

## Data System

Data System.....	2
Outliner.....	2
Pack and Unpack Data.....	2
Unpack Data.....	2
Data-Blocks.....	2
Users (Garbage Collection).....	3
Fake User.....	4
Users (Sharing).....	4
Removing Data-Blocks.....	4
Data-Block Types.....	5
Scenes.....	6
Adding a Scene.....	7
Removing a Scene.....	8
Background Set Scene.....	8
Linking to a Scene.....	8
Files.....	8
Opening Files.....	9
Usage.....	9
Options.....	10
Other File Open Options.....	10
Saving Files.....	10
Options.....	11
Importing and Exporting Files.....	12
Relative Paths.....	12
Supported Graphics Formats.....	12
Image Formats.....	13
Channel Depth.....	13
Metadata.....	14
Format Details.....	14
Cineon & DPX.....	14
OpenEXR.....	14
Output Options.....	14
Radiance HDR.....	15
Supported Video Formats.....	15
Video Formats.....	15
Advanced Encoding.....	16
Video Settings.....	16
Video Containers.....	17
Video Codecs.....	18
Options.....	18
Audio Settings.....	19
Tips.....	19
Append and Link.....	20
Proxy Objects.....	21
Known Limitations.....	22
Circular Dependencies.....	22
Object Rigid-Body Constraints.....	22

## Data System

Each `.blend` file contains a database. This database contains all scenes, objects, meshes, textures, etc. that are in the file.

A file can contain multiple scenes and each scene can contain multiple objects. Objects can contain multiple materials which can contain many textures. It is also possible to create links between different objects.

### Outliner

You can easily inspect the contents of your file by using the *Outliner* editor, which displays all of the data in your `.blend` file.

The *Outliner* allows you to do simple operations on objects, such as selecting, renaming, deleting, linking and parenting.

*Read more about the Outliner*

### Pack and Unpack Data

Bforartists has the ability to encapsulate (incorporate) various kinds of data within the `.blend` file that is normally saved outside of the `.blend` file. For example, an image texture that is an external `.jpg` file can be put “inside” the `.blend` file via File ▶ External Data ▶ Pack into `.blend` file. When the `.blend` file is saved, a copy of that `.jpg` file is put inside the `.blend` file. The `.blend` file can then be copied or emailed anywhere, and the image texture moves with it.

You know that an image texture is packed because you will see a little “Christmas present gift box” displayed in the header.

### Unpack Data

When you have received a packed file, you can File ▶ External Data ▶ Unpack into Files.... You will be presented with the option to create the original directory structure or put the file in the `//` (directory where the `.blend` file is). Use “original locations” if you will be modifying the textures and re-packing and exchanging `.blend` files, so that when you send it back and the originator unpacks, his copies of the textures will be updated.

## Data-Blocks

The base unit for any Bforartists project is the data-block. Examples of data-blocks include: meshes, objects, materials, textures, node-trees, scenes, texts, brushes and even screens.

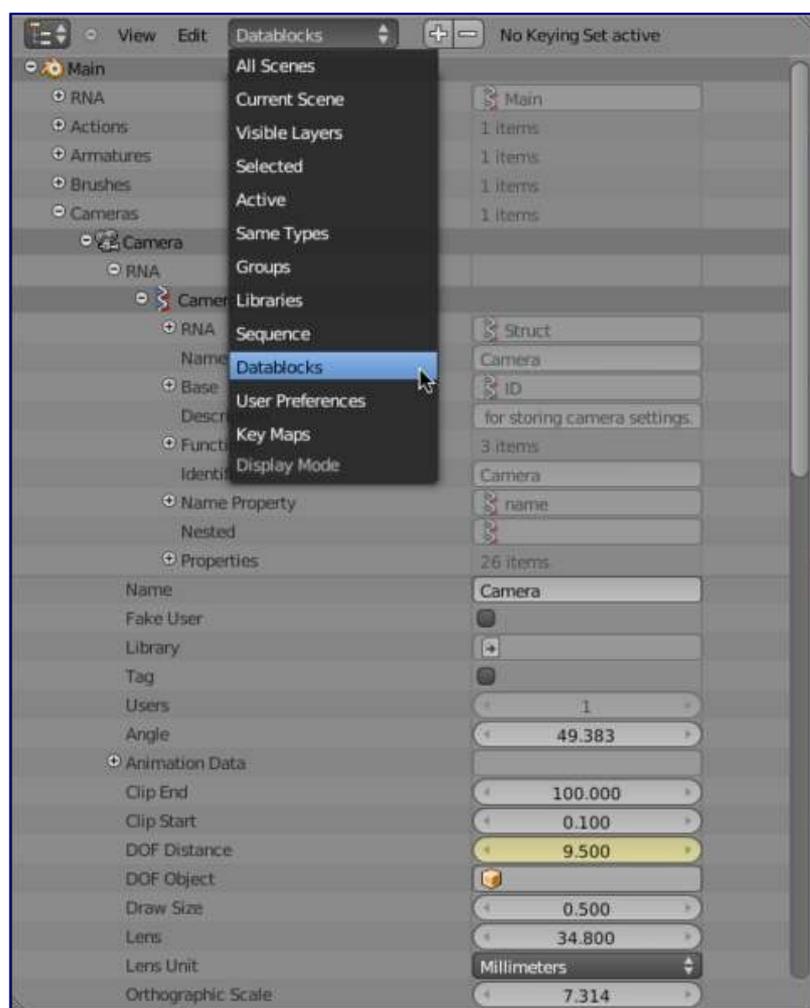
For clarity, bones, sequence strips and vertex groups are **not** data-blocks, they belong to armature, scene and

mesh types respectively.

