority “0” actions will override all others. This field is only important when two actions overlap.

Overlapping Actions

It is now possible to have two non-conflicting action actuators play simultaneously for the same object. For example, one action could specify the basic movements of the body, while a second action could be used to drive facial animation. To make this work correctly, you should ensure that the two actions do not have any action channels in common. In the facial animation example, the body movement action should not contain channels for the eyes and mouth. The facial animation action should not contain channels for the arms and legs, etc.

3.5.8. Python

The following methods are available when scripting the action actuator from python.

getAction()

Returns a string containing the name of action currently associated with this actuator.

getBlendin()

Returns a floating-point number indicating the number of blending frames currently specified for this actuator.

getEnd()

Returns a floating-point number specifying the last frame of the action.

getFrame()

Returns a floating-point number indicating the current frame of playback.

getPriority()

Returns an integer specifying the current priority of this actuator.

getProperty()

Returns a string indicating the name of the property to be used for “Property-Driven Playback”.

getStart()

Returns a floating-point number specifying the first frame of the action.

setAction(action, reset)

Expects a string action specifying the name of the action to be associated with this actuator. If the action does not exist in the file, the state of the actuator is not changed.

If the optional parameter reset is set to 1, this method will reset the blending timer to 0. If the reset is set to 0 this method leaves the blending timer alone. If reset is not specified, the blending timer will be automatically reset. Calling this method does not however, change the start and end frames of the action. These may need to be set using setStart and setEnd

setBlendin(blendin)

Expects a positive floating-point number blendin specifying the number of transition frames to generate when switching to this action.

setBlendtime(blendtime)

Expects a floating-point number blendtime in the range between 0.0 and 1.0. This can be used to directly manipulate the internal timer that is used when generating transitions. Setting a blendtime of 0.0 means that the result pose will be 100% based on the last known frame of animation. Setting a value of 1.0 means that the pose will be 100% based on the new action.

setChannel(channelname, matrix)

Accepts a string channelname specifying the name of a valid action channel or bone name, and a 4x4 matrix (a list of four lists of four floats each) specifying an overriding transformation matrix for that bone. Note that the transformations are in local bone space (i.e. the matrix is an offset from the bone's rest position).

This function will override the data contained in the action (if any) for one frame only. On the subsequent frame, the action will revert to its normal course, unless the channel name passed to setChannel is not specified in the action. If you wish to override the action for more than one frame, this method must be called on each frame.

Note that the override specified in this method will take priority over all other actuators.

setEnd(end)

Accepts a floating-point number end, which specifies what the last frame of the action should be.

setFrame(frame)

Passing a floating-point number frame allows the script to directly manipulate the actuator's current frame. This is low-level functionality for advanced use only. The preferred method is to use Property-Driven Playback mode.

setPriority(priority)

Passing an integer priority allows the script to set the priority for this actuator. Actuators with lower priority values will override actuators with higher numbers.

setProperty(propertyname)

This method accepts a string propertyname and uses it to specify the property used for Property-Driven-Playback. Note that if the actuator is not set to use Property-Playback, setting this value will have no effect.

setStart(start)

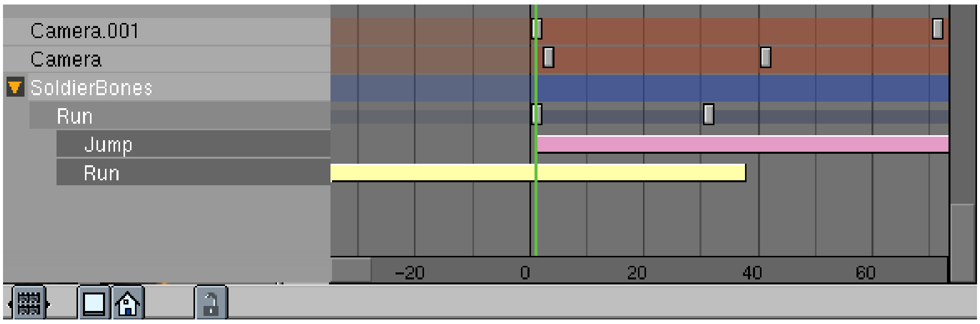
To specify the starting frame of the action, pass a floating-point number start to this method.

3.5.9. NLAWindow (Non Linear Animation)

This window gives an overview of all of the animation in your scene. From here you can edit the timing of ALL ipos, as if they were in the action window. Much of the editing functionality is the same as the Action window.

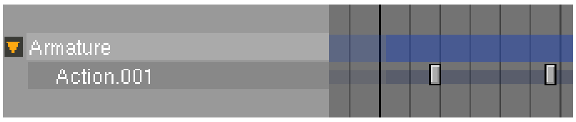
You can display the NLAWindow with **CTRL+SHIFT+F12**.

Figure 3-22. NLAWindow



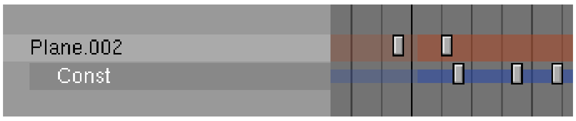
You can also use this window to perform action blending and other Non-Linear Animation tasks. You add and move action strips in a fashion similar to the sequence editor, and generate blending transitions for them.

In the NLA window actions are displayed as a single strip below the object's stip; all of the keyframes of the action (constraint channel keyframes included) are displayed on one line.



To see an expanded view of the action, use the Action window.

Objects will constraint channels will display one or more additional constraint strips below the object strip. The constraint strip can be selected independently of its owner object.



RMB clicking on object names in the NLA window will select the appropriate objects in the 3dWindow. Selected object strips are drawn in blue, while unselected ones are red.

You can remove constraint channels from objects by clicking **RMB** on the constraint channel name and pressing **XKEY**.

Note that only armatures, or objects with ipos will appear in the NLA window.

Working with Action Strips

Action strips can only be added to Armature objects. The object does not necessarily need to have an action associated with it first.

Add an action strip to an object by moving the mouse cursor over the object name in the NLA window and pressing **SHIFT+A** and choosing the appropriate Action to add from the popup menu. Note that you can only have one action strip per line.

You can select, move and delete action strips along with other keyframes in the NLA window.

