The Show

Manual

System requirements:

~ AMD Athlon 1200 MHz 512MB RAM
GPU Minimum : ~GeForce 6200 128MB
tested with Windows XP
Sound card, Mouse
DirectX 8.0 and filter codecs for the videos
Unzip to hard disk

http://www.show-homepage.com/

mailto: Buelles_Marc@t-Online.de

Overview

The idea of this program is, to have some objects flying around in space.

Legal disclaimer

The program uses the Microsoft Retained Mode, DirectDraw 2.0 and DirectDraw 7.0. For playing videos, it uses the DirectX 8.0 Direct Show toolkit and the installed filters.
The LccWin32 Compiler, version 3.3 was used. When I downloaded the files, there was no difference commercial or Non commercial.
The zip-files on the show home page were created with a zip program without non commercial restriction.

The program seems to be legal.

I will no longer visit the Microsoft Homepage and Nvidia homepage, to improve the show program. The compiler will not be updated for free, to keep the show free for commercial purposes. The DirectX toolkits 9.x or future toolkits will not be used.

The provided objects (x-Files) were created with the freeware version 1.2 of gmax, and exported with a maxscript and copy / paste.
Anim8or, a 3D computer animation program was used too, to make objects. See http://www.anim8or.com/main/ .

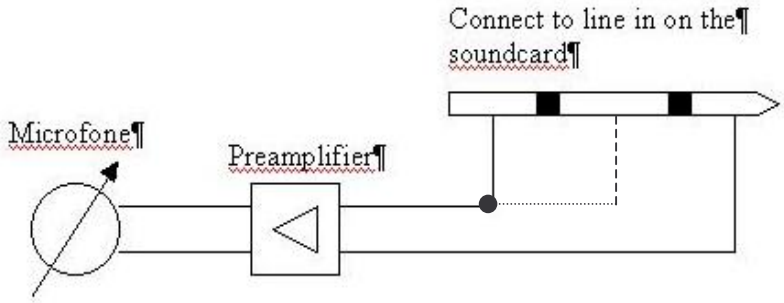
I guess it was said “So far, but not more”.

There was an invitation, to follow the technology via newsletter.

August 2006

The Sound things

It must be a 16-bit sound card.
Use one channel of the linein jack on the sound card, to get the sound input. See the picture on the right :



In the computer, the values are then limited from –32768 to 32767.

Start small show.
Select Debugging Sound in the More dialog.
The values minold and maxold show the lowest value and the highest value, that have been sampled, start time was the click to the sound radio button in the More dialog.
The display AInOlCount (Analog in Overload Counter) shows how often the values –32768 and 32767 have been sampled.

These three indicators can be resetted by clicking to the Sound radio button in the More dialog.

I have a simple microfone without preamplifier connected to line in and I get minold and maxold values from –5000 to 10000 approximately, if I speak very loud in the microfone. So, a signal input like that is approximately good enough. The microphone could be a little bit more sensitive.

To do more settings for sound input, see the channels dialog.

At the moment, the program takes 1024 sound samples way back from the blitting moment, to calculate the frequency channels. Together with the smplerate of 22050 Hz, there is a minimum framerate of **21,5 fps** (46,4mSec). So, if the framerate is lower than 21,5 fps, the sound signal is only partially used for frequency calculation.

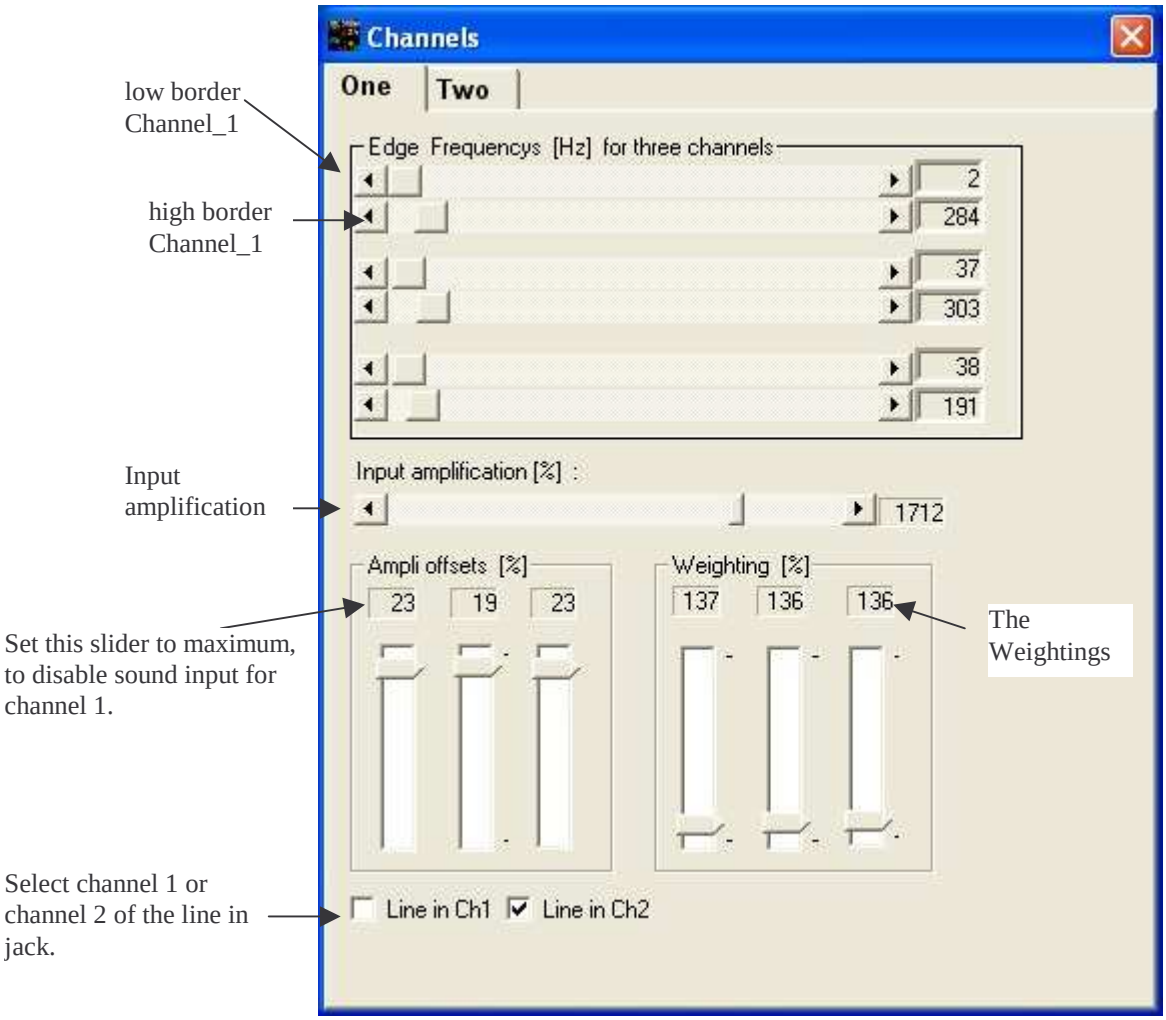
To get the frequencies from the sound input, I did very easy and downladed an algorithm from Don Cross. This algorithm can be found at http://www.fftw.org/links.html .

The dialogs

When a dialog is moved, the other dialogs are not repainted, if the program detects low performance bandwidth left.

Channels “One”

The programm supports three frequency channels (Channel_1, Channel_2, Channel_3) for the objects.

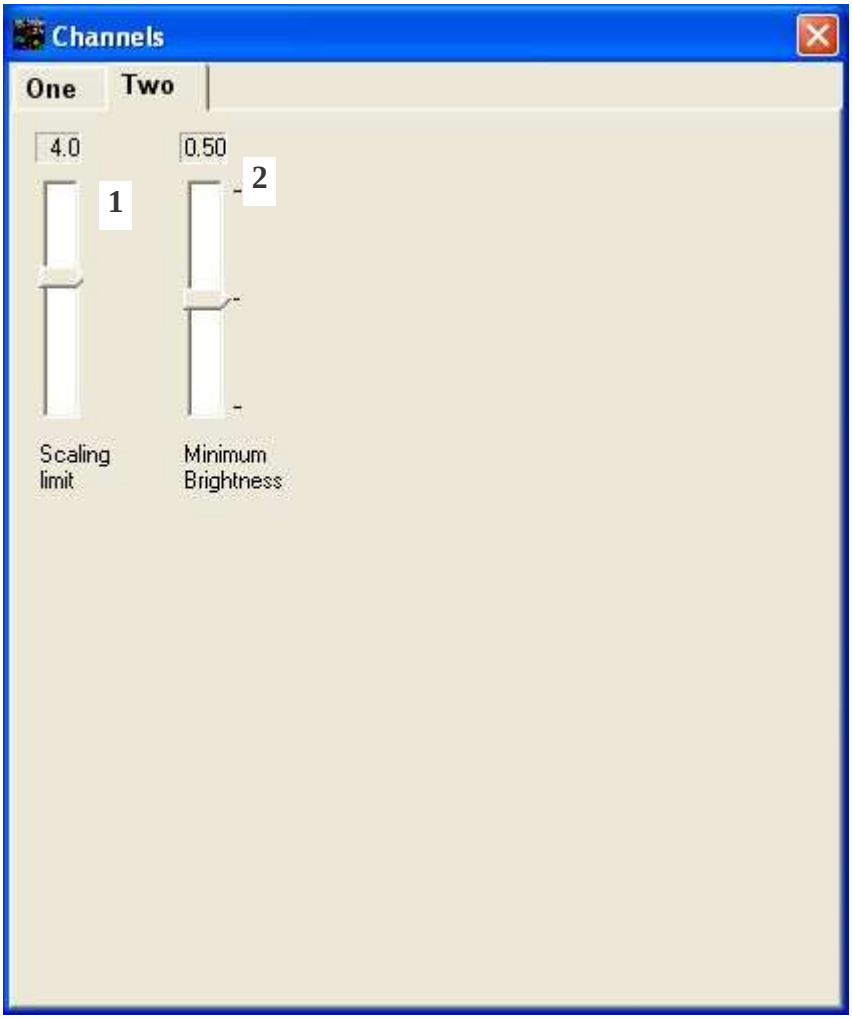


How to do the adjustements:

- Set the edge frequencies as shown above.
- Set the three weightings to half way.
- Set the three amplitude offsets to zero.
- Vary the input amplification (This is a software amplification.).
- Vary the weightings.

Set the amplitude offsets as you like it.