Some common characteristics:

- They're the primary contents of the `.blend` file.
- They can link to each other, for reuse and instancing. (*child/parent, object/object-data, with modifiers and constraints too*).
- Their names are unique.
- They can be added/removed/edited/duplicated.
- They can be linked between files (*only enabled for a limited set of data-blocks*)
- They can have their own animation data.
- They can have custom properties.

When doing more complex projects managing data-blocks becomes more important, especially when inter-linking `.blend` files.



Data-blocks view

## Users (Garbage Collection)

It's good to be aware of how Bforartists, handles data-blocks life-time, when they are freed and why.

Bforartists follows the general rule where unused data is eventually removed.

Since its common to add and remove a lot of data while working, this has the advantage of not having to manually manage every single data-block.

This works by skipping zero user data-blocks when writing `.blend` files.

In some cases you want to save a data-block even when its unused (*typically for re-usable asset libraries*). see Fake User.

## Fake User

Since zero user data-blocks aren't saved. There are times when you want to force the data to be kept irrespective of its users.

If you're building a `.blend` file to serve as a library of things that you intend to link-to from *other* files, you'll need to make sure that they don't accidentally get deleted from the library file.

Do this by giving the data-blocks a *Fake User*, by pressing the *F* button next to the name of the data-block. This prevents the user count from ever becoming zero: therefore, the data-block won't be deleted. (since Bforartists doesn't keep track of how many other files link to this one.)

## Users (Sharing)

Many data-blocks can be shared among other data-blocks,

Examples where sharing data is common.

- Sharing textures among materials.
- Sharing meshes between objects (instances).
- Sharing animated actions between objects, for example to make all the lights dim together.

You can also share data-blocks between files, see.

- *linked libraries*.

## Removing Data-Blocks

As covered in Users (Garbage Collection), data-blocks are typically removed when they're no longer used.

There are some exceptions to this however.

The following data-blocks can be removed directly: Scene, Text, Group and Screen.

Other data-blocks such as groups and actions can be *Unlinked* from the *Outliner* context menu.

### Tip

Some data (images especially) is hard to keep track of, especially since image views are counted as users.

For data-blocks that can be unlinked - hold **Shift** while pressing on the *X* button, This force-clears the user-count, so the data-block will be removed on reload.

## Data-Block Types

For reference, here is a table of data-blocks types stored in `.blend` files.

**Link:** Library Linking, *supports being linked into other blend files.*

**Pack:** File Packing, *supports file contents being packed into the blend file.*

Type	Link	Pack	Description
Action	✓	X	Stores animation FCurves. Used as data-block animation data, and the Non-Linear-Editor.
Armature	✓	X	Skeleton used to deform meshes. Used as object-data & by the Armature Modifier.
Brush	✓	X	Used by paint tools.
Camera	✓	X	Used as object-data.
Curve	✓	X	Used by camera, font & surface objects.
Font	✓	✓	References font files. Used by Font object-data.
GreasePencil	✓	X	2D/3D sketch data. Used as overlay <i>helper</i> info, by the 3D-View, Image, Sequencer & MovieClip editors.
Group	✓	X	Reference object's. Used by dupli-groups & often library-linking.
Image	✓	✓	Image files. Used by textures & shader nodes.
Lamp	✓	X	Used as object-data.
Lattice	X	X	Grid based lattice deformation. Used as object-data and by the Lattice Modifier.
Library	X	✓	References to external <code>.blend</code> files. Access from the outliner's <i>Blendfile</i> view.
LineStyle	✓	X	Used by the FreeStyle render-engine.
Mask	✓	X	2D animated mask curves. Used by compositing nodes & sequencer strip.
Material	✓	X	Set shading and texturing render properties. Used by objects, meshes & curves.
Mesh	✓	X	Geometry verts/edges/faces. Used as object-data.
MetaBall	✓	X	An isosurface in 3D space. Used as object-data.
MovieClip	✓	X	Reference to an image sequence or video file. Used in the motion-tracking editor.
NodeGroup	✓	X	Collections of re-usable nodes. Used in the node-editor.
Object	✓	X	An entity in the scene with location, scale, rotation. Used by scenes & groups.
Particle	✓	X	Particle settings.