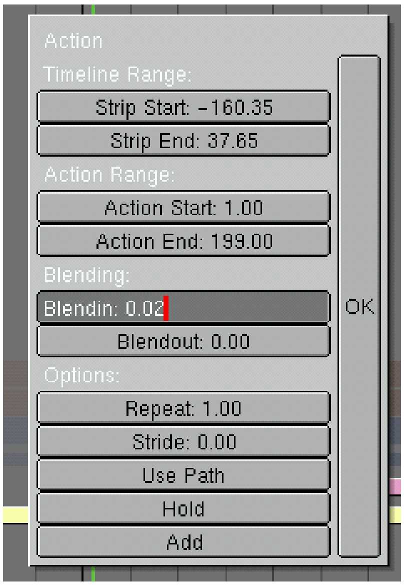
The strips are evaluated top to bottom. Channels specified in strips later in the list override channels specified in earlier strips.

You can still create animation on the armature itself. Channels in the local action on the armature override channels in the strips. Note that once you have created a channel in the local action, it will always override all actions. If you want to create an override for only part of the timeline, you can convert the local action to an action strip by pressing **CKEY** with your mouse over the armature's name in the NLA window. This removes the action from the armature and puts it at the end of the action strip list.

Action Strip Options

Each strip has several options which can be accessed by selecting the strip and pressing **NKEY**. The options available are as follows:

Figure 3-23. Action Strip Options



Strip Start: -160.35

Strip End: 37.65

Action Start: 1.00

Action End: 199.00

Blendin: 0.02

Blendout: 0.00

Repeat: 1.00

Stride: 0.00

StripStart/StripEnd

The first and last fame of the action strip in the timeline

ActionStart/ActionEnd

The range of keys to read from the action. The end may be less than the start which will cause the action to play backwards.

Blendin/Blendout

The number of frames of transition to generate between this action and the one before it in the action strip list.

Repeat

The number of times the action range should repeat. Not compatible with "USE PATH" setting.

Stride

The distance (in Blender units) that the character moves in a single cycle of the action (usually a walk cycle action). This field is only needed if "USE PATH" is specified.

Use Path

If an armature is the child of a path or curve and has a STRIDE value, this button will choose the frame of animation to display based on the object's position along the path. Great for walkcycles.

Hold

If this is enabled, the last frame of the action will be displayed forever, unless it is overridden by another action. Otherwise the armature will revert to its rest position.

Add

Specifies that the transformations in this strip should ADD to any existing animation data, instead of overwriting it.

3.5.10. Constraints

Constraints are filters that are applied to the transformations of bones and objects. These constraints can provide a variety of services including tracking and IK solving.

Constraint Evaluation Rules

Constraints can be applied to objects or bones. In the case of constraints applied to bones, any constraints on the armature OBJECT will be evaluated before the constraints on the bones are considered.

When a specific constraint is evaluated, all of its dependencies will have already been evaluated and will be in their final orientation/positions. Examples of dependencies are the object's parent, its parent's parents (if any) and the hierarchies of any targets specified in the constraint.

Within a given object, constraints are executed from top to bottom. Constraints that occur lower in the list may override the effects of constraints higher in the list. Each constraint receives as input the results of the previous constraint. The input to the first constraint in the list is the output of the ipos associated with the object.

If several constraints of the same type are specified in a contiguous block, the constraint will be evaluated ONCE for the entire block, using an average of all the targets. In this way you can constrain an object to track to the point between two other objects, for example. You can use a NULL constraint to insert a break in a constraint block if you would prefer each constraint to be evaluated individually.

Looping constraints are not allowed. If a loop is detected, all of the constraints involved will be temporarily disabled (and highlighted in red). Once the conflict has been resolved, the constraints will automatically re-activate.

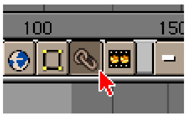
Influence

The influence slider next to each constraint is used to determine how much effect the constraint has on the transformation of the object.

If there is only a single constraint in a block (a block is a series of constraints of the same type which directly follow one another), an influence value of 0.0 means the constraint has no effect on the object. An influence of 1.0 means the constraint has full effect.

If there are several constraints in a block, the influence values are used as ratios. So in this case if there are two constraints, A and B, each with an influence of 0.1, the resulting target will be in the center of the two target objects (a ratio of 0.1:0.1 or 1:1 or 50% for each target).

Influence can be controlled with an ipo. To add a constraint ipo for a constraint, open an ipo window and change its type to constraint by clicking on the appropriate icon.



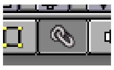
Next click on the Edit Ipo button next to the constraint you wish to work with. If there is no constraint ipo associated with the constraint yet, one will be created. Otherwise the previously assigned ipo will be displayed. At the moment, keyframes for constraint ipos can only be created and edited in the ipo window, by selecting the INF channel and **CTRL+LEFTMOUSE** in the ipo space.

When blending actions with constraint ipos, note that only the ipos on the armature's local action ipos are considered. Constraint ipos on the actions in the motion strips are ignored.

In the case of armatures, the constraints ipos are stored in the current Action. This means that changing the action will change the constraint ipos as well.

Creating Constraints

To add a constraint to an object, ensure you are in object mode and that the object is selected. Switch to the constraint buttons window (the icon looks like a pair of chain links) and click on the "Add" button.



A new constraint will appear. It can be deleted by clicking on the “X” icon next to it. A constraint can be collapsed by clicking on its orange triangle icon. When collapsed, a constraint can be moved up or down in the constraint list by clicking on it at choosing “Move Up” or “Move Down” from the popup menu.

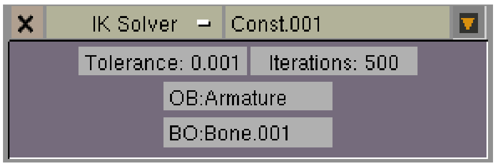
For most constraints, a target must be specified in the appropriate field. In this field you must type in the name of the desired target object. If the desired target is a bone, first type in the name of the bone’s armature. Another box will appear allowing you to specify the name of the bone.

Adding Constraints to Bones

To add a constraint to a bone, you must be in pose mode and have the bone selected.