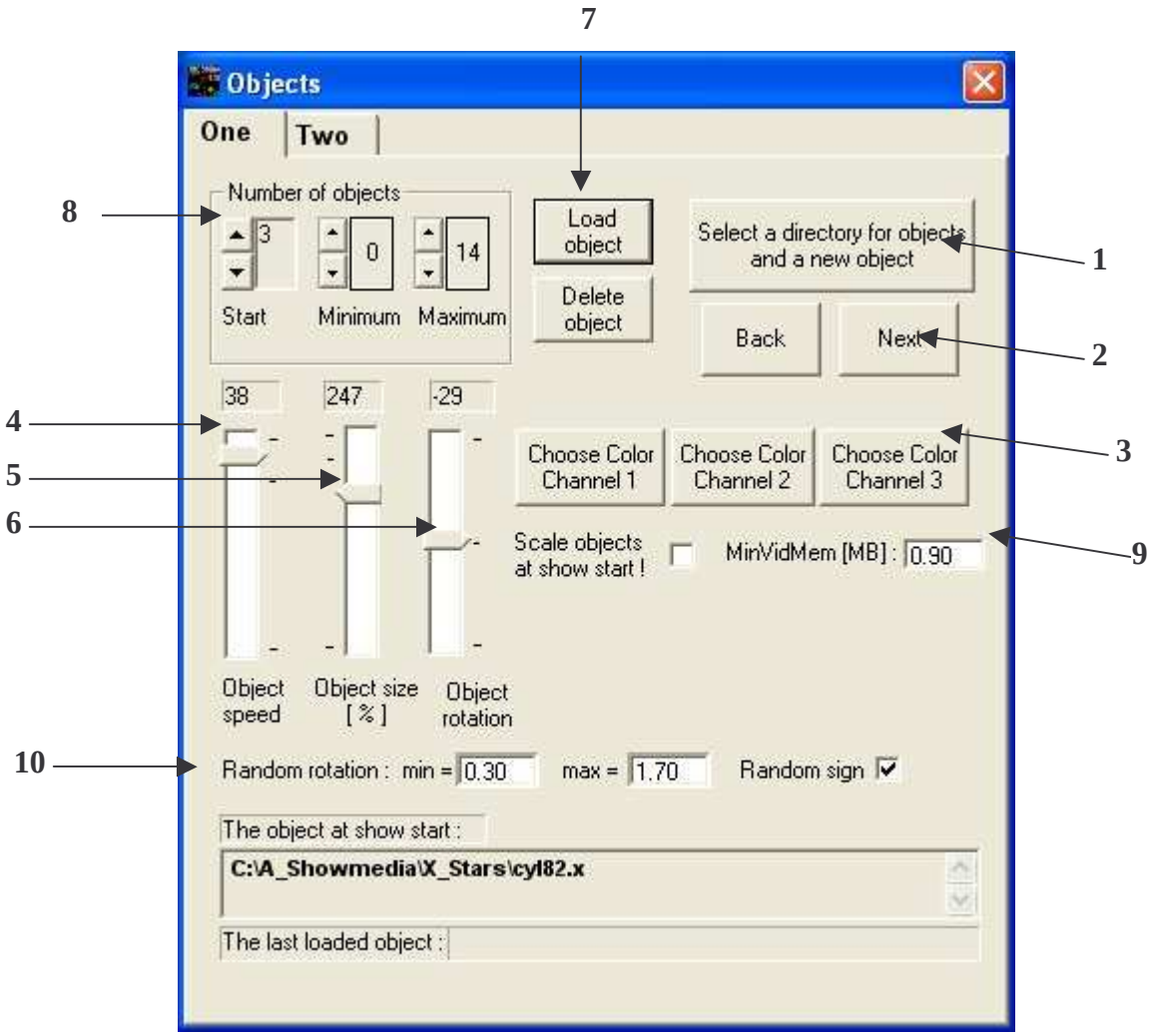
Channels “Two”



1. Scaling limit
The objects are scaled bigger and smaller, depending on the sound signal. This slider sets the maximum factor for bigger scaling. If an object is scaled too big, the viewer turns into the object. The object is then transparent (not visible). The value **four** is a proper setting.

2. Minimum brightness
The objects look darker and brighter, depending on the sound signal. This slider sets the minimum brightness.
Set to 1.0 : The objects have always maximum brightness. With “Painton” , high values prevent the objects beeing painted dark.

Objects “One”



1. Select a directory for objects and a new object
Select a x-file. When you push the button, the the fileopendialog appears and shows the name of the x-file, that will be loaded, when you push ”Load object”. After having made a selection, the selection is shown. If an x-file is selected with this dialog, the x-files are counted and reinitialized. So you can add, delete and rename files, while the program is running, but it is necessary, to reinitialize by selecting a file. During initialisation, the alphabets are changed to upper and then the filenames are sorted in ASCII-Order. At program start, the initialisation is done automatically. With Full screen, the right mouse button (help function) is disabled.

2. Select the next object in the directory; go one object back
Select the next x-file in the directory. Push the button Load object to load and show the x-file.
Choose the colors for the channels/objects. Example: If 5 objects are loaded, there are 2 for bass, 2 for middle and 1 for high.

3. Choose a color for the object
With Full screen, the right mouse button (help function) is blocked.

4. Object speed
Speed of the objects.

5. Object size
Change the size of the objects.When an object is loaded, it is scaled along the x-axis to the size 1.0 and then the resulting scaling factor is used to scale the object in y-direction and z-direction. The object has then the size 100%. This slider offers an additional scaling.

6. Object rotation
The rotation of the objects

7. Load object
Loads a new object. When a new object **name.x** is loaded, the program loads too a picture, to be wrapped around the object. First try is, to load a picture **name.jpg**, which must be located in the same directory as the object. If the picture name.jpg contains more colors than black and white, the color of the objects/channels should be set to white, to see each color of the picture. If no picture name.jpg is found, the program tries to load **name.bmp**. If no picture name.bmp is found, the picture **start.bmp** is loaded. It is allowed, to edit start.bmp. The textures should have a width in pixels, which is a power of two and a height in pixels, which is a power of two (Examples: 16x16, 128x128, 512x512), to improve the performance.

6. Number of objects
Set the number of objects to be loaded at show start.

9. MinVidMem [MB]
Set this value bigger than the biggest amount of memory, that an object will need, together with texture and wrp file and additionally approximately 100 kBytes. The 100 kBytes are used internally by DirectX, to store some characteristics.
Sizeof textue[Bytes] = height of bitmap * width of bitmap * 4 Bytes

1.00 Megabytes is a proper value. Change this value, if very very big objects and textures or very small display card.

10. Random rotation
When a object is loaded, a random number in the range from min to max is multiplied to to the value of the slider.
With the random sign, the spin will be randomly inverted.

About x-files:

Not each x-file fits to the program.
Each x-file has a file size. The amount of x-file data is the filesize multiplied with the number of objects.

Loading much too much x-file data makes the program uncontrollable.

Loading much too much x-file data makes the hardware driver reach the limit of performance. The program tries then to delete objects automatically. The program may become uncontrollable for some seconds. One object is hold as minimum. If the hardware is still overloaded, the framerate is decreased. Keyboard 8 restores the framerate. Restore framerate is rarely used, because the the performance bandwidth of computers has increased and the source code is improved (version 2.8).

The amount of data that can be loaded without problems depends on

- speed of PCI-bus,
- speed of AGP-bus,
- selected frame rate,
- power of CPU,
- power of display card (Riva TNT2, GeForce).

Three cubes (cube.x), each 1KB should be loaded, without problems, with a CPU speed of 600 MHz.

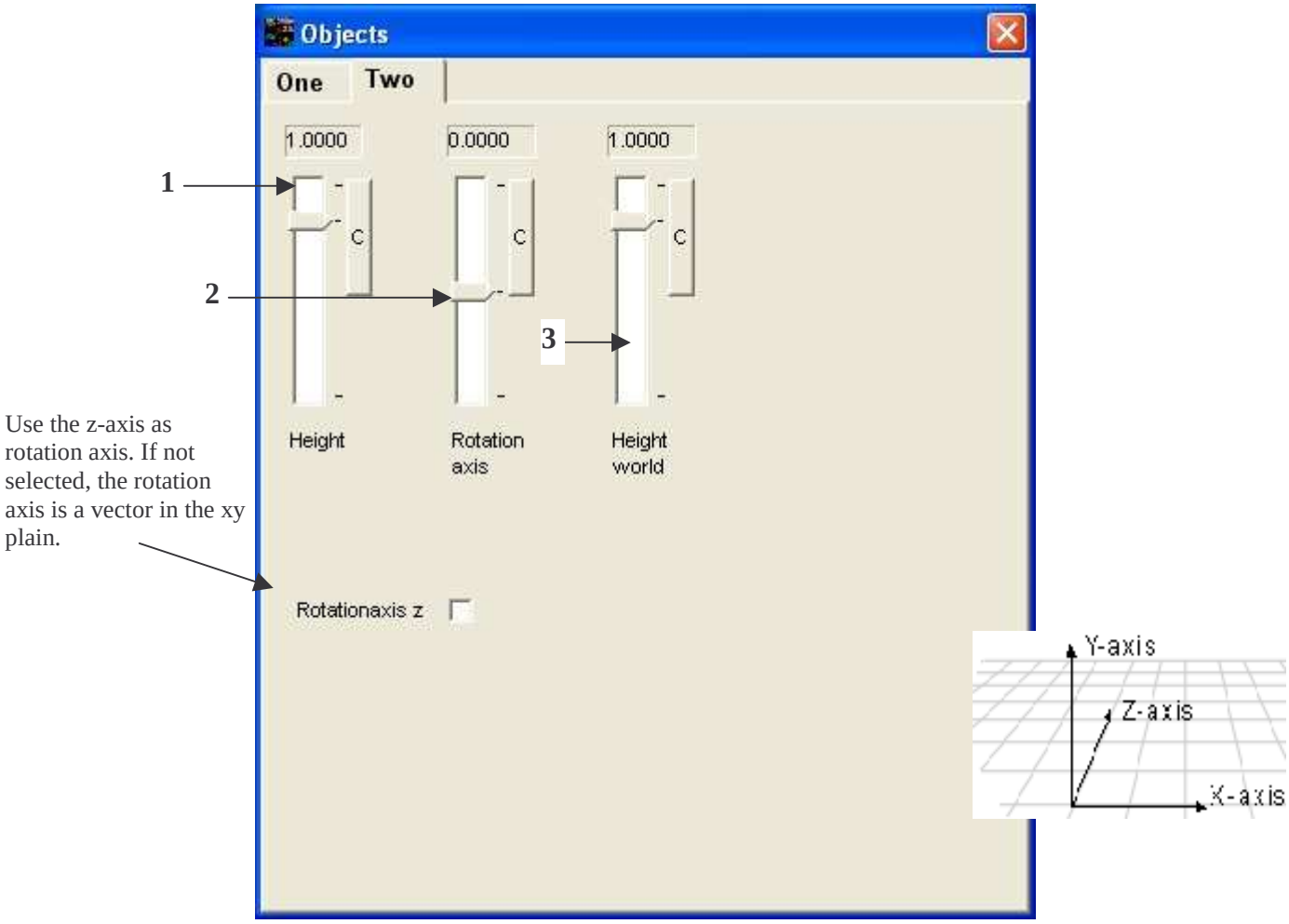
A good homepage for 3D objects is http://www.3dcafe.com/ . I downloaded from there many 3-D models in 3ds file format. These 3ds files are then to be converted to x-files. Use any suitable converting program, for example conv3ds.exe, which seems to be free for non commercial purposes.

```
Example:  C:\A_Show\XFiles_1\conv3ds.exe -m diamond1.3ds
```

conv3ds.exe can be downloaded as a zip filefrom http://www.microtower.com. There are some more locations, to download this program.

I downloaded from 3DCafe also a 3D-model viewer, named quick 3D viewer.