Type	Link	Pack	Description
			Used by particle systems.
Palette	✓	X	Store color presets. Access from the paint tools.
Scene	✓	X	Primary store of all data displayed and animated. Used as top-level storage for objects & animation.
Screen	X	X	Screen layout. Used by each window, which has its own screen.
ShapeKeys	X	X	Geometry shape storage, which can be animated. Used by mesh, curve and lattice objects.
Sounds	✓	✓	References to sound files. Used by speaker objects and the game-engine.
Speaker	✓	X	Sound sources for a 3D scene. Used as object-data.
Text	✓	X	Text data. Used by Python scripts and OSL shaders.
Texture	✓	X	2D/3D textures. Used by materials, world and brushes.
World	✓	X	Used by scenes for render environment settings.

## Scenes

Scenes are a way to organize your work. Usually you work with just one scene. But each `.blend` file can contain multiple scenes which share other data such as objects and materials

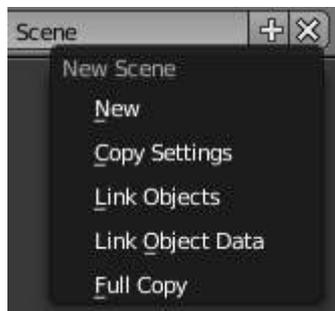
Scene management and library appending/linking is based on Bforartists's *Library and Data System*, so it is a good idea to read that manual page first if you're not familiar with the basics of that system.

You can select and create scenes with the *Scene selector*



## Adding a Scene

To add a scene, click on the scene list button, and select *Add New*. While you are adding a new scene, you have these options:



Add scene pop-up menu.

### **New**

Creates an empty scene with default values.

### **Copy Settings**

Creates an empty scene but also copies the settings from the active scene into the new one.

### **Link Objects**

This option creates a new scene with the same settings and contents as the active scene. However, instead of copying the objects, the new scene contains links to the objects in the old scene. Therefore, changes to objects in the new scene will result in the same changes to the original scene, because the objects used are literally the same. The reverse is also true.

### **Link Object Data**

Creates new, duplicate copies of all of the objects in the currently selected scene, but each one of those duplicate objects will have *links to* the object-data (meshes, materials and so on) of the corresponding objects in the original scene.

This means that you can change the position, orientation and size of the objects in the new scene without affecting other scenes, but any modifications to the object-data (meshes, materials *etc*) will also affect other scenes. This is because a *single instance* of the “object-data” is now being shared by all of the objects in all of the scenes that link to it.

More information at the *Window Type* page. This has the effect of making a new independent copy of the object-data.

### **Full Copy**

Using this option, nothing is shared. This option creates a fully independent scene with copies of the active scenes contents. Every object in the original scene is duplicated, and a duplicate, private copy of its object-data is made as well.

### **Note**

To choose between these options, it's useful to understand the difference between *Objects* and *Object Data*. See *Duplication*.

The choices for adding a scene, therefore determine just how much of this information will be *copied from* the

active scene to the new one, and how much will be *shared* (linked).

## Removing a Scene

You can delete the current scene by clicking the *X* next to the name in the Info Editor.

## Background Set Scene

You can use a scene as a background, this is typically useful when you want to focus on animating the foreground for example, without background elements getting in the way.

You can assign a *Background* to your current scene from the *Properties Editor Scene* panel.

This scene can have its own animation, physics-simulations etc, but you will have to select it from the *Scene* browser if you want to edit any of its contents.

This can also be used in combination with Linking to a Scene, where one `.blend` file has the environment, which can be re-used in many places.

## Linking to a Scene

You can link any object from one scene to another. Just open the scene where these objects are, from the 3d-view header access *Object -> Make Links...* and choose the scene where you want your objects to appear. The selected objects will be added to that scene but remain linked to the original objects.

To make them single user (independent and unlinked) in a given scene go to that scene, select them, then from the 3d-view header access *Object -> Make Single User*. You will be presented with a few options that allow you to free up the data-blocks (Object, Material, Texture...) that you want.

## Files

The options to manage files are:

### **New**

Clears the current scene and loads `startup.blend`

### **Open**

Open a blend file

### **Open Recent**

Displays a list of recently saved `.blend` files to open

### **Recover last session**

This will load the `quit.blend` file Bforartists automatically saves just before exiting. So this option enables you to recover your last work session, e.g. if you closed Bforartists by accident

### **Recover Auto Save**

This will open an automatically saved file to recover it.

### **Save**

Save the current blend file.

### **Save As**

Opens file browser to specify file name and location of save.

### **Save Copy**

Saves a copy of the current file.

### **User Preferences**

Opens the user preferences dialog.

### **Save User Settings**

Saves the current scene and preferences to startup.blend.

### **Load Factory Settings**

Restore the default scene to the factory settings.

### **Link or Append**

You don't have to load a complete file, you can load in only selected parts from another file if you wish.

### **Import**

Bforartists can use information stored in a variety of other format files which are created by other graphics programs.

### **Export**

Normally you save your work in a .blend file, but you can export some or all of your work to a format that can be processed by other graphics programs.