Constraint Types

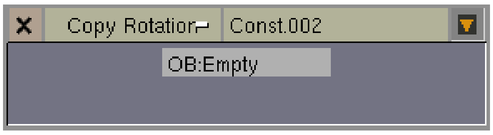
IK Solver



IK Solver
Const.001
Tolerance: 0.001
Iterations: 500
OB:Armature
BO:Bone.001

To simplify animation of multi-segmented limbs (such as arms and legs) you can add an IK solver constraint. IK constraints can only be added to bones. Once a target is specified, the solver will attempt to move the ROOT of the constraint-owning bone to the target, by re-orienting the bone’s parents (but it will not move the root of the chain). If a solution is not possible, the solver will attempt to get as close as possible. Note that this constraint will override the orientations on any of the IK bone’s parents.

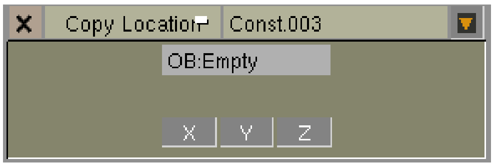
Copy Rotation



Copy Rotation
Const.002
OB:Empty

This constraint copies the global transformation of the target and applies it to the constraint owner.

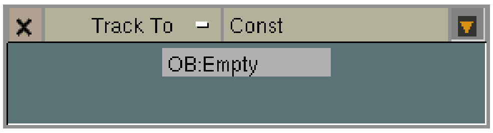
Copy Location



Copy Location
Const.003
OB:Empty

The constraint copies one or more axes of location from the target to the constraint owner.

Track To

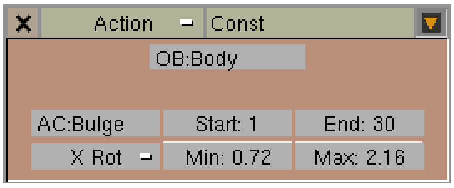


OB:Empty

This constraint causes the constraint owner to point its Y-axis towards the target. The Z-axis will be oriented according to the setting in the anim-buttons window. By default, the Z-axis will be rolled to point upwards.

Action

An action constraint can be used to apply an action channel from a different action to a bone, based on the rotation of another bone or object. The typical way to use this is to make a muscle bone bulge as a joint is rotated. This constraint should be applied to the bone that will actually do the bulging; the target should point to the joint that is being rotated.



OB:Body

AC:Bulge
Start: 1
End: 30
X Rot
Min: 0.72
Max: 2.16

The AC field contains the name of the action that contains the flexing animation. The only channel that is required in this action is the one that contains the bulge animation for the bone that owns this constraint

The Start and End fields specify the range of motion from the action.

The Min and Max fields specify the range of rotation from the target bone. The action between the start and end fields is mapped to this rotation (so if the bone rotation is at the Min point, the pose specified at Start will be applied to the bone). Note that the Min field may be higher than the Max.

The pulldown menu specifies which component of the rotation to focus on.

Null

This is a constraint that does nothing at all; it doesn't affect the object's transformation directly. The purpose of a null constraint is to use it as a separator. Remember that if several constraints of the same type follow one another, the actual constraint operation is only evaluated once using a target that is an average of all of the constraints' targets. By inserting a null constraint between two similarly-typed constraints, you can force the constraint evaluator to consider each constraint individually. This is normally only interesting if one or more of the constraints involved have an Influence value of less than 1.0.

Chapter 4. About locking and security

4.1. Practical usage of locking

For artists and companies who wish to publish Blender content, locking will prevent other Blender users from editing files or retrieving parts of it. It will emphasize the fact that the content was only meant for viewing, and that only illegal acts will allow someone to retrieve it.

Blender’s locking is not intended to send secret proprietary information. For that other technologies are available, which NaN does not offer.

For a clear understanding of the system, the next sections describe the background of Blenders locking system.

4.2. The Key file

The Blender Publisher Key consists of a public and private key-pair. The public part of a Key is always written in signed or locked files. The private part is never distributed. For this reason, to prevent others to unlock your Blender content, the Key file should never be distributed or made public.

4.3. How secure is Locking?

Using the key-pair, Blender Publisher can encrypt the data in a Blender file. This then can only be decrypted by the original owner, or by the 3D Plug-in and stand-alone player (which do not have editing or file-saving included). Encryption and decryption is done with a system based on public and private key pair technology. It is a proven technology which is widely accepted for secure internet transport.

Since it does not make sense to lock Blender files that cannot be opened again, the fact that a Blender Plug-in or stand-alone Player can read a locked file is an obvious security hole. So instead of hacking the encryption system, which is very difficult, a hacker could retrieve pieces of information from memory instead. This is the common hacking approach, and will usually only result in fragmented data recovery or corrupt files.

All licenses included with NaN software do not allow reverse engineering of the code. It’s simply illegal, and will refine distribution of Blender file crackers to the illegal circuit. NaN will do anything that is in its legal and technical scope to prevent or to solve abuse of the Publisher locking system. Our Publisher support system will keep track of that.

However, NaN can not take any liability or give guarantees that information contained in a locked Blender file can not be extracted in any way.

4.4. How secure is signing?

The signature system is meant to prove ownership of a specific file. The signature becomes invalid or disappears the moment anyone changes that file. The encryption system as used by Publisher will make it very hard for others to sign content with your identification, or alter anything in the file while keeping the signature valid. As with locking, signing is based on an encryption system which is widely accepted for secure internet transport.

4.5. Python security

Blenders scripting language - Python (http://www.python.org) - is available in the 3D Plug-in, stand-alone player and run-times as well. Since Python is an extremely powerful language, we had to make sure that Blender 3D plug-in content using Python cannot be used to perform any actions that can cause loss of files or other harm to your system.

This could be:

- Trojan Horses: a program which sets up a network connection in the background, allowing access to the clients machine or sending back relevant data to the hacker.
- Viruses: programs that may alter system configuration files or executables and reproduce themselves that way

The Python implementation in the Blender 3D plug-in prevents the above by:

- Disabling file open functionality, meaning, no writing or reading of files is possible
- Disabling access to external modules, e.g. prohibiting network access, manipulation of the file system, memory, or any other functionality that is not related to the played back content.
- Full Python functionality is available in the stand-alone player.