Objects “Two”

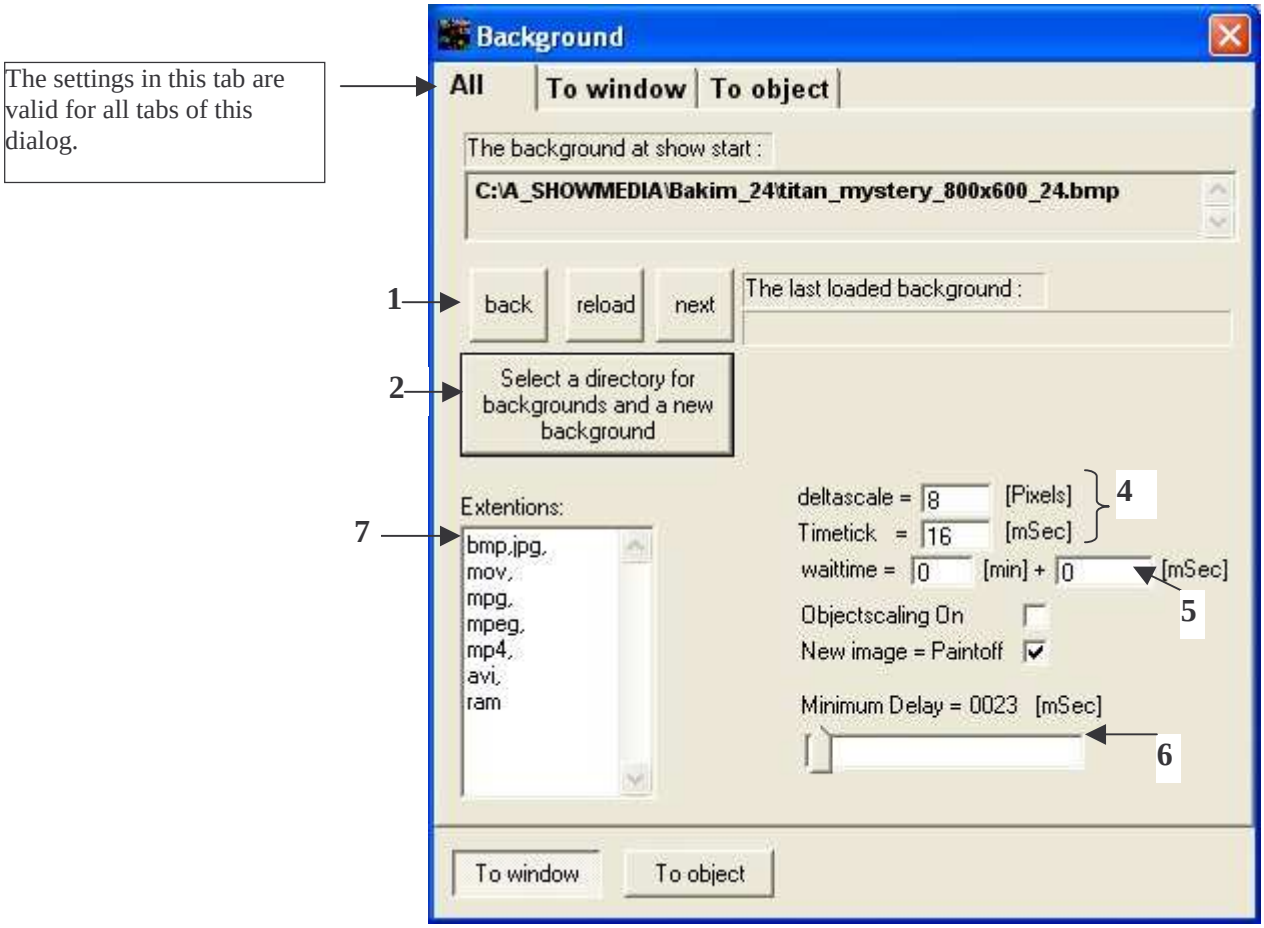


1. Set the height of the objects
To scale in direction of the rotation axis. This scaling factor is applied too, in the wrap dialog and for background objects.

2. Rotate the the objects in the xy plain
This rotation is applied too, in the wrap dialog.
Clicking to the slider or to C stops the rotation for a short moment, and aligns the objects along the x-axis. Keeping the radial relationship of the objects and the rotation is not implemented.

3. Height world
To adjust the height vertically. This scaling fctor is applied too, in the wrap dialog and for background objects.

Background All



1. Load a new background
back: The previous background is displayed.
reload: The same background is loaded again.
next: The next background in the directory is displayed.

2. Select a directory for backgrounds and a new background
Select a new background in any directory. When you push the botton, the the fileopendialog appears. If you select an image file with this dialog, the files are counted and initialized. So you can add, delete and rename files, while the program is running, but it is necessary, to reinitialize by selecting a file. During initialisation, the alphabets are changed to upper and then the filenames are sorted in ASCII-Order. At program start, the initialisation is done automatically. With Fullscreen, the right mouse button (help function) is disabled.

4. deltascale, Timetick and Obj_Scale_On
deltascale:
The picture grows deltascale pixels each frame and each direction. Allowed values are from 1 to 99; Grow from center uses deltascale / 2 pixels in each direction.
Timetick:
The time in milliseconds, from one picture to the next, while the new picture comes in. Allowed values are from 1 to 25. If the displaycard can't go as fast as selected, it goes as fast as possible.

Obj_Scale_On:
If selected, the objects take some time, to disappear and to return.

5. waittime

The time, until the objects return. If the waittime is bigger than 2 seconds, a little dialog is displayed. The window is not repainted, if this dialog is moved with two screens and full screen. This is not a fatal error.
The time in milliseconds, to look at the new picture, before the objects return. Allowed values are from 0 to 10000. This value is valid, even if the objects are scaled so small, that they are not visible.
If no objects are loaded, the program uses the waittime 0 msec.

6. Minimum Delay
The mimimum delay, from one background to the next. Comes in sight, with keyboard N, keyboard B and Next Background in the Auto dialog, if the four time fields are set to small values or zero.
Setting the four time fields for next background to zero sets an additional timer, to increase steadiness.
Debug D3DRM_1 shows the value **BackgrndCnt.** Together with a look at the watch, it is possible to calculate the average time per picture. Add approximately 15% and set the Minimum Delay. The pictures will then be steady from one to the next. Immediate was selected.

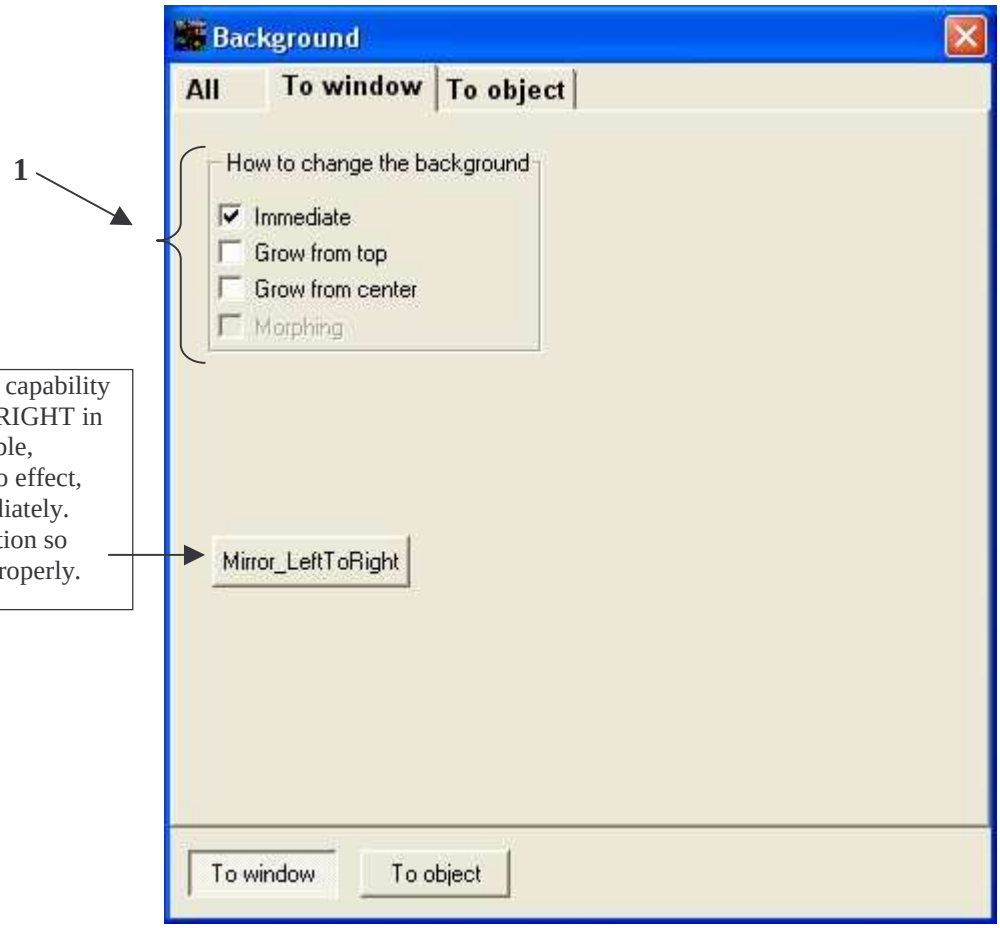
7. Extentions
jpg and bmp are implemented in the program. Other picture formats are not supported.
Videos: Use graphedit.exe, to proov, that the necessary filters are installed. Then add the file extention to the list. Extentions are not case sensitive. They have to be seperated by comma, CR is optional. Blanks are taken as part of the extention.
Select a background, to make changes in the extentions available for the show program.
Download graphedit.exe(March 2009): http://www.digital-digest.com/dvd/downloads/showsoftware_graphedit_141.html
The program reserves 250 characters for the extentions. The Show decodes .lnk files. The referenced file must be decodable and enumerated in the extentions list.

A picture (bitmaps and jpgs) for the background must have a **minimum size of 20 * 20 pixels .**
Only uncompressed bitmaps are loaded. (Nearly all bitmaps are uncompressed.).

Unloadable backgrounds are skipped and the next is loaded. If the back button is used, the program loads always the same background, next to the unloadable background in ASCII order.

Repeated switching of the same background image from window to object results in a loss of quality (a blur filter…), because the surfaces have different sizes.
The size of the window surface is the screen resolution for fullscreen, or size of the window for windowed mode.
The size of the object's surface is the size of of the texture, loaded with the object.

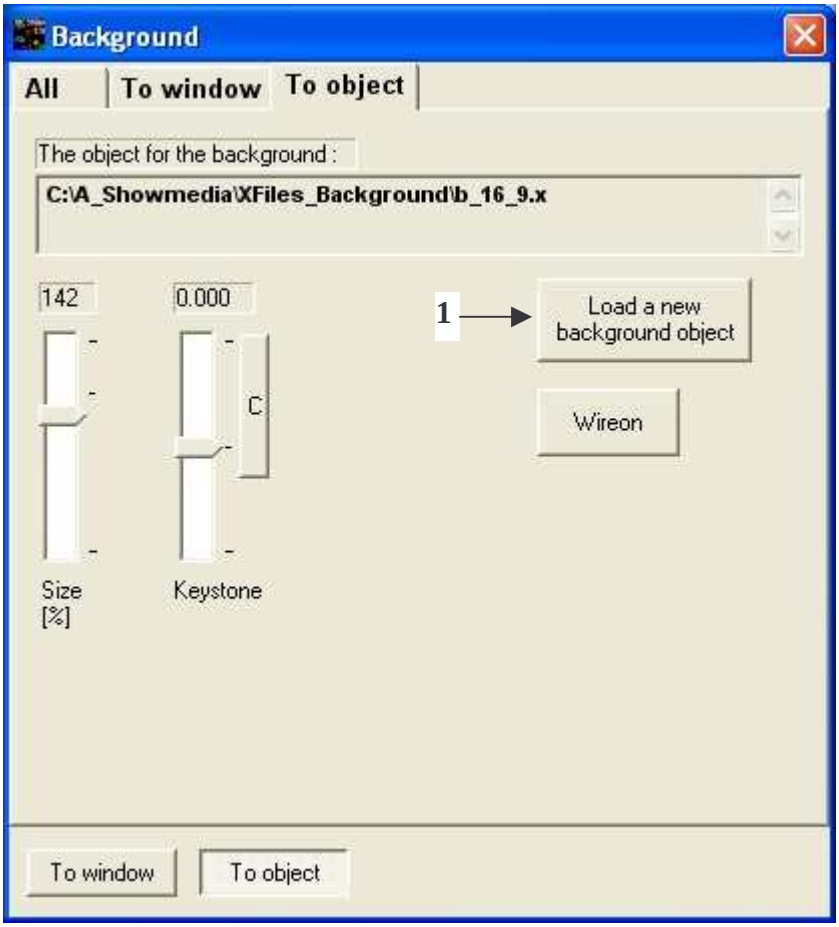
Background To window



1. How to change the background
The mode, how to replace the old image. If more than one mode is selected, a random selection is taken. If none of the radio buttons is selected, the background is black.

Background to object

If a new background object is loaded, while a show is on, it is displayed in the foreground, but not in the background. This bug is not yet fixed.
The show must be started again, to have the object in the background.



The sliders in the objects dialog **Height**, **Rotation axis** and **Height world** are applied too to this background object.