### **External Data**

#### **Pack into .blend**

Pack all used external files into the .blend

#### **Unpack into Files**

Unpack all files packed into this .blend to external ones

#### **Make all paths Relative**

Make all paths to external files relative to current .blend

#### **Make all paths Absolute**

Make all paths to external files absolute

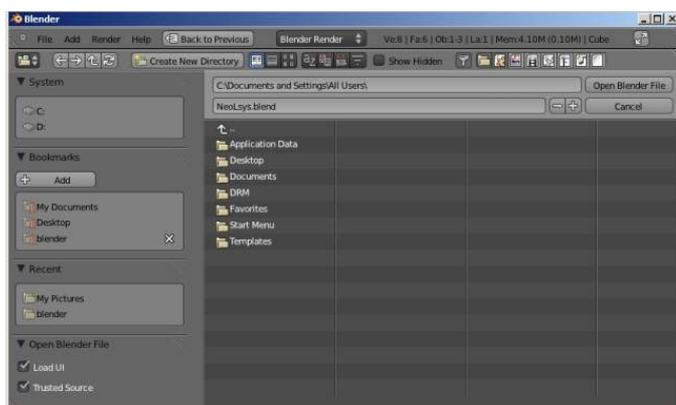
#### **Report Missing Files**

Report all missing external files

#### **Find Missing Files**

Try to find missing external files

## Opening Files



## Usage

### Reference

Menu: File ▸ Open

The upper text box displays the current directory path, and the lower text box contains the selected filename.

### Warning

For Linux and Mac-OSX users:

When existing you are **not** asked to save unsaved changes to the scene you were previously working on. So take care to save your work.

On MS-Windows there is a Save & Load option to warn on exit.

## Options

### Load UI

Inside each .blend file, Bforartists saves the user interface arrangement. By default, this saved UI is loaded, overriding any user defaults or current screen layouts that you have. If you want to work on the blend file using your own defaults, start a fresh Bforartists, then open the file browser and turn off the *Load UI* button, and then open the file.

### Trusted Source

When enabled, Python scripts and drivers that may be included in the file will be run automatically. Enable this only if you created the file yourself, or you trust that the person who gave it to you did not include any malicious code with it. See *Scripting & Security* to configure default trust options.

## Other File Open Options

From the *File* menu, you can also open files with the following tools:

### Open Recent

Lists recently used files. Click on one to load it in.

### Recover Last Session

This will load the `quit.blend` file Bforartists automatically saved just before exiting. This option enables you to recover your last work session if, for example, you closed Bforartists by accident.

### Recover Auto Save

This will allow you to open an automatically saved file to recover it.

### See also

Auto Saves

## Saving Files

### Reference

Editor: Info

## Menu: File

There are a number of slightly different methods you can use to save your blend file to your hard drive:

### Save

Save an existing blend file over itself.

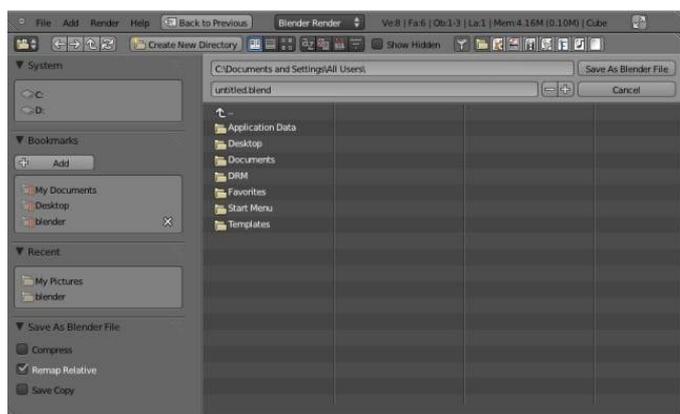
### Save As

Choose a file to save the blend to.

### Save Copy

Choose a file to save the blend to, but return to editing the original file upon completion. This can be used to save backups of the current working state without modifying the original file.

If the file name doesn't end with `.blend`, the extension is automatically appended. If a file with the same given name already exists, the text field will turn red as a warning.



## Tip

Use the **plus/minus** buttons to the right of the file name, or NumpadPlus/NumpadMinus to increase/decrease a number at the end of the file name (e.g. changing `file_01.blend` to `file_02.blend`).

## Options

The save options appear at the bottom of the sidebar.

### Compress File

When enabled, the saved file will be smaller, but take longer to save and load.

### Remap Relative

This option remaps *relative paths* (such as linked libraries and images) when saving a file in a new location.

### Save Copy

This option saves a copy of the actual working state, but does not make the saved file active.

### Legacy Mesh Format

Save the blend file, but ignore faces with more than 4 vertices (“ngons”) so that older versions of Bforartists (before 2.63) can open it.