Chapter 5. Licenses

With Blender Publisher 2.25 four license types are available:

Table 5-1. License types of Blender Publisher

	Individual	**Company**	**Educational**	**Unlimited**
# of Key licenses	1	15	Unlimited	Unlimited
Key bound to:	The person. Only this person can use it at any workstation	The legal entinity, one for each key installed	A legal educational entity for all workstations owned	The legal entinity
Distribution	1.000 copies	15.000 copies	Unlimited for non-commercial usage	Unlimited

The individual license is meant for freelance media professionals or one-man companies. The License (and Key) will be put on the name of this person only. The owner can personally use the Publisher Key on any computer he/she owns, also when these computers require different OS versions of Blender Publisher. Individual licensees can distribute media containing Blender players up to 1000 copies.

The company license works like a site license. This License is put on the name of the legal entity. Up to 15 computers or OS's within the company are allowed to have the Publisher Key installed, independent of the amount of people who have access to this. Company licensees can distribute media containing Blender players up to 15000 copies

The unlimited license is meant for large publishers or media companies who don't want to bother with a limited amount of Publisher installations or who want to distribute mass media with Blender players.

The right to distribute only applies to media products which include either a Blender Plug-in or Blender stand-alone Player or a Blender run-time. Copying and distributing Blender data files (`.blend` or `.blip`) are not limited by any NaN license. A product can be any collection of digital files, distributed via CDROM or any other digital media, offered as download or bundled in with any piece of hardware. Since distribution via the internet as download is hard to limit to a certain maximum, the duplication limits do no apply for non-commercial downloads, where anyone has free access to the download.

For a full description of the license, please refer to the text file that is included in the download.

Frequently asked questions and answers (FAQs)

Q: I am a freelance artist, and I was asked to make an interactive 3D presentation for a large publisher who aims to make 50k copies. Who is supposed to buy the Publisher license?
A: You should get an Individual license yourself, and ask the publisher to buy the Unlimited license; which grants him to make all the copies he wants.

Q: Can I transfer the right to re-distribute Blender runtimes to someone else? Or in other words, when I create the content, can someone else then make the copies?

A: No, this right is not transferable. This is pretty common for producers who work with copyright issues on creative content. For that reason most media companies deliver media duplicates to the clients themselves, providing an important addition to their revenues. When you - instead - sell the right to duplicate your Blender runtimes, the new owner has to buy a Blender Publisher them self.

Chapter 6. Usage of Blender 3D Plug-in

6.1. Introduction

The Blender 3D Plug-in allows you to publish interactive 3D Blender productions for web browsers on different computer platforms. On most platforms content will be handled by the Blender 3D Plug-in for Netscape. Besides being a plug-in for Netscape itself, the Netscape plug-in can also be used in Mozilla.

For Internet Explorer on Windows, we have created a so-called ActiveX control. The Blender 3D Plug-in ActiveX control can also be used to publish content in other applications that support OLE/COM. Among those are Word, PowerPoint and Macromedia Director.

6.2. Functionality

The Blender 3D Plug-in is able to display two kinds of Blender files: regular Blender files and Publisher Blender files. Regular Blender files are created with the free Blender Creator and Publisher Blender files are created with Blender Publisher viable to those that own a Publisher license. When the plug-in displays a regular Blender file, the Blender logo's are displayed on top of the content. Owners of a Blender Publisher license can generate Publisher Blender files. This new file format supports compression for faster download, signing to signify file ownership and locking so that your content can not be altered. Another important advantage of Publisher files is that the plug-ins will not display the Blender logo's. In addition, a Publisher license enables you to create custom loading animations that replace the build-in loading animation

Figure 6-1. Blender 3D Plug-in functionality diagramm

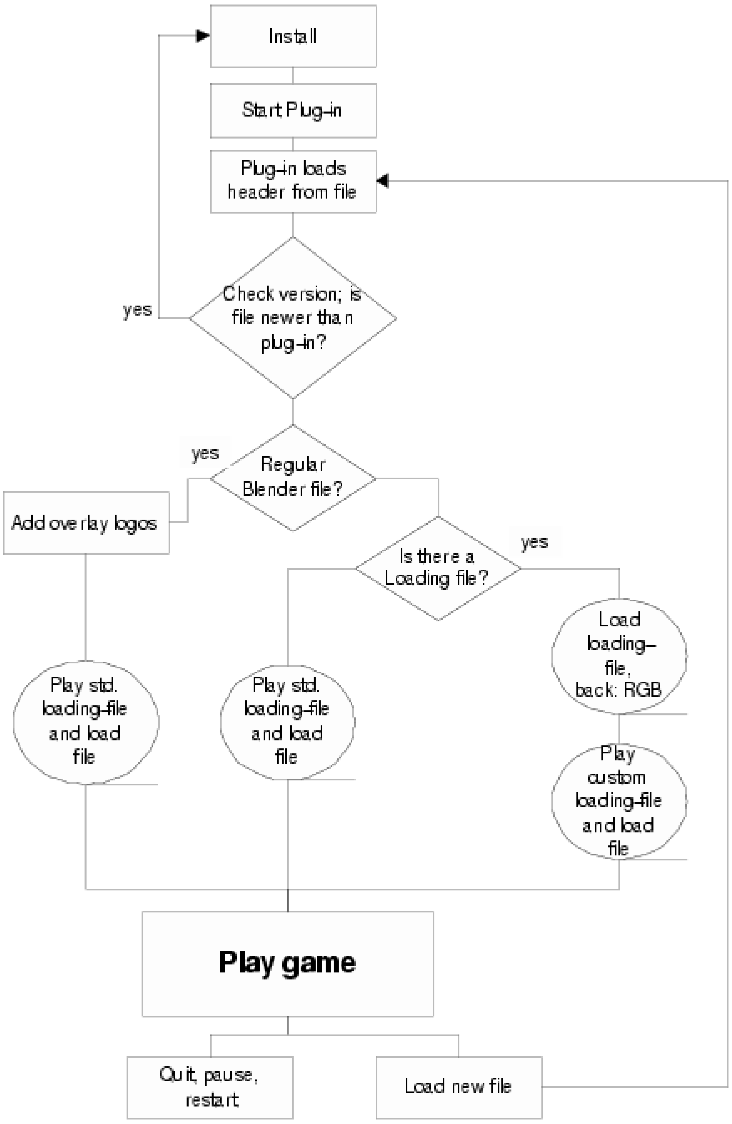

Install
Start Plug-in
Plug-in loads header from file
Check version; is file newer than plug-in?
yes
Regular Blender file?
yes
Add overlay logos
Is there a Loading file?
yes
Load loading-file, back: RGB
Play std. loading-file and load file
Play std. loading-file and load file
Play custom loading-file and load file
Play game
Quit, pause, restart
Load new file

Figure 6-1 gives you an overview of how the plug-ins will process the different file types. When the plug-in is loaded, it determines whether a custom loading animation is requested. If so, it will commence downloading this file while displaying a solid color (if it is not already in the cache on the client system). The color can be specified in the HTML code or else, if missing from HTML, the background color of the HTML page is chosen. At download completion, the plug-in will download the main Blender file to be displayed while displaying the custom loading animation. The main Blender file must be a Publisher Blender file. If not, the plug-in will not play the file downloaded.