GeForce 6600, Radeon 9200 : Full screen successfully tested

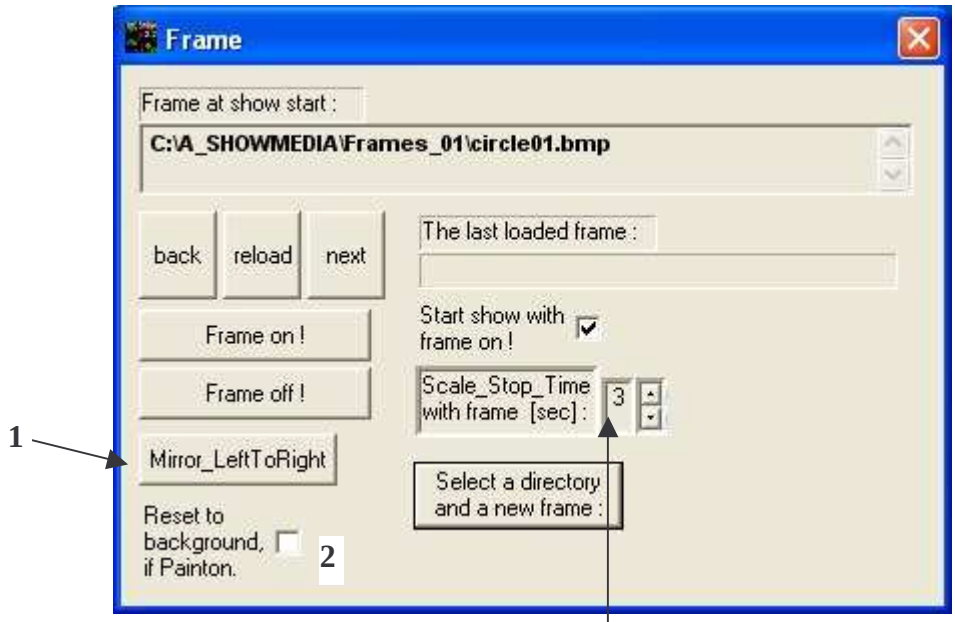
Mirror left to right can be done, by changing the parameter su of the wrap for the object to negative. See **b_16_9.x** and **b_16_9LR.x**. Mirror upside down can be done, by changing the parameters su and sv of the wrap to positive. See **b_16_9.x** and **b_16_9USD.x**. su <0 and sv>0 mirrors left to right and upside down. These files are stored in **X_Files.zip**. More vertices increase the needed performance bandwidth but also the accuracy of this "Keystone" transformation.
If only a part of the texture is used, it is more complicated.

Paint mode is not available, where the background object is visible on the screen.
The displayed background should have the same width/height ratio as the picture, wrapped around the object.
Background to object needs more performance bandwidth, than background to window.
Transparency is available too, but the Filter must be set to 0xff (dark black).
The width and height of the object's texture limits the resolution for the displayed background. Example: A video of 1024*1024 pixels and a texture of 512*512 pixels will result in a resolution of 512*512 pixels on the object.
The bigger width and height of the object's texture, the more performance bandwidth is needed, to render the object.
The object is located on the z-axis `z=30.0`.

Using a texture with a height and width of power of two improves the performance.

1. Load a new background object
Needs a lot of time, to get performed.

Frame



Scale_Stop_Time with frame [sec]
The time from klick to the Scale_Stop button to end of show. 5 Seconds is a proper value. Set bigger values, if the pictue has not changed to black, when the show stops. This value does not effect the speed of smaller scaling.

With Fullscreen, the right mouse button (help function) is disabled !

With this frame option it is possible to cover a part of the screen.
The white pixels of the frame are transparent, all other colors are opaque. In the root directory of the show program, there are two example frames **frame.bmp** and **frame_01.jpg** .

The Scalestop, to stop the Full screen uses a daring function, that some GPUs and drivers don't perform. The frame is then not scaled, no error message is written to Show_Err.txt.

1 Mirror_LeftToRight
Mirrors the frame; The data, that seems to be hidden by the frame does not exiset.

2 Reset to background, if Painton.
This option is used, with the paint mode.
On : The selected background is displayed.
Off : A newly selected frame is displayed, but the old frame is not deleted , until it is painted away by the the objects.

Video “One”

The digital audio output on the sound card is very high dynamic, as each digital signal is. There is no warranty, for the loud speaker.
Audio only files are not played !!!

If a system overload by video is detected, frames will be dropped. The monitor is automatically switched off; may be switched on again... See ‘More dialog’, debug video. The value **dwDropped** counts the dropped frames.

Plays video files, depending on the installed filters and the extentions in the background dialog. To play the selected video, the Show constructs a filter graph from the installed filters, but does the rendering itself. It needs an unrendered pin of major type video and subtype 32 or 24 bit and (obviously) a kind of DirectDraw capability. The ffdshow MPEG-4 video decoder offers this, if the overlay mixer option is switched off.
New versions of ffdshow audio and video decoder must have a checkmark set in the ffdshow compatibility manager, that they are allowed to be loaded into show.exe.

The K-Light Codec Pack http://www.k-litecodecpack.com/ (december 2006) contains many filters, to decode video formats.

Some more filter examples:
For mp4, the show may need some filters which are not installed on the computer system. The Haali Media Splitter and the ffdshow MPEG-4 video decoder can be used, to play video files. This ffdshow decoder needs the option ‘Use overlay mixer’ switched off !! See http://www.matroska.org/downloads/windows.html (december 2006) !
To decode avi files, the indeo filters may be used. See http://www.free-codecs.com/download/Indeo_Codec.htm (december 2006) !

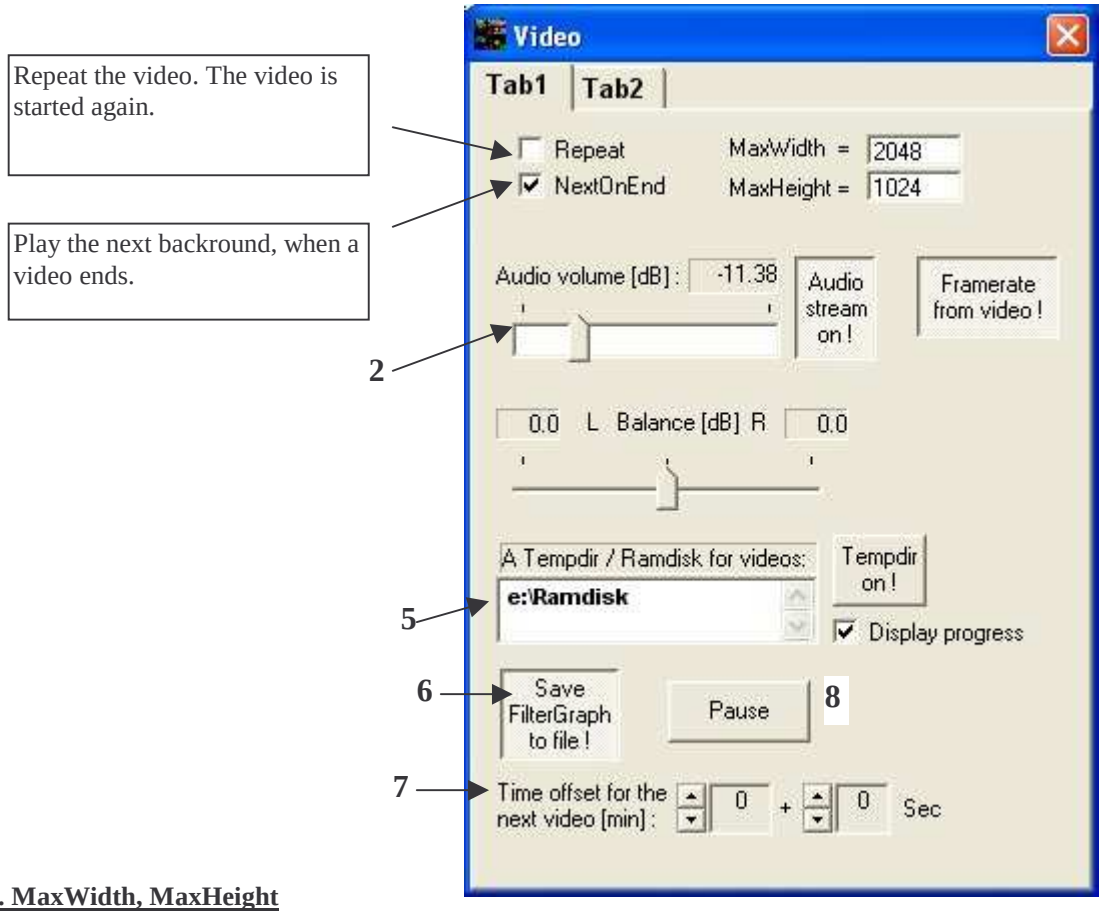
To decode mov, here is an example for a source filter: The file quicktime.ax must be registered. quicktime.ax comes with the Media Player Classic Version 1.56. Quicktime.ax is registered when the player is installed, if not, it can be registered with regsvr32.exe. The file quicktime.ax contains the Cyberlink Quicktime Sourcefilter. See http://www.free-codecs.com/download/QuickTime_Alternative.htm . Media Player Classic Version 1.56 is quite outdated. Today, Haali media splitter and ffdshow decoders may support mov.

See the more dialog, debug video. The **video.dwu** value shows, how long it takes, to get 1 picture from the video stream. This update time varies from one picture to the next. The value **video.dwu** is valid, if **1 Surface sync update** is selected.
Example: Even if 24 msec per frame are selected in the More dialog, some updates will take 30, 50 or 80 milliseconds. The 30, 50 or 80 milliseconds updates will show the frame delayed.

The waittime in the Background dialog is ignored with videos.

The debug video option shows a milliseconds value in the first line. This is the fastest rate, to get pictures from the video stream. Full screen runs always with the framerate, selected in the more dialog. If enough time has passed, a new picture is fetched from the stream.
Bad: The video will not always play with the “best” framerate.
Good: This may not be visible.

The show loads now too GRF files (the filter graphs stored in a file). They can be built with the program graphedit.exe. A filtergraph becomes invalid, if the used filters are uninstalled, or a new version is installed. So it is enough, to use a view GRF files and print them out. At the moment, the video renderer must be removed, before a Graphedit file can be loaded into the show program successfully.
The show needs some seconds, to load and start a GRF file.
The Show is not capable of loading Infinite Pin Tee filters, if more than one output pin is connected. Branching out is not supported.
Writing to a file is not supported.
The show loads simple GRF Files according to the examples below.
It could be tried, to replace the file sources by a TV Tuner capture filter.
The debug option 'video' has now a value Load: The time, needed for one frame, divided by the timer delay in the More dialog.



1. MaxWidth, MaxHeight

Set these values to the maximum source width and height, that a video will have. These values ensure, that there is always enough video memory for one picture of the video. These values are used for background to window and background to object, when a video plays. Input of power of two values seems to increase the performance.

MaxWidth, MaxHeight too big :	Performance and video memory is wasted.
MaxWidth, MaxHeight too small :	If enough free video memory, a big enough surface is allocated and the video is played, else the video clip is skipped. A message is written to Show_Err.txt.

2. Audio volume

The scale is logarithmic.

3. Audio stream on !

Disabling the audio stream saves some performance bandwidth. This may be useful with a 340*240 video on a 1200MHz system or for any reason. This button is disabled, when a video stream or audio stream is started. When GRF files are loaded, this button is not in operation.

4. Framerate from video !

Sets the framerate in the "More" dialog to the framerate of the video source and ensures, that frames are fetched from the video stream with the selected framerate.

Advantage : The quality is improved.
Disadvantage : Needs more performance bandwidth

5. TempDir / RamDisk

Select a temporary directory to which the videos are copied, before they are played.

6. Save FilterGraph to file

Used, to verify which filters are loaded. Filename: **ShowFG.GRF** in the working directory of show.exe. The file is saved, when the video is loaded. The show has added a CLSID_AMMultiMediaStream object. A filtergraph becomes invalid, when the sourcefile is deleted (Ramdisk).