## See also

Auto Save

## Importing and Exporting Files

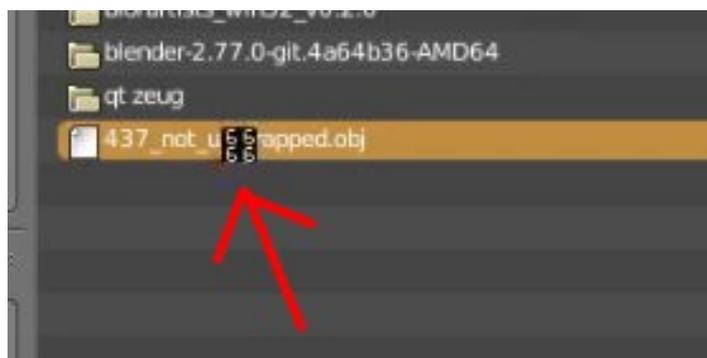
Bforartists supports import and export to and from other file formats (e.g. OBJ, FBX, 3DS, PLY... etc).

These formats can be accessed from the menus: File ▶ Import and File ▶ Export.

Popular formats are enabled by default, other formats are also supported and distributed with Bforartists, these can be enabled in the user-preferences through the use of *Add-ons*.

A list of these add-ons can be found on the [Bforartists Add-ons Catalog](#)

In case you wonder why the mouse pointer at import and export turns into this square symbol, this is not a bug. This little thing is a progress bar. Handy for large files where you could have the impression that the import or export is frozen.



## Relative Paths

Many Bforartists files reference external images or other linked `.blend` files. A path tells Bforartists where to look for these files. If the external files are moved, the blend file that references them won't look right.

When you specify one of these external files, the default option is to make the path relative. Bforartists stores a partial path evaluated relative to the directory location of the referencing blend file. This choice helps when you need to reorganize folders or move your files.

With a relative path you can move the `.blend` file to a new location provided the externally linked files are moved along with it. For example you could send someone a folder that contains a `.blend` file and a sub-folder of external images that it references.

Most file selection windows provide a *Relative Path* check box, or when you type in a path into a text field, use a double slash prefix (`//`) to make it so.

Relative paths is the default but this can be changed in the *File Preferences Tab* of the User Preferences Editor.

### Note

You can't enter relative paths into a new *untitled* blend file. Save it before linking to external files.

### Hint

If it's necessary to relocate a blend file relative to its linked resources, use Bforartists's *File Save As* function which has an option to *Remap Relative* file links.

## Supported Graphics Formats

### Image Formats

This is the list of image file formats supported internally by Bforartists:

Format	Channel Depth	Alpha	Metadata	DPI	Extensions
BMP	8bit	X	X	✓	.bmp
Iris	8bit	✓	X	X	.sgi .rgb .bw
PNG	8, 16bit	✓	✓	✓	.png
JPEG	8bit	X	✓	✓	.jpg .jpeg
JPEG 2000	8, 12, 16bit	✓	X	X	.jp2 .j2k .j2c
Targa	8bit	✓	X	X	.tga
Cineon & DPX	8, 10, 12, 16bit	✓	X	X	.cin .dpx
OpenEXR	float 16, 32bit	✓	✓	✓	.exr
Radiance HDR	float	✓	X	X	.hdr
TIFF	8, 16bit	✓	X	✓	.tif .tiff

#### Hint

If you aren't interested in technical details, a good rule of thumb for selecting an output format for your project is:

#### Use OpenEXR

if you intend to do compositing or color-grading on these images.

#### Use PNG

if you intend on-screen output or encoding into multiple video formats.

#### Use JPEG

for on-screen output where file size is a concern and quality loss is acceptable.

*All these formats support compression which can be important when rendering out animations.*

#### Note

Quicktime

On OSX, Quicktime can be used to access file formats not natively supported (such as GIF).

### Channel Depth

Image file formats support a varying number of bits per pixel. This affects the color quality and file-size.

Commonly used depths:

**8 bit (256 levels)**

Most common for on-screen graphics and video

**10,12,16 bit (1024,4096,65536 levels)**

Used for some formats focusing on photography and digital film formats (such as DPX and JPEG 2000).

**16 bit half float**

Since full 32bit float is often more than enough precision, half float can save on disk-space while providing high dynamic range.

**32 bit float**

Highest quality color depth.

Internally Bforartists's image system supports either:

- 8 bit per channel (4 x 8 bits).
- 32 bit float per channel (4 x 32 bits) - *using 4x as much memory.*

Images higher than 8 bits per channel will be converted into float on loading into Bforartists.

**Note**

Floating point is often used for HDRI,

When an image has float colors, all imaging functions in Bforartists default to use that. This includes the Video Sequence Editor, texture mapping, background images, and the Compositor.

## Metadata

Bforartists can save details such as render-time, marker, camera... etc, into the file. See: *Render Metadata*.

Only some files support this however.

## Format Details

### Cineon & DPX

Cineon is Kodak's standard for film scanning, 10 bits/channel and logarithmic. DPX has been derived from Cineon as the ANSI/SMPTE industry standard. DPX supports 16 bits color/channel, linear as well as logarithmic. DPX is currently a widely adopted standard used in the film hardware/software industry.