If a custom loading animation was not specified, the plug-in will download the main Blender file while displaying the build-in Blender loading animation. After completion, the plug-in displays the file downloaded with or without logo's depending on the file type (regular Blender or Publisher Blender file).

6.3. 3D Plug-in installation

The installation procedure of the Blender 3D Plug-in depends on the type of plug-in and on the operating system. The Active X control can be installed automatically from the HTML page when the HTML code is properly written. For details, read Section 6.4.3 on how to embed the plug-ins in HTML. This installation process also includes automatic updates when a new plug-in becomes available.

Netscape

Downloading and installing the Netscape Blender 3D Plug-in is almost done automatically. If the plug-in is not available for your browser, you will be redirected to the Blender 3D Plug-in download page. There, you will find instructions on how to proceed.

In case there are problems when installing the plug-in, please read the FAQ section in Section 6.7.

6.4. Creating content for the plug-ins

When creating content for the plug-in, there is not much difference with creating and running content inside Blender or as a stand-alone game. There is one difference however. The plug-in might have dimensions that do not correspond to the settings made in the Blender file. This might create a difference in aspect ratios. The plug-in will match the two as good as possible.

The next figures show the situation of a perfect match. In the 3D view of Figure 6-2, the outer dotted rectangle shows the area that is to be displayed. The size and shape of this rectangle is set by changing the value in the SizeX and SizeY buttons in the DisplayButtons (F10) of Blender. The size of the plug-in in Figure 6-3 has been set to the same values.

Figure 6-2. File in 3D view

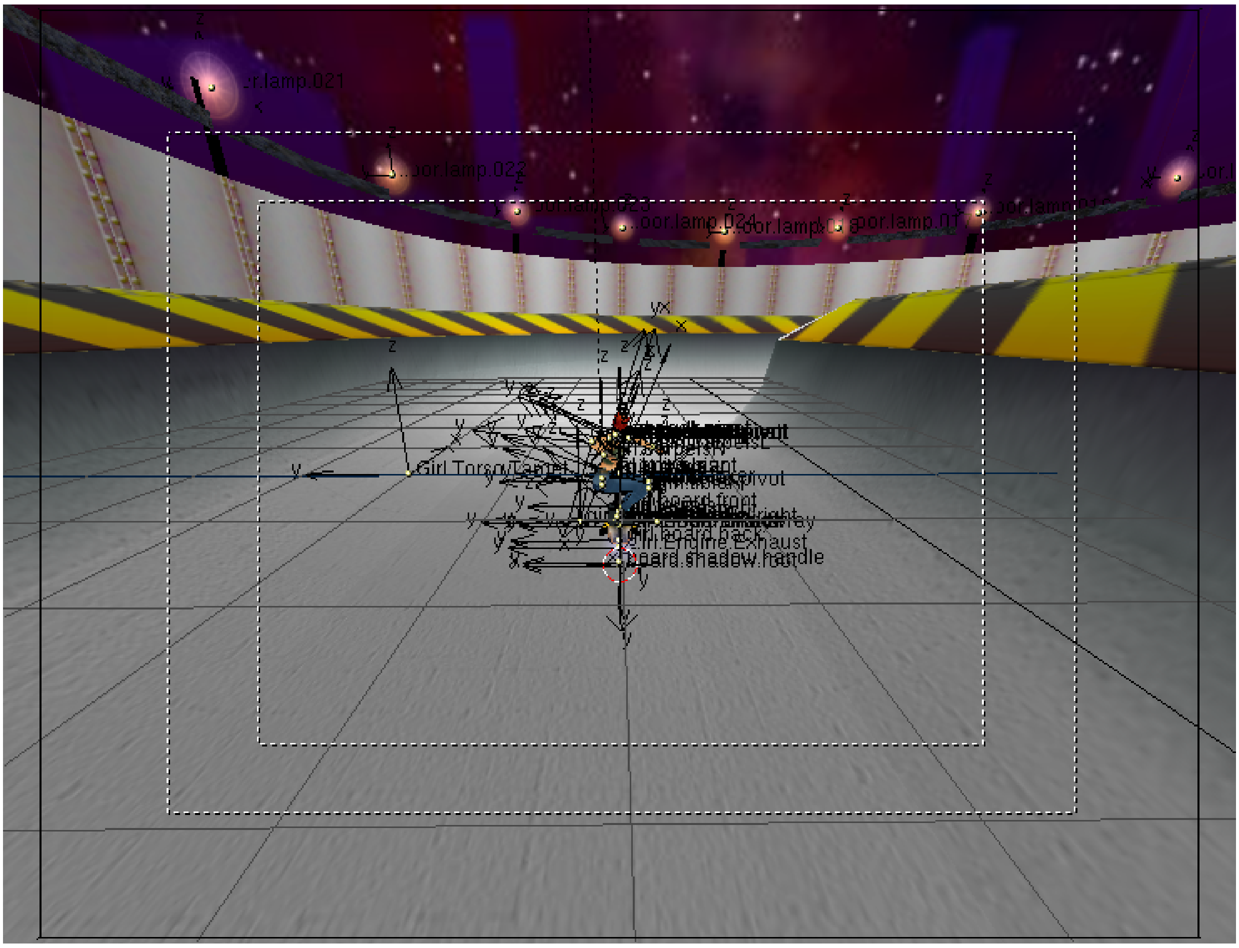


Figure 6-3. Perfect match in the plug-in

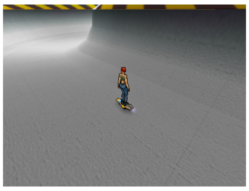
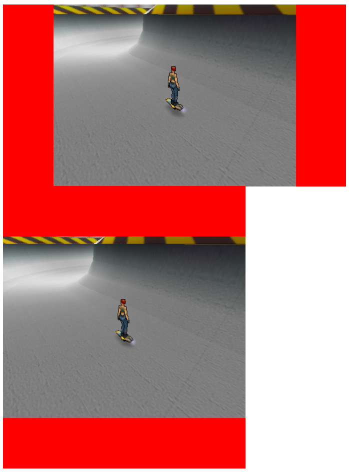