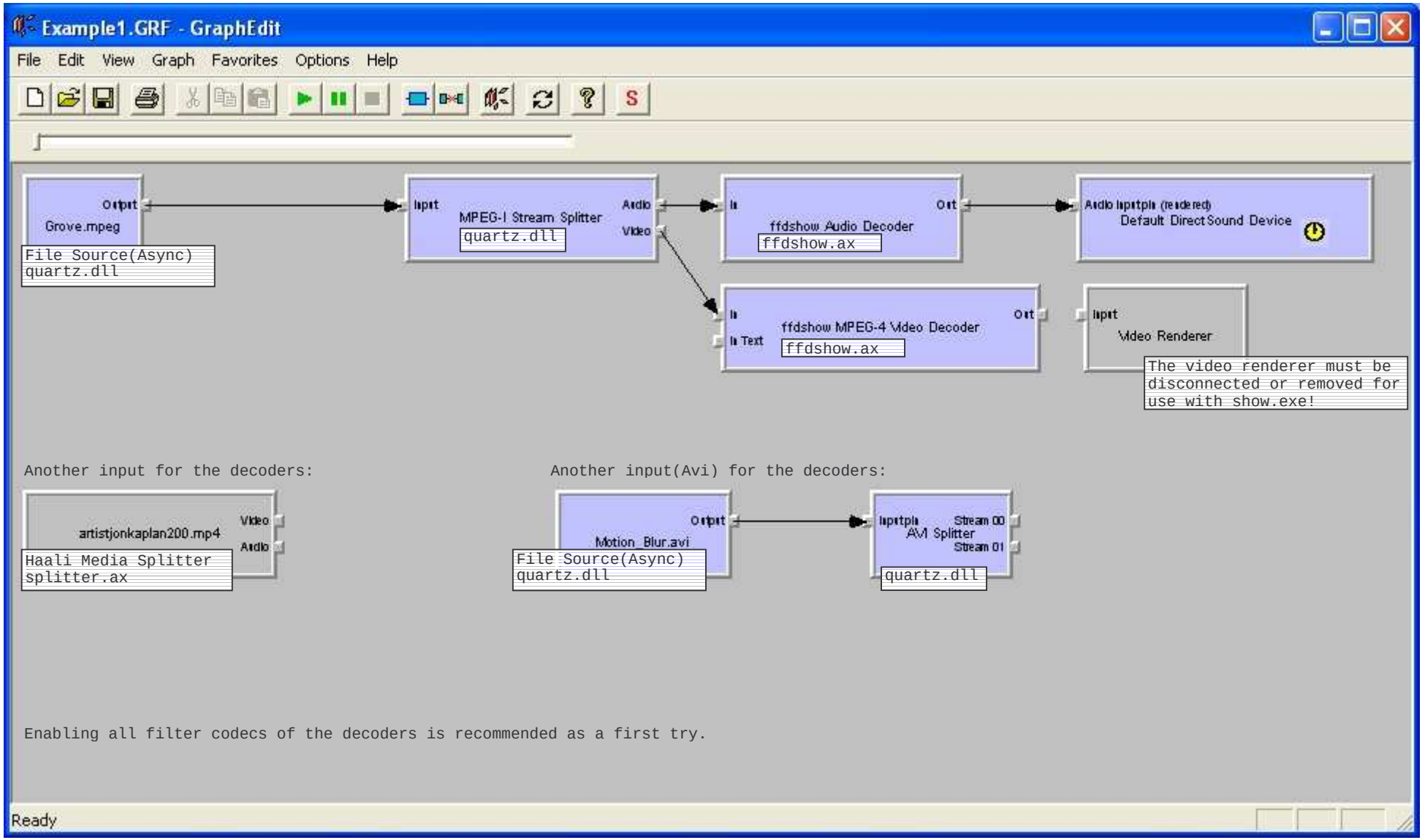
7. Time offset

Used, to start a video not at the beginning, but the selected minutes and seconds later. Applies only to the next video. The value is resetted to zero, when the program terminates. The time offset is ignored, if the remaining playtime is less than 20 seconds.

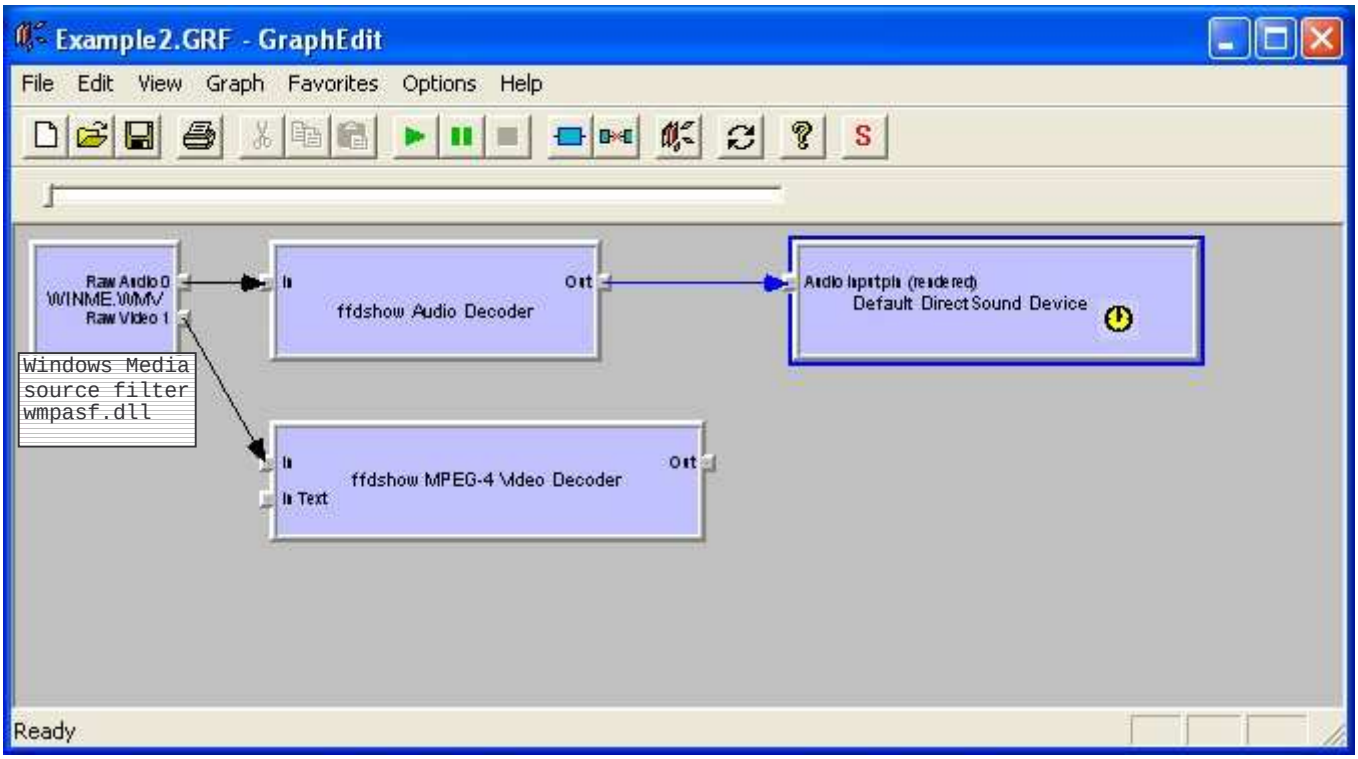
8. Pause
Pause On/Off may take a moment due to Stop/Run the filtergraph, queuing …

Loading GRF Files:

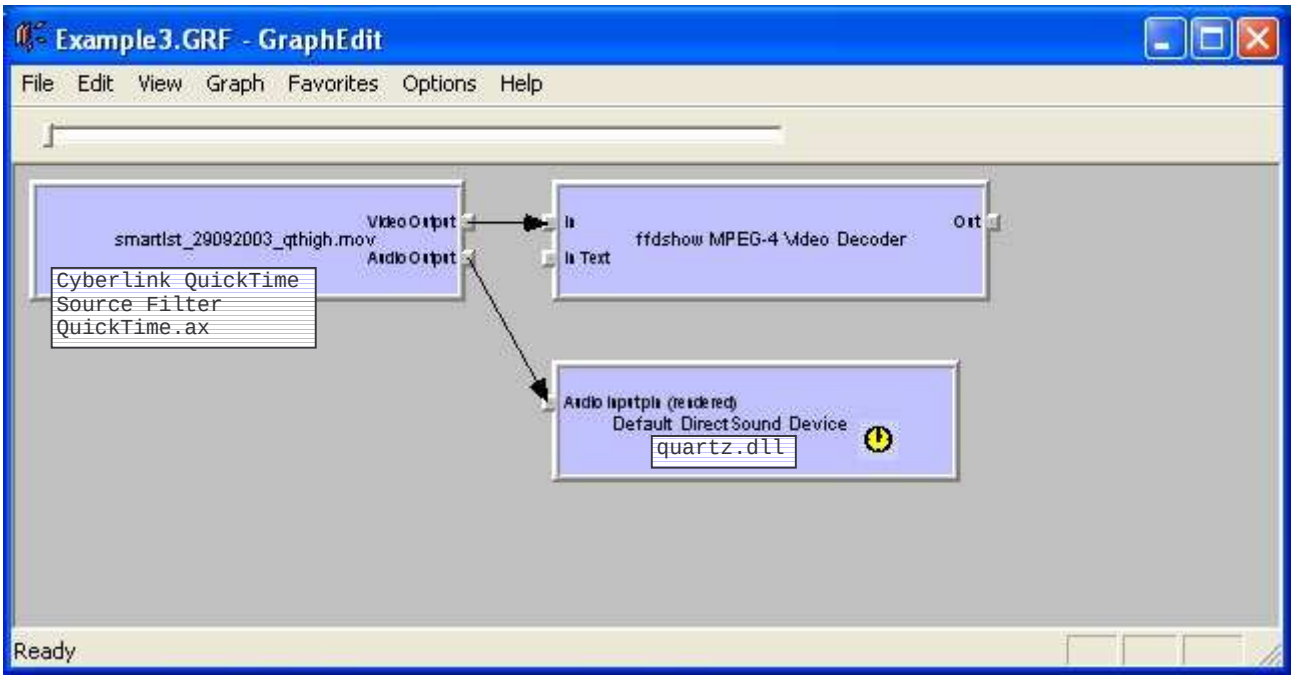
Example1 for GRF files:



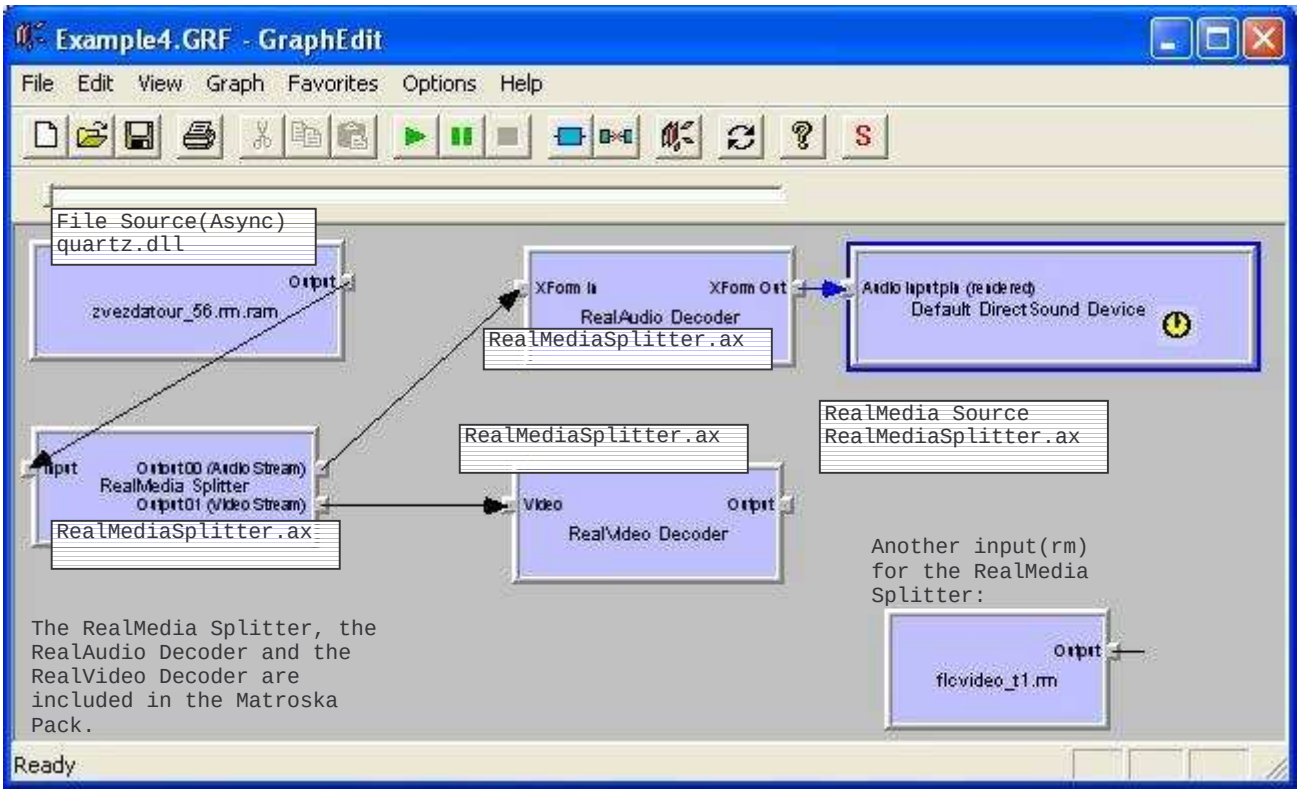
Example2 for GRF files:



Example3 for GRF files:



Example4 for GRF files:



Video “Two”

The filters property pages are not clipped to
the screen of the dialogs. If they are moved outside,
they will be visible on the second screen.

The show enumerates a maximum of 32 filters in a filter graph. To get the filter name, the QueryVendorInfo method is used, which is optional; filters are not required to support it. Most filters support QueryVendorInfo. The program doesn’t search the registry, to get the filter name.
Filters, which don’t support property pages, are not listed.

Buffers for video frames :

Four Surfaces async update: The video frames are delayed three* (Milliseconds per frame setting in the More Dialog).
The filtergraph is less responsive on each command (Run, Pause,Seek, Windowsizing…), because of queuing the frames.
The objects are not delayed.

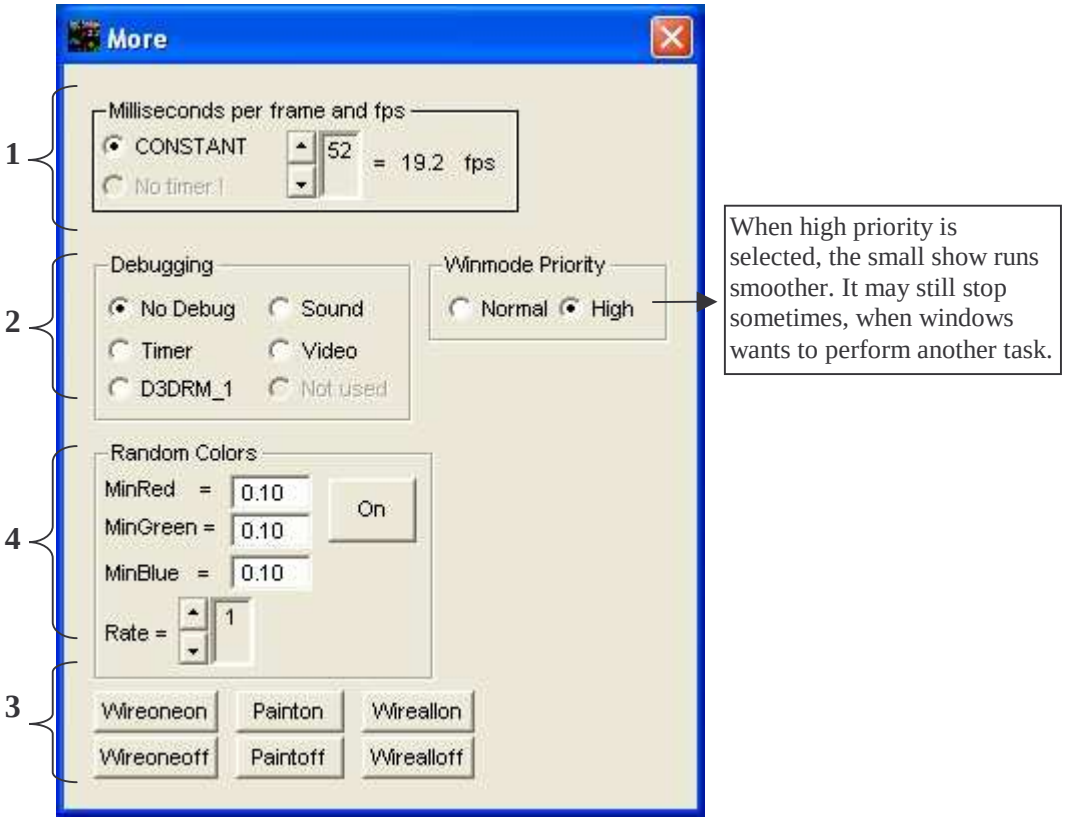
More Dialog ”Debug video”:

CDone (CompletionStatus done) are the milliseconds, the Show has waited, for time difference to the next frame; should be zero; Adjust the number of buffers for video frames.

If dwDropped counts, the system is overloaded. The delay are the milliseconds, the video frames are delayed to the audio stream.

“1 Surface sync update” was the update mode in versions older than The Show 2.7.

More



Select the framerate, by selecting the time from one frameto the next. In the picture above, the framerate is
fps = 1 / 51mSec = 19,6 frames per second.

The program allows to set frame rates from 10,0 fps (100 mSec) to 100,0 fps (10 mSec). The displays for fps in the debug options show the real frame rate, calculated each time from the time between two frames.

1. Milliseconds per frame
If No timer is selected, the program renders, blits, samples and runs as fast as possible. User input becomes nearly impossible. This fullspeed option is more or less an option for testing. A fatal error may occur. It is disabled.

2. Debugging
Some values.

3.
Wireallon: The wireframe mode is more difficult to perform for the display card. **Wireallon** and **Wireonon** cannot be switched on together. At the moment, **Wireonon** doesn't show random colors. The object, that is wired, may look good 100% bright.
Painton: The paint mode is more difficult to perform for the display card. Maybe, the program becomes uncontrollable for some seconds and some objects will be deleted.

4. Random colors
MinRed : The red part of the color is randomly choosen between MinRed and 1.0. Mingreen, Minblue the same…
Rate: The change of color happens each Rate frame. If Rate = 3, the color changes each third frame.

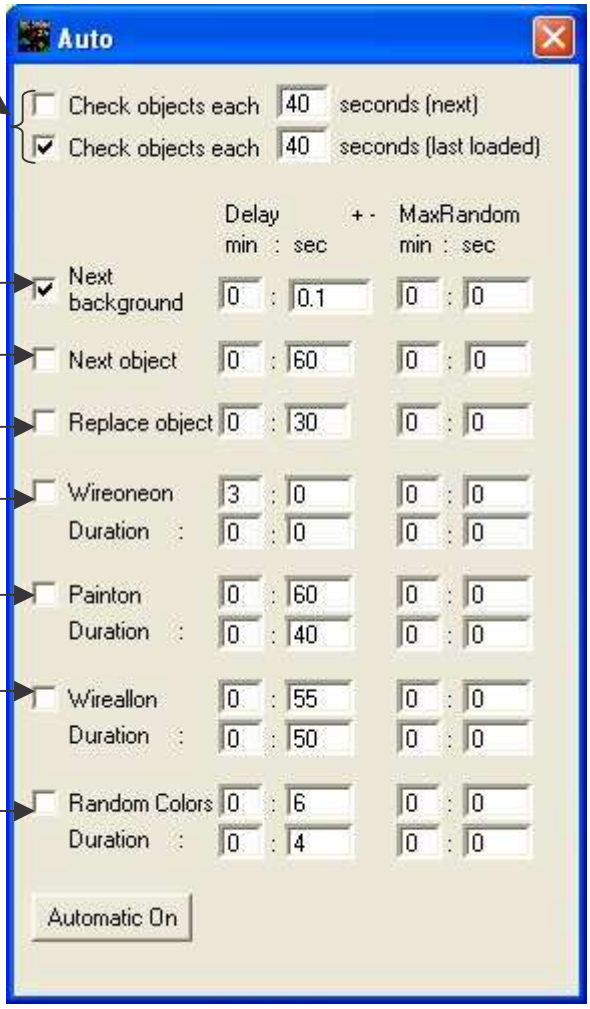
The choose color from the objects dialog and the brightness from sound input and the color from the texture is still valid.

Remark: Choosing color blue (0.0, 0.0, 1.0) in the objects dialog, together with a green (0.0, 1.0, 0.0) texture makes a black object.

Auto

Tries to hold the number of objects constant.
The objects are counted, when the option is clicked on and when auomatic is switched on.
The x-files in the directory should have approximately the same file size.

If a video is playing, there will be no new image or video.

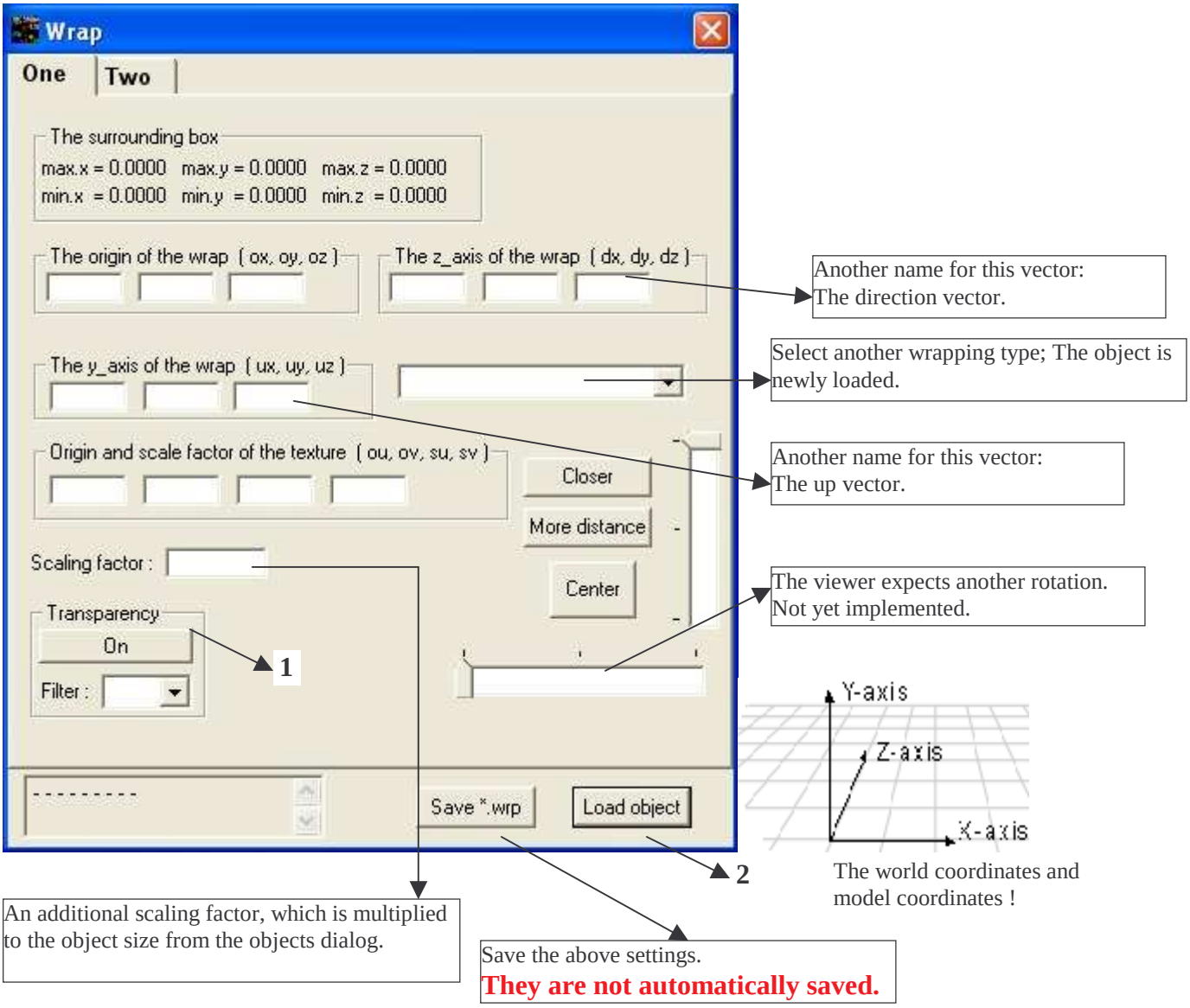


Commands from the automatic are ignored, while a choosecolor dialog (see objects dialog) is open and while a filedialog (see frame dialog) is open.