DPX as well as Cineon only stores and converts the "visible" color range of values between 0.0 and 1.0 (as result of rendering or composite).

### OpenEXR

ILM's OpenEXR has become a software industry standard for HDR image files, especially because of its flexible and expandable structure.

An OpenEXR file can store multiple layers and passes. This means OpenEXR images can be loaded into a compositor keeping render layers, passes intact.

## Output Options

Available options for OpenEXR render output are:

### Half

Saves images in a custom 16 bits per channel floating point format. This reduces the actual “bit depth” to 10 bits, with a 5 bits power value and 1 bit sign.

### Zbuf

Save the depth information. In Bforartists this now is written in floats too, denoting the exact distance from the camera in “Bforartists unit” values.

### Preview

On rendering animations (or single frames via command line), Bforartists saves the same image also as a JPEG, for quick preview or download.

### Compression

*This button is below the Image menu button, default set to “None”*

#### PIZ

lossless wavelet compression. Compresses images with grain well.

#### ZIP

standard lossless compression using zlib.

#### RLE

runlength encoded, lossless, works well when scanlines have same values.

#### PXR24

lossy algorithm from Pixar, converting 32 bits floats to 24 bits floats.

## Radiance HDR

Radiance is a suite of tools for lighting simulation. Since Radiance had the first (and for a long time the only) HDR image format, this format is supported by many other software packages.

Radiance (.hdr) files store colors still in 8 bits per component, but with an additional (shared) 8 bits exponent value, making it 32 bits per pixel.

## Supported Video Formats

### Video Formats

These formats are primarily used for compressing rendered sequences into a playable movie (they can also be used to make plain audio files).

A codec is a little routine that compresses the video so that it will fit on a DVD, or be able to be streamed out over the Internet, over a cable, or just be a reasonable file size. Codecs compress the channels of a video down to save space and enable continuous playback. *Lossy* codecs make smaller files at the expense of image quality. Some codecs, like H.264, are great for larger images. Codecs are used to encode and decode the movie, and so must be present on both the encoding machine (Bforartists) and the target machine. The results of the encoding are stored in a container file.

There are dozens, if not hundreds, of codecs, including XviD, H.264, DivX, Microsoft, and so on. Each has

advantages and disadvantages and compatibility with different players on different operating systems.

Most codecs can only compress the RGB or YUV color space, but some support the Alpha channel as well.

Codecs that support RGBA include:

- animation (quicktime)
- PNG TIFF Pixlet - not loss-less, and may be only available on Mac-OSX.
- Lagarith Loss-less Video Codec

### AVI Codec

AVI codec compression. Available codecs are operating-system dependent. When an AVI codec is initially chosen, the codec dialog is automatically launched. The codec can be changed directly using the *Set Codec* button which appears (*AVI Codec settings*).

### AVI Jpeg

AVI but with Jpeg compression. Lossy, smaller files but not as small as you can get with a Codec compression algorithm. Jpeg compression is also the one used in the DV format used in digital camcorders.

### AVI Raw

Audio-Video Interlaced (AVI) uncompressed frames.

### Frameserver

Bforartists puts out frames upon request as part of a render farm. The port number is specified in the OpenGL User Preferences panel.

### H.264

Encodes movies with the H.264 codec. See Advanced Encoding.

### MPEG

Encodes movies with the MPEG codec. See Advanced Encoding.

### Ogg Theora

Encodes movies with the Theora codec as Ogg files. See Advanced Encoding.

### QuickTime

Apple's Quicktime .mov file. The Quicktime codec dialog is available when this codec is installed on OSX. See *Quicktime* in Video Formats.

### Xvid

Encodes movies with the Xvid codec. See Advanced Encoding.

## Advanced Encoding



If the *H.264*, *MPEG*, *Ogg Theora*, or *Xvid* codecs are chosen, an *Encoding* panel becomes available. This has settings for encoding these file types, and other formats using FFMpeg.

FFmpeg, short for Fast Forward Moving Pictures Expert Group, is a collection of free and open source software

libraries that can record, convert and stream digital audio and video in numerous formats. It includes libavcodec, an audio/video codec library used by several other projects, and libavformat, an audio/video container mux and demux library.

## Video Settings

Here you choose which video codec you want to use, and compression settings. With all of these compression choices, there is a tradeoff between file size, compatibility across platforms, and playback quality.

When you view the *System Console*, you can see some of the output of the encoding process. You will see even more output if you execute Bforartists as `Bforartists -d`.

You can use the presets, DV, SVCD, DVD, etc. which choose optimum settings for you for that type of output, or you can manually select the format (MPEG-1, MPEG-2, MPEG-4, AVI, Quicktime (if installed), DV, H.264, or Xvid (if installed)). You must have the proper codec installed on your computer for Bforartists to be able to call it and use it to compress the video stream.