Figure 6-4. Framing of content



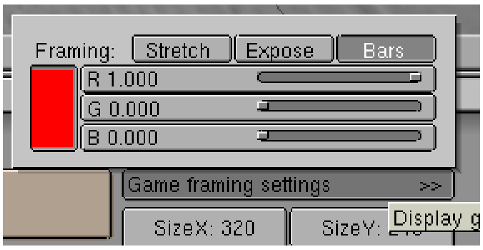
In the images in Figure 6-4, the aspect ratio of the plug-in does not match the Blender file. You'll notice that the plug-in or stand-alone player will try to show the content as big as possible without distortion of the content. The extra areas within the plug-in are drawn in a solid color (red in the figures) that can be either set in the HTML or in the Blender user interface.

How the plug-in or stand-alone player solves the difference in aspect ratio can be controlled in Blender. In the Display Buttons (F10) click on the "Game framing settings" button. You now can select three options: Stretch, Expose or Bars. If you select Bars the extra areas are filled with the color you set with the color sliders.

You can have different settings for different scenes. So you can have a 3D scene with bars and an overlay scene that is stretcht to fit to the output size.

If you select for a single 3D scene the settings as they are shown in Figure 6-5 you'll get results similar to the pictures in Figure 6-4.

Figure 6-5. GUI for Framing in the DisplayButtons (F10)



Framing: Stretch Expose Bars

R 1.000

G 0.000

B 0.000

Game framing settings >>

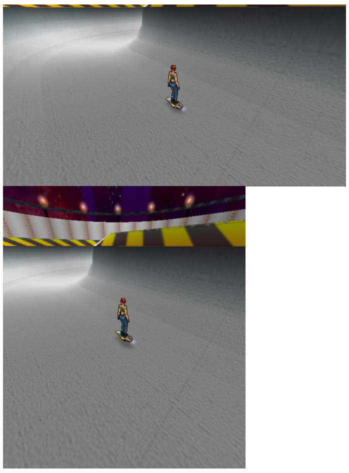
SizeX: 320

SizeY:

Display g

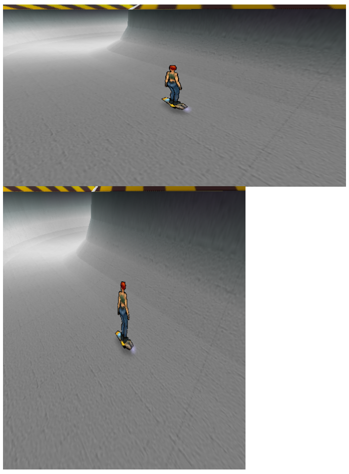
If you select "Expose", the plug-in or stand alone player will simply show more of the 3D environment. This will usually produce the most natural results but if you have enemies coming from the top or from the sides the user might see them pop up.

Figure 6-6. Extended framing



And finally, if you select "Stretch", the plug-in or stand-alone player will simply stretch the image horizontally or vertically to fit. This will distort the image somewhat (See Figure 6-7) but you’ll never see bars or more from the 3D world as defined in Blender.

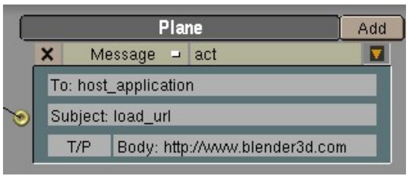
Figure 6-7. Stretched framing



6.4.1. Jumping to another HTML page

It is possible to have the browser load a new URL from within your Blender file. Send a message with the "To:" field set to "host_application" and the "Subject:" "load_url". The messages "Body:" should contain the full URL you want the browser to load.

Figure 6-8. Browsing HTML pages from within the Plug-in



To: host_application

Subject: load_url

T/P

Body: http://www.blender3d.com

6.4.2. Creating a custom loading animation

The file `load.blend` (download at http://www.blender3d.com/plugin/blend/load.blend) is an example of a custom loading animation for the Blender 3D Web Plug-in. The selected object has all of the important logic for showing file loading progress. It has a property called "progress". The value of that property drives the Ipo animation curve of the object, causing its size to increase in the Z direction. This is the most convenient way to use the loading information, because it’s easy to set up and preview an Ipo animation.

The only tricky part is to get the file loading progress information being sent by the plug-in. The plug-in sends game messages with the subject "progress" as the file loads. Each message has a body which is a floating point value between 0.0 and 1.0 sent as a text string. The value 0.0 means that none of the file is loaded yet. 1.0 means that the file is completely loaded. This information is extracted by the Python script "progress.py", which gets all "progress" messages from the message sensor (in case more than one message is received within a single cycle of the game logic). It evaluates the body of the last message, converting it back to a numerical value and assigns the value to the "progress" property of the object.

The camera has some logic attached which causes it to send artificial progress messages for testing the animation. This logic should be deleted from the camera before the file is actually put into use.

Tips:

1. The purpose of a loading animation is to occupy the viewer’s attention while a larger file is loading. It should be as small as possible so that it loads very quickly. Textures, especially *.tga images, increase file size a lot. Use JPEG images or avoid images completely. Save your file using Blender’s file compression tool.
2. Most of the complexity in this example is for showing the download progress of the larger file. Showing the loading progress is very reassuring to the viewer, but not absolutely necessary. You can use any real-time Blender scene as a loading animation.

6.4.3. Embedding Blender 3D Plug-in in web pages

To embed the Blender 3D Plug-in on your web pages, you will need to add some HTML code to your web pages. Also, you will want to add a link to the Blender 3D Plug-in page (http://www.blender3d.com/plugin/) where users can install the plug-in if it is not already installed on their system. Click-able images that you can use to forward users to the download page are available at the Blender 3D Plug-in page.