3. Replace object
Deletes one object, then loads an object; If next object is not selected, the objects in the directory are played one by one. See the More Dialog Debug option D3DRM_1. NrLoaded shows the total number of objects, that have been loaded.

4. Wireoneon
5. Painton
6. Wireallon
7. Random Colors

Wrap “One”



1. Black transparent
Black for JPG : Some pictures or textures may not contain the color (R,G,B,) = (0,0,0), but (0,2,0), (1,0,0) . Each value higher than the selected byte in the combo box ‘Black for JPG’, is set to 0xff. 0xff is here black. The selected byte is applied to red, green and blue channel. The alpha channel is not used. This option defines somehow a colorspace. This option reduces the number of colors, if a value different from 0xff is selected.
When 0xff is selected, no filtering is proceeded (the fastest way). However, the program is capable of processing this filter …
See object **`sphereT.x`** in X_stars.zip as example.
The option ‘Black’ is implemented for jpeg’s and 24 bit top down bitmaps. Minimum size is 20*20 pixels.

2. Load object
With Full screen, the right mouse button (help function) is disabled !

The textures must have a width and height in pixels, which is a power of 2.
Wrapping around a cylinder (cylinder.x) seems to need a squared texture additionally. No matter, which wrapping type is used.

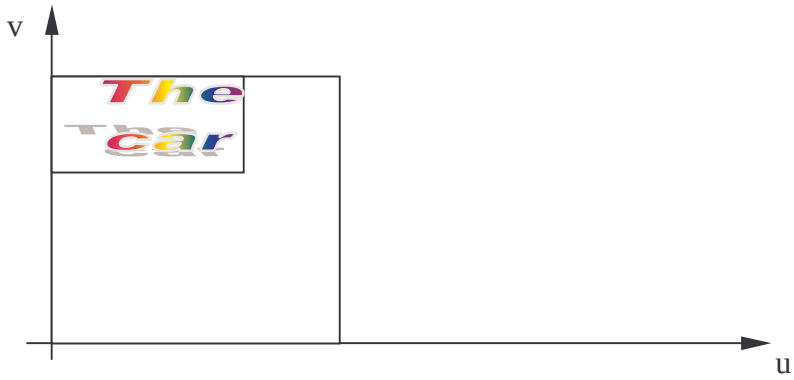
1. Example D3DRMWRAP_FLAT boxasplane_plant.x boxasplane_plant.bmp

```
ox = min.x      oy = min.y       oz = 0.0
dx = 0.0        dy = 0.0         dz = 1.0
ux = 0.0        uy = 1.0         uz = 1.0
ou = 0.0        ov = 0.0         su = 1.0      sv = -1.0
```

sv = 1.0 shows the picture upside down.
ou is the horisontal offset in the texture 0.0 … ou … 1.0 0.0 is pixel zero and 1.0 relates to the width in pixels
ov is the vertical offset in the texture 0.0 … ov … 1.0 0.0 is pixel zero and 1.0 relates to the height in pixels
sv = 2.0 wraps the picture around 2 times **vertically**.
su = 2.0 wraps the picture around 2 times horizontal.

2. Example D3DRMWRAP_FLAT plane_car.x plane_car.bmp
Place the car in a picture, which sides are a power of two: 175, 131 à 256, 256
Make a x-file, which has the same ratio as the pane_car.bmp : ratio = 131 / 175 = 0,74857 with any program.

When loading the texture, a coordinate system for the texture is defined, in which the height and the width of the texture are set to 1.0, no matter, what is the width and height in pixels.



The vertical offset in the texture: 1.0 – (131 / 256) = ov = 0.4883
The scaling factors : 175 / 256 = 0.68359

```
ox = min.x      oy = min.y       oz = 0.0
dx = 0.0        dy = 0.0         dz = 1.0
ux = 0.0        uy = 1.0         uz = 1.0
ou = 0.0        ov = 0.4883      su = 0.6836     sv = -0.6836
```

In this example, the x-file has the normals at the backside set to zero, so the backside is transparent.
If the xy ratio of the whole texture is not 1.0 (here: 256*256), it must be taken into account maually.

Example D3DRMWRAP_SPHERE earth.x earth.bmp

```
ox = 0.0        oy = 0.0         oz = 0.0
dx = 0.0        dy = 1.0         dz = 0.0
ux = 0.0        uy = 1.0         uz = 1.0
ou = 0.0        ov = 0.0     su = 1.0        sv = 1.0
```

The origin of the texture should be in the middle of mass of the object. Is this always (0.0, 0.0, 0.0) ?
D3DRMWRAP_SPHERE is a common wrapmode for complicated objects.

Example D3DRMWRAP_CHROME
Not used in this program.. The reference frame is not oriented relative to the camera. I did not really understand this wrap type.

Example D3DRMWRAP_CYLINDER cylinder.x cylinder.bmp

```
ox = 0.0        oy = min.y     oz = 0.0
dx = 0.0        dy = 1.0       dz = 0.0
ux = 0.0        uy = 0.0       uz = 1.0
```

```
ou = 0.0      ov = 1.0      su = 1.0
sv = 1.0 / height = 1.0  , to cover the height of object exactly
```
By trying, a value sv = 0.97 was found, to have the right color on the upper side of the cylinder.

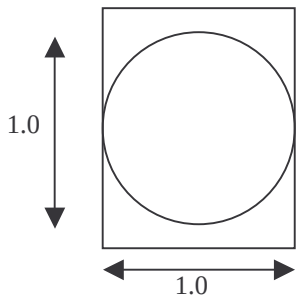
Example D3DRMWRAP_BOX
```
ox =        oy =          oz =
dx =        dy =          dz =
ux =        uy =          uz =
ou =        ov =          su =        sv =
```
I've not yet found an example or a description.

Example D3DRMWRAP_SHEET
```
ox =        oy =          oz =
dx =        dy =          dz =
ux =        uy =          uz =
ou =        ov =          su =        sv =
I've not yet found an example or a description.
```

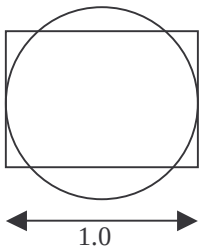
UV_FROM_OBJECT
Takes the texture coordinates from the object.
Many files *.x contain uv coordinates. Some x-files ,often professionally made x-files, which come along with a texture, have these texture coorninates properly set. The origin, z_axis, y_axis and origin and scale factor of the texture are not used, if UV_FROM_OBJECT is selected.

The scaling factor :
The bigger the scaling factor, the bigger the object, when it is loaded in the show.
An aid : Load a sphere with a scaling factor of 1.0 in the wrap dialog. Then measure the size with a lineal. Then load the desired object in the wrap dialog. Then change the scaling factor, until the object has the desired size relative to the sphere.

When the objects are loaded, they are first of all scaled to an x-size of 1.0 ! So, if the object has an xy-ratio != 1.0, the objects will appear with different sizes, if loaded as portrait or landscape.

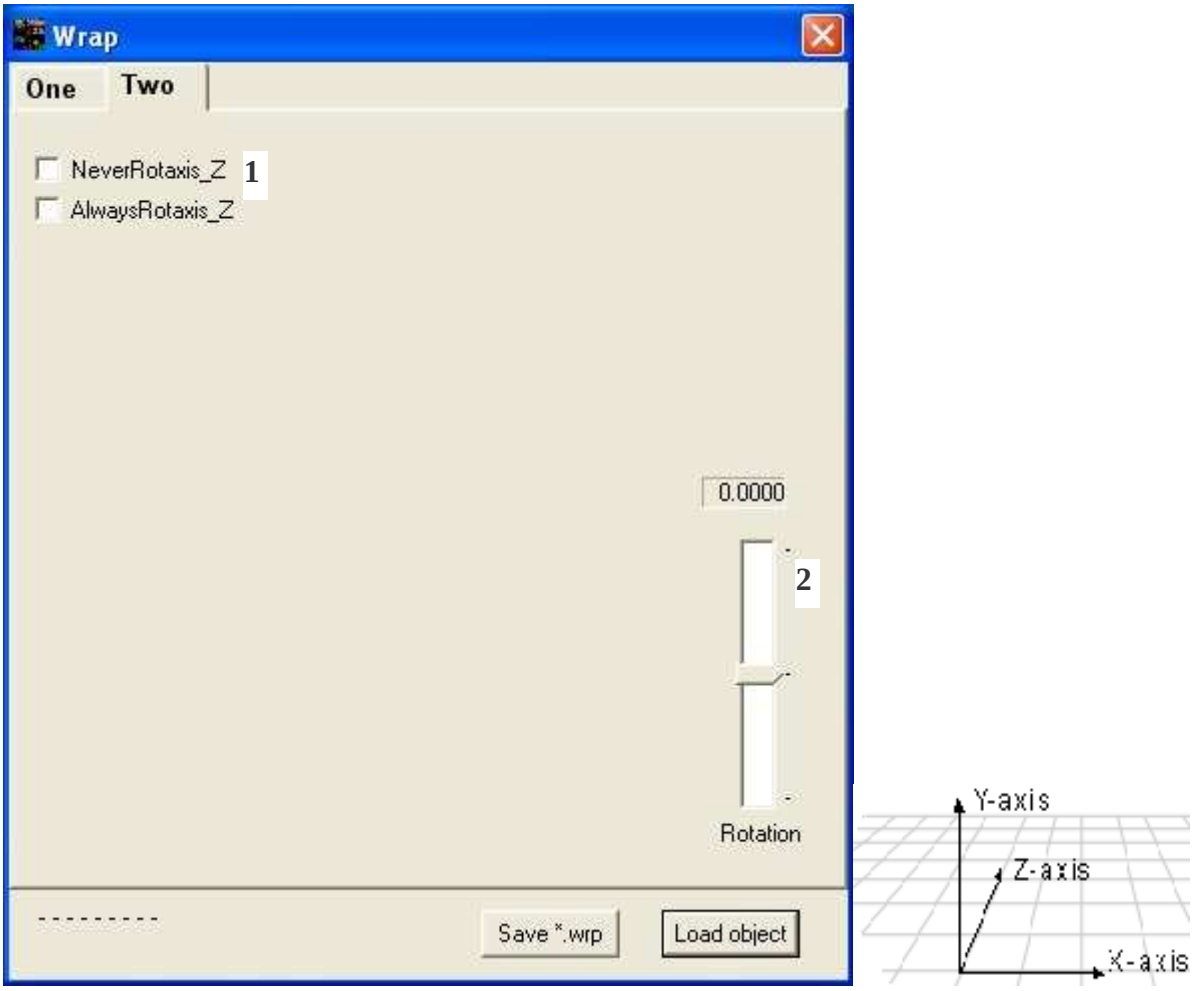


The sheet on the right appears too small :



The scale factor can be used to adjust this.
Many sheets have an aspect ratio of sqrt (2) = 1,41. This will be then a proper scaling factor for the ladscape object, to make it as big as the portrait object.
The other way round: Use a scaling factor of 1 / 1,41 to make the portrait object as big as the landscape object.