## Video Containers

### MPEG-1: .mpg, .mpeg

A standard for lossy compression of video and audio. It is designed to compress VHS-quality raw digital video and CD audio down to 1.5 Mbit/s.

### MPEG-2: .dvd, .vob, .mpg, .mpeg

A standard for “the generic coding of moving pictures and associated audio information”. It describes a combination of lossy video compression and lossy audio data compression methods which permit storage and transmission of movies using currently available storage media and transmission bandwidth.

### MPEG-4(DivX): .mp4, .mpg, .mpeg

Absorbs many of the features of MPEG-1 and MPEG-2 and other related standards, and adds new features.

### AVI: .avi

A derivative of the Resource Interchange File Format (RIFF), which divides a file’s data into blocks, or “chunks.”

### Quicktime: .mov

A multi-tracked format. QuickTime and MP4 container formats can use the same MPEG-4 formats; they are mostly interchangeable in a QuickTime-only environment. MP4, being an international standard, has more support.

### DV: .dv

An intraframe video compression scheme, which uses the discrete cosine transform (DCT) to compress video on a frame-by-frame basis. Audio is stored uncompressed.

### H.264: .avi for now.

A standard for video compression, and is currently one of the most commonly used formats for the recording, compression, and distribution of high definition video.

### Xvid: .avi for now

A video codec library following the MPEG-4 standard. It uses ASP features such as b-frames, global and quarter pixel motion compensation, lumi masking, trellis quantization, and H.263, MPEG and custom quantization matrices. Xvid is a primary competitor of the DivX Pro Codec.

### Ogg: .ogg, .ogv

A free lossy video compression format. It is developed by the Xiph.Org Foundation and distributed without licensing fees.

### Matroska: .mkv

An open standard free container format, a file format that can hold an unlimited number of video, audio,

picture or subtitle tracks in one file.

**Flash: .flv**

A container file format used to deliver video over the Internet using Adobe Flash Player.

**Wav: .wav**

An uncompressed (or lightly compressed) Microsoft and IBM audio file format.

**Mp3: .mp3**

A highly-compressed, patented digital audio encoding format using a form of lossy data compression. It is a common audio format for consumer audio storage, as well as a de facto standard of digital audio compression for the transfer and playback of music on digital audio players.

## **Video Codecs**

**None**

*For audio-only encoding.*

**MPEG-1**

See Video Formats.

**MPEG-2**

See Video Formats.

**MPEG-4(DivX)**

See Video Formats.

**HuffYUV**

Loss-less video codec created by Ben Rudiak-Gould which is meant to replace uncompressed YCbCr as a video capture format.

**DV**

See Video Formats.

**H.264**

See Video Formats.

**Xvid**

See Video Formats.

**Theora**

See Ogg in Video Formats.

**Flash Video**

See Video Formats.

**FFmpeg video codec #1**

A.K.A. FFV1, a loss-less intra-frame video codec. It can use either variable length coding or arithmetic coding for entropy coding. The encoder and decoder are part of the free, open-source library libavcodec in FFmpeg.

## **Options**

**Bitrate**

Set the average bitrate (quality), which is the count of binary digits per frame. See also: `ffmpeg -b:v`

**Rate**

The bitrate control also includes a *Minimum* and a *Maximum*.

**Buffer**

The decoder bitstream buffer size.

**GOP Size**

The number of pictures per Group of Pictures. Set to 0 for “intra\_only”, which disables inter-frame video. From ffmpeg docs: “For streaming at very low bitrate application, use a low frame rate and a small GOP size. This is especially true for RealVideo where the Linux player does not seem to be very fast, so it can miss frames”

## Autosplit Output

If your video is HUGE and exceeds 2Gig, enable Autosplit Output. The main control over output filesize is the GOP, or keyframe interlace. A higher number generally leads to a smaller file, but needs a higher-powered device to replay it.

## Mux

Multiplexing settings.

### Rate

Maximum bit rate of the multiplexed stream.

### Packet Size

(Undocumented in ffmpeg)

## Note

### Standards

Codecs cannot encode off-the-wall video sizes, so stick to the XY sizes used in the presets for standard TV sizes.

## Audio Settings

Audio is encoded using the codec you choose.

### Audio Codecs

#### MP2

A lossy audio compression format defined by ISO/IEC 11172-3.

#### MP3

See MP3 in Video Formats above.)

#### AC3

Audio Codec 3, an audio compression technology developed by Dolby Laboratories.

#### AAC

Advanced Audio Codec,” a standardized, lossy compression and encoding scheme for digital audio.

*AAC generally achieves better sound quality than MP3 at similar bit rates.*

#### Vorbis

An open-standard, highly-compressed format comparable to MP3 or AAC.

*Vorbis generally achieves better sound quality than MP3 at similar bit rates.*

#### FLAC

Free Loss-less Audio Codec. Digital audio compressed by FLAC’s algorithm can typically be reduced to 50-60% of its original size, and decompressed into an identical copy of the original audio data.