The current version (2.25) of the Active X control (the Internet Explorer plug-in) is now able to support multiple plug-ins on a one HTML page. The Netscape version does not support this however. Therefore, you are still advised not to put two plug-ins on a HTML page. This will be fixed in one of our forthcoming releases.

Insert the following HTML tag into your web page and change the parameters to suit your content:

```
<p>
<object
  classid="clsid:5DB05CB8-7751-469D-A1DD-45C8C201C013"
  id=Blender3DPlugin
  width = 640
  height = 480
  codebase="http://www.blender.nl/plugin/Blender3DPlugin.cab#Version=2,25,4,0">

<param name="blenderURL" value="http://www.yoursite.com/yourproduction.blend">
<param name="loadingURL" value="http://www.yoursite.com/yourloadinganimation.blend">
<param name="ForeColor" value=65280>
<param name="BackColor" value=255>
<param name="useFileBackColor" value=1>
<param name="frameRate" value=20>

<EMBED
  type="application/x-blender-plugin"
  PLUGINSPAGE="http://plugin.blender.nl"
  name="NPBlender"
  WIDTH=640
  HEIGHT=480
  SRC="http://www.yoursite.com/yourproduction.blend"
  loadingURL="http://www.yoursite.com/yourloadinganimation.blend"
  ForeColor=65280
  BackColor=255
  useFileBackColor=1
  frameRate=20>
</EMBED>
</object>
</p>
```

This code works for both the ActiveX control and the Netscape plug-ins.

The part between the `<object>` and the `<EMBED>` tags relates to the ActiveX control.

- The classid is the unique identifier of the Blender 3D Plug-in.

- The id is the name of the plug-in on the page. This can be used to identify the plug-in in Javascript.
- The width and height parameters allow you to set the dimensions of the plug-in on the page. In the example, width and height are given in pixels. You can also specify the dimensions in percentages of the size of the page (e.g. width = "50%").
- The codebase is the URL the ActiveX control will be downloaded from when it is not installed on the system the page is viewed on. The version number after the hash sign (#) should read the minimum version of the ActiveX control needed to view your content. Internet Explorer will compare this version to the version of the ActiveX control installed on the system. If the ActiveX control installed is older, the newer version is downloaded and installed automatically.

The `<object>` tag is followed by a list of parameters:

- blenderURL (required) is the URL of the Blender file to be viewed.
- loadingURL (optional) is the URL of the custom loading animation.
- ForeColor (optional) is the color to be shown while the custom loading animation is downloaded.
- BackColor (optional) is the color used to draw the extra areas when the aspect ratio of the plug-in does not match the aspect ratio set in the Blender file.
- useFileBackColor (optional) read the color to draw the extra areas with from the Blender file. If neither BackColor or useFileBackColor are set the HTML background color is used to draw the extra areas.
- frameRate (optional) is the maximum number of frames per second. When your animation does not need to be viewed at maximum frame rate possible (e.g. web banners), set this value to a meaningful maximum. With lower frame rates the client system will remain more responsive. For other types of content (e.g. games) you probably want to set this value to the maximum of 100 or leave out the parameter in which case the plug-in will use 100 as well.

The Netscape plug-in settings can be found between the `<EMBED>` and `</EMBED>` tags. Most of the parameters are the same as those of the ActiveX control.

Parameters that differ are:

- PLUGINSPAGE (optional) The URL of the web page displayed by the browser when the plug-in is not installed on the client system.
- name (required) The id is the name of the plug-in on the page. This can be used to identify the plug-in in Javascript. It is the equivalent of the id of he Active control.
- SRC (required) is the URL of the Blender file to be viewed. The equivalent of the ActiveX blenderURL parameter.

Color values should be passed to the ActiveX control in a format known as OLE_COLOR. The red, green and blue components of the color are stored in a single value. To determine a BGR value, specify blue, green and red (each of which has a value from 0 - 255) in the following formula: BGR value = (blue * 65536) + (green * 256) + red

Some parameters of the plug-ins can be dynamically accessed (through Javascript for example). This means that you can interact with the plug-in from your HTML code. For instance, you can change the URL of the Blender file from the HTML page with a button by adding the following to your HTML page:

```
<form>
<input type="button"
value="load other production"
onClick="Blender3DPlugin.blenderURL='http://www.yoursite.com/other.blend';">
</form>
```

This feature is currently not available for the Netscape plug-ins. We are planning to add Javascript capabilities in the next release.

Another very powerful option is to send messages to the Blender file running inside the plug-in. This way you can change the behaviour of your production from the HTML. The following form allows you to type in messages in the HTML and send them to the plug-in:

```
<p>
<form onSubmit="Blender3DPlugin.SendMessage(
     the_to.value,
     the_from.value,
     the_subject.value,
     the_body.value); return false;">

<table><tr>

<td>to:</td>
<td><input type="text" name="the_to" value="" size=30></td></tr>

<tr><td>from:</td>
<td><input type="text" name="the_from" value="" size=30></td></tr>

<tr><td>subject:</td>
<td><input type="text" name="the_subject" value="" size=30></td></tr>

<tr><td>body:</td>
<td><input type="text" name="the_body" value="" size=30></td></tr>

<tr><td>&nbsp</td>
<td>
<input type="button" value="send"
     onClick="Blender3DPlugin.SendMessage(the_to.value,
     the_from.value, the_subject.value, the_body.value);">
</td></tr>

</table>
</form>
```

```
31 </p>
```

6.5. Embedding the ActiveX control in other applications

If Blender content is played in Internet Explorer, it is played using the Blender 3D Plug-in Active X Control. Active X is a Microsoft technology that allows Active X controls to run inside a host application. Internet Explorer is one example of such a host but there are many more. Another application that can use Active X controls is the popular PowerPoint application. For detailed information on Active X controls see Microsoft's Active X pages (http://www.microsoft.com/com/tech/activex.asp).