#### PCM

Pulse Code Modulation, a method used to digitally represent sampled analog signals. It is the standard form for digital audio in computers and various Blu-ray, Compact Disc and DVD formats, as well as other uses such as digital telephone systems

#### Bitrate

For each codec, you can to control the bitrate (quality) of the sound in the movie. This example shows MP3 encoding at 128kbps. Higher bitrates are bigger files that stream worse but sound better. Stick to

powers of 2 for compatibility.

### **Samplerate**

Samplerate controls the number of samples per second of the audio. The default, 44100, is standard for many file types, including CD audio, and produces a high quality sound.

### **Volume**

Set the output volume of the audio.

## **Tips**

Choosing which format to use depends on what you are going to do with the image.

If you are animating a movie and are not going to do any post-processing or special effects on it, use either **AVI-JPEG** or **AVI Codec** and choose the XviD open codec. If you want to output your movie with sound that you have loaded into the VSE, use **FFMPEG**.

If you are going to do post-processing on your movie, it is best to use a frame set rendered as **OpenEXR** images; if you only want one file, then choose **AVI Raw**. While AVI Raw is huge, it preserves the exact quality of output for post-processing. After post-processing (compositing and/or sequencing), you can compress it down. You don't want to post-process a compressed file, because the compression artifacts might throw off what you are trying to accomplish with the post-processing.

Note that you might not want to render directly to a video format. If a problem occurs while rendering, you have to re-render all frames from the beginning. If you first render out a set of static images (such as the default PNG, or the higher-quality OpenEXR), you can stitch them together with an Image Strip in the *Video Sequence Editor (VSE)*. This way, you can easily:

- Restart the rendering from the place (the frame) where the problem occurred.
- Try out different video options in seconds, rather than minutes or hours.
- Enjoy the rest of the features of the VSE, such as adding Image Strips from previous renders, audio, video clips, etc.

## **Append and Link**

These functions help you reuse materials, objects and other *data-blocks* loaded from an external source `.blend` file. You can build libraries of common content and share them across multiple referencing files.

*Link* creates a reference to the data in the source file such that changes made there will be reflected in the referencing file the next time it is reloaded.

Where as *Append* makes a full copy of the data into your `.blend`. You can make further edits to your local copy of the data, but changes in the external source file will not be reflected in the referencing file.

### **Reference**

Mode: All Modes

Menu: *File* → *Append or Link*

In the *File Browser* window navigate to the external source `.blend` file and select the data-block you want to reuse.

Options:

### Relative Path

Available only when linking, see *relative paths*.

### Select

Makes the object *Active* after it is loaded.

### Active Layer

Enabled by default, the object is assigned to the visible layers in your scene. Otherwise, it is assigned to the same layers it resides on in the source file.

### Instance Groups

This option links the Group to an object, adding it to the active scene.

When you select an Object type, it will be placed in your scene at the cursor. Many other data types - cameras, curves, and materials for example - must be linked to an object before they become visible.

Newly added Group types are available in `Add ▸ Group Instances` in 3D View, or for NodeTree groups, the same menu in the Node Editor.

Look in the Outliner, with display mode set to *Bforartists File*, to see all your linked and appended data-blocks. `Ctrl-LMB` on a file name allows you to redirect a link to another file.

## Hint

You cannot move a linked object. Its position is defined in its source file.

If you want to modify the object locally you can either:

### Use *Dupli-Groups*

Instead of linking in *Objects* directly, its often more useful to link in *Groups*, which can be assigned to empties and moved, while maintaining the link to the original file.

Its also useful to be able to add/remove objects from the group without having to manage linking in multiple objects.

### Make Objects Local

Use `Object ▸ Make Local ▸ Selected Objects` to make the position editable.

This means that object data (animation, constraints, modifiers...) will be local to your `.blend` file. But the object-data will still be linked and remain immutable.

## Note

Appending data you already have linked, will add objects / groups to the scene, but will keep them linked (and un-editable).

This is done so existing relationships with linked data remain intact.

## Proxy Objects

Used with rigged models, proxy objects, allow specified bone layers to be linked back to the source file while the remainder of the object and its skeleton are edited locally.

`Ctrl-Alt-P*` makes the active linked object into a local proxy, appending `_proxy` to its name.

*\* I hate it. Dear Blender developers. Ever thought that users wants to know WHAT TOOL is meant, not WHAT HOTKEY ? - To Do*

Set the *Protected Layers* in the source file using using the Skeleton panel of the Armatures tab. See Armature Layers. The bones in protected layers will have their position restored from the source file when the referencing file is reloaded.

## Known Limitations

For the most part linking data will work as expected, however there are some corner-cases which aren't supported.

## Circular Dependencies

In general dependencies shouldn't go in both directions.

Attempting to link or append data which links back to the current file will likely result in missing links.

## Object Rigid-Body Constraints

When linking objects **directly** into a `.blend` file, the *Rigid Body* settings won't be linked in since they're associated with their scenes world.

As an alternative you could link in the entire scene and set it as a Background Set Scene.