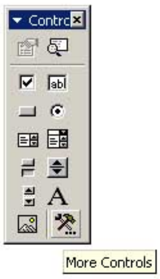
When a control is needed for a certain type of playback, it is automatically downloaded from a secure location, the control installs itself, and then becomes available in Windows. The advantage of Active X controls is that although they are generally downloaded and used in Internet Explorer, once they exist on your system any Windows application that understands Active X can use the controls to extend their applications.

6.5.1. Embedding Blender content in PowerPoint

As example how to embed Blender content into other applications using the Active X control, we use here Microsofts Powerpoint.

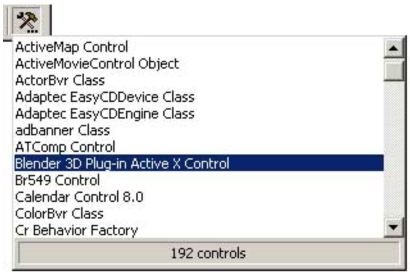
Steps to embed Blender content in PowerPoint 2000:

1. Start PowerPoint 2000
2. Open the Active X Control Toolbox by choosing View > Toolbars > Control Toolbox.
3. The item in the bottom right corner is the "More Controls" button.



Click on the button to have a list of Active X Controls appear that are installed on the machine.

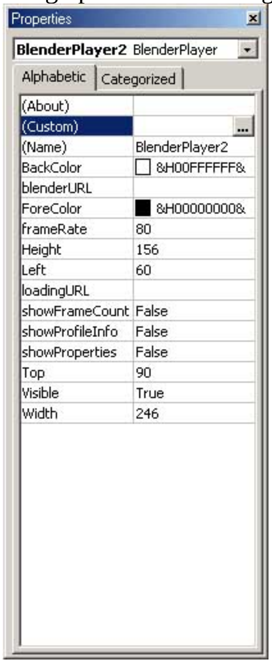
4. Choose "Blender 3D Plug-in Active X Control" from the list:



5. You will notice that the cursor changes to a crosshair. This indicates that you can draw a rectangle to define the location and dimensions of the control in your presentation.

6. Press the left mouse button at the desired location and drag the mouse (while keeping the button pressed) to define its size.

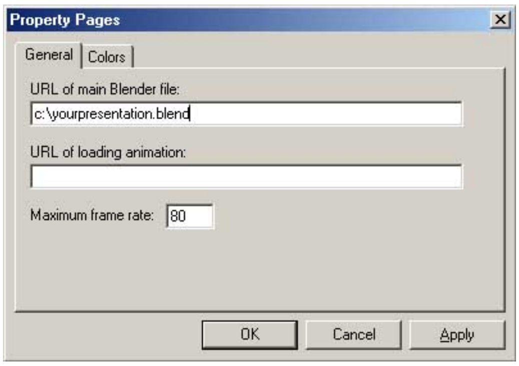
7. After you have created the control, right click on the control you have just created. This will bring up a window listing the properties of the Blender 3D Plug-in:



BlenderPlayer2 BlenderPlayer

(About)	
(Custom)	...
(Name)	BlenderPlayer2
BackColor	□ &H00FFFFFF&
blenderURL	
ForeColor	■ &H00000000&
frameRate	80
Height	156
Left	60
loadingURL	
showFrameCount	False
showProfileInfo	False
showProperties	False
Top	90
Visible	True
Width	246

8. You can choose to edit the properties in this window or you can choose to open a separate property window by selecting custom and clicking on the button with the three dots ("..."):



URL of main Blender file: c:\yourpresentation.blend

URL of loading animation:

Maximum frame rate: 80

9. Enter the URL of the presentation or a direct path as shown in the image. If you have created a custom loading animation (Blender Publisher license owners only) enter the URL or file path of that presentation as well. You can also enter a maximum frame rate. By reducing the frame rate, you can reduce the load the plug-in puts on the system.

6.6. Future development

The plug-ins are in development and we can already mention some of the features we plan to implement shortly.

These are ActiveX specific features:

- Standalone installer. This will make it possible to install the ActiveX control on systems without an Internet connection.
- Data persistence. This will make it possible to embed the Blender file data into your documents and distribute them as one file. For example, when creating Powerpoint presentations, you will be able to include the Blender file in the presentation itself.

Netscape specific features:

- Javascript support.
- Support more platforms (e.g. Mac OS9, OSX, Irix, Sun).
- SmartUpdate installation for Linux platform
- more than one Netscape plug-in per page

6.7. Blender 3D Plug-in FAQs

Blender 3D Plug-in FAQs

1. All plug-ins:

Q: The plug-in has loaded, but it doesn’t respond to the keyboard or mouse. What’s wrong?

A: You have to click on the plug-in to give it the keyboard/mouse focus.

Q: Playback is very slow, what can I do?

A: Like all other 3D applications, you need a 3D accelerated videocard to get good performance. Also make sure you have installed the latest drivers for your card. In addition, make sure that the HTML code contains no frameRate parameter or that is is set to a high value (see Section 6.4.3).

2. ActiveX specific:

Q: Installation on the client system fails, what can I do?

A: When the ActiveX control refuses to install it could be that Internet Explorer on the client system has been configured with strict security settings that prevent ActiveX controls being downloaded from the Internet. The security settings of Internet Explorer can be accessed in the Internet Options available from the Tools menu. If security settings are not the cause, check the windows version.

- On Windows 2000, the user type determines whether she or he is allowed to install ActiveX controls downloaded from the Internet. Depending on the Windows 2000 configuration, only Administrator type users are generally allowed to install the ActiveX control.
- On Windows98 occasionally, the plug-in refuses to install. Often, the client system has outdated or corrupt DCOM drivers. This can be solved by downloading the updated DCOM98 drivers for Windows 98 (version 1.3) from http://www.microsoft.com/com/dcom/dcom98/download.asp.

3. Netscape specific:

Q: What are the known problems of the Netscape plug-in?

A: There are some problems which are currently under investigation and will be fixed soon:

- Netscape/Linux window resize. Resizing the Netscape window can crash the plug-in under Linux.
- Netscape/Linux sound disabled. The Netscape/Linux version of the plug-in does not have sound support in this release.
- Netscape/Linux: ’exception in thread "main"’. This is a known problem; we are working on it. Just click on ok and everything will run fine.

- Netscape/Linux: no OpenGL software rendering. If you do not have a 3D accelerated X-server the plug-in may fail.

Index