Wrap “Two”



1. NeverRotaxis_Z, AlwaysRotaxis_Z
NeverRotaxis_Z : Set this, if the object doesn’t look nice, if it rotates around the z-axis !
AlwaysRotaxis_Z : An additional option …
The trackbar for the rotation axis objects “two” is locked, although some objects with **NeverRotaxis_Z** may be loaded.

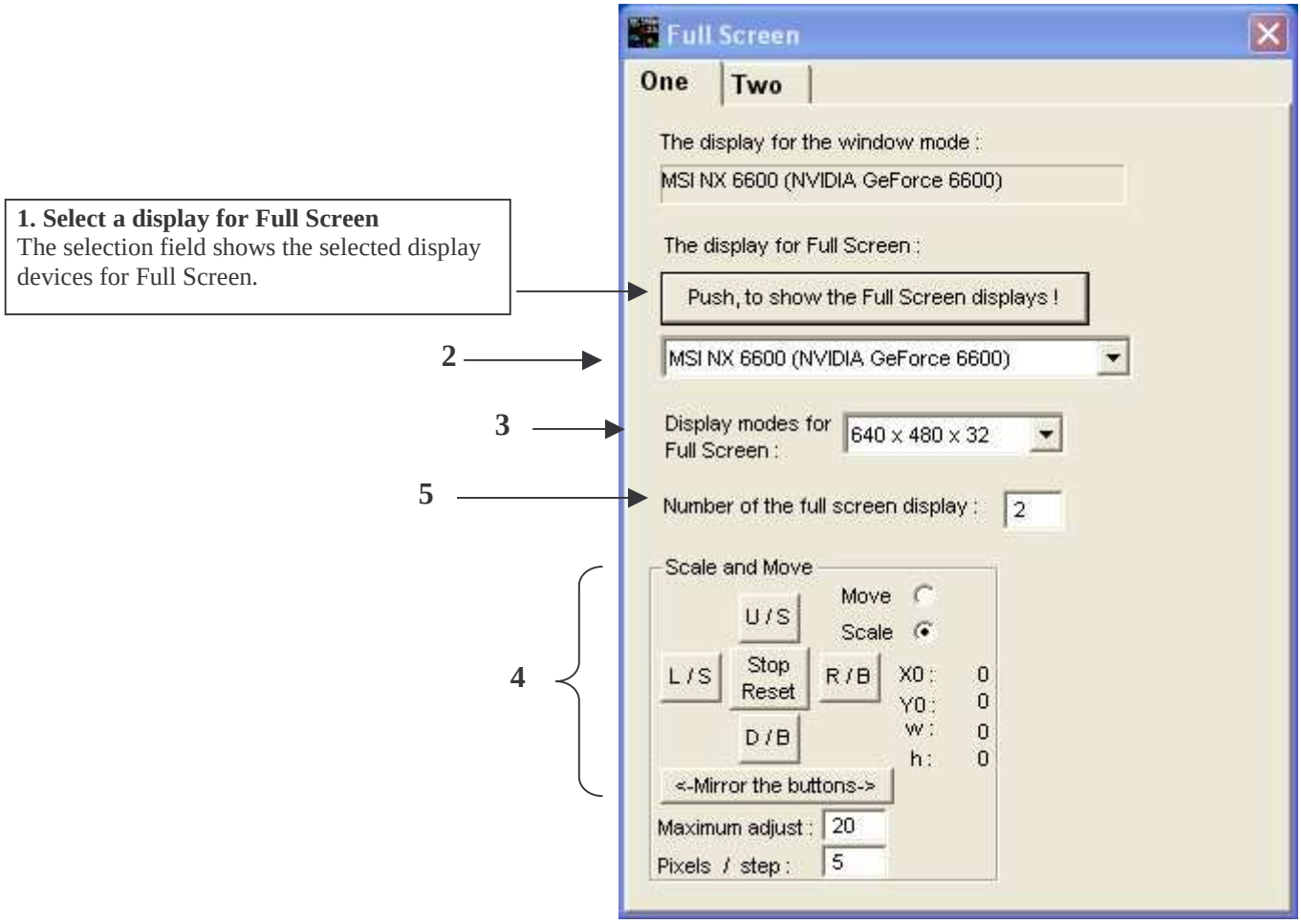
2. Rotation
Sets the rotation of the object. The rotation axis is the z-axis in the world coordinate system. When a new wrap object is loaded, the rotation is set to zero. The rotation is not saved to the wrap file, it is an additional.

Full Screen Device “One”

To use the display card in the AGP slot for the full screen should result in a better performance. It is possible too, to run the full screen with the display card, which is plugged in a PCI slot.

When the full screen is started, the process priority class is set to a very high value.
Advantage: Better performance
Disadvantage: If there are tasks running in the background, they will be rarely performed.

If more than one display card is attached to the screen, with the monitors switched on, moving black lines may appear (interferences). It could be an improvement, to set the monitor frequencies to different values.
It could be a good idea too, to plug in the two displaycards in slots far away from each other, even if todays display cards are well screened.



2. The display for the full screen
Pushing this button starts a function, to find the displays which can be used to play the full screen. The function runs about one or two seconds. Then a drop down list is shown with some items. The first two enties in the list refer to the same display, once enumerated as the windows primary screen and once with it’s GUID. Aligning the displays vertically is not supported.
The file startini.123 can be deleted, to retrieve invisible dialogs.
The data, which is found during the search for display cards is not saved to a file. Only the selection is saved.
The ESC button stops the full screen mode.

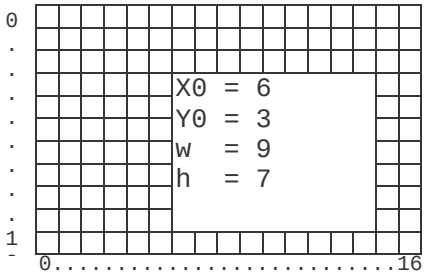
If the full screen mode is to run on a second display, this second display must be attached to the desktop.

3. Display modes for full screen
There is no guarantee, that the monitor can perform the selected display mode. When Full screen is started, the display mode is only changed, if these settings differ from the windows settings. Changing the display mode may cause a flicker on the screen, when full screen is started.

4.Scale and Move
Implemented for full screen and windowed mode.
The implementation for windowed mode is more or less to get familiar with the controls. Each window sizing, each new background image and each restart of windowed mode resets the scale and move values for the windowed mode.
When scaling and moving the fullscreen screen, the unused areas of the screen are set to black. The settings are saved, when the program is terminated.
Moving and scaling stops automatically, if the maximum value is reached, or a timeout is detected.
Pixels / step : big value → scale and move faster; small value → scale and move slower

If the paintmode is selected, the repeating, initialized with a double click, will be quite slow, because moving and scaling is then more difficult to perform.

Example:
Width of the screen = 17 pixels
Height of the screen = 11 pixels
(A small screen)
w, h : The width and height of the used area of the screen.



Mirror buttons : Mirrors the function of the left and right buttons.

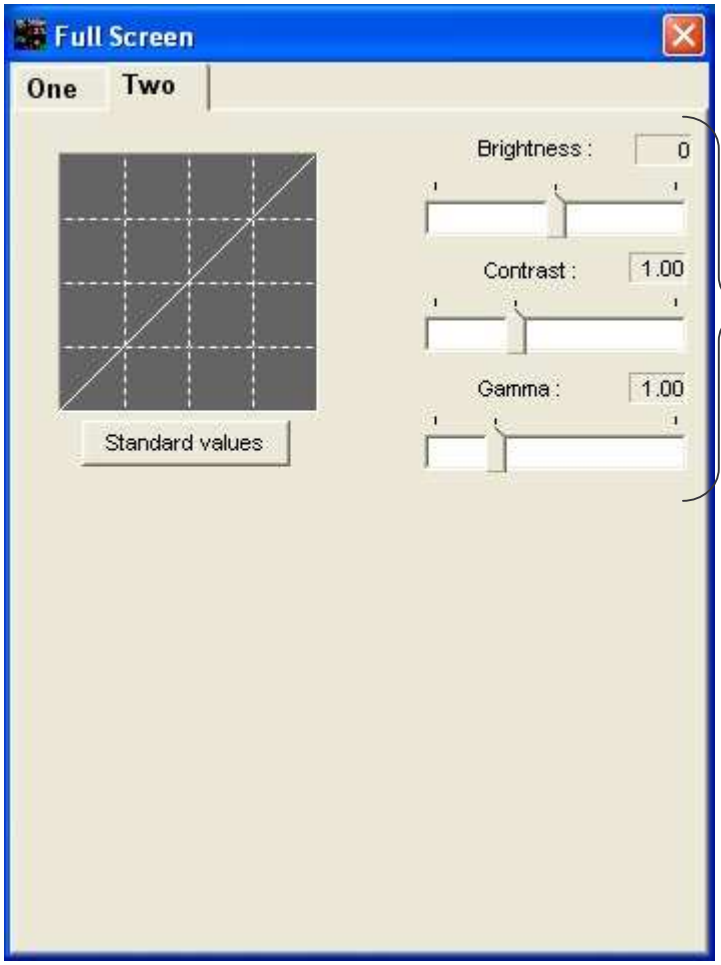
With a screen size of 800*600 and a maximized window with window mode, the program may show w = 800. This value is retrieved by the GetClientRect function. This could be an error. Where is the window border ? For all other windows, and with full screen, the displays should be okay.

5. The display number for the full screen
The number of the display, as it is shown in the windows display settings. The number in this edit field is used, if an error with the gamma ramp initialisation occurs at start of full screen. The message

```
“hcb.hdc == NULL in Set_Hue_Color_Brightness_Init() !”
```

shows, that the wrong number for the display was setted.

Full Screen Device “Two”



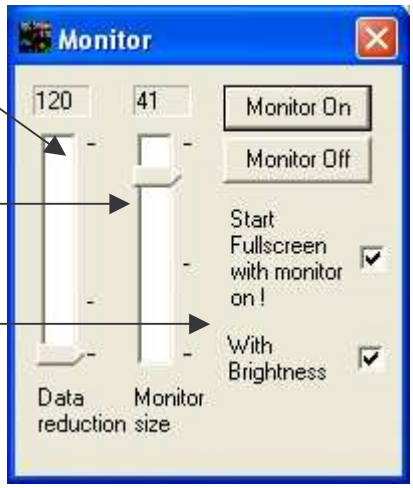
1. The Gammaramp (Brightness, Contrast, Gamma)
Affects the mapping between color values in the frame buffer and DAC values. Affects always the whole monitor. The windows system settings are not changed and ignored.
With two screens: The monitor function copies the screen before the gamma ramp is applied. So gamma changes are not automatically visible on the monitor. The gamma changes are implemented for the monitor by software, pixel for pixel, and an additional option in the Monitor dialog.
Brightness is programmed in a way, that dark black (R,G,B,) = (0,0,0) stays black; important for frames.
Brightness is only available for full screen.
Contrast and Gamma can only be changed with two screens and when full screen show is on.

Monitor

Data reduction
High values is high data reduction .
High values make the live easier for the data bus (AGP, PCI,…).

Monitor size
Small values make small pictures.
The performance is rarely effected.

With Gamma
Applies Brightness for the Monitor. Needs more time, because brightness is applied to each pixel by software. See **m.s1_res** in the debug option Debug_D3DRM in the more dialog, to see how long it takes, to perform the monitor option.



Each frame of the fullscreen is shown. Recommended: Set both screens to **32 bits per pixel**. Do not chick around with 16 bit.
The monitor function for window mode is not implemented.
See the "More" dialog, debug option D3DRM_1. The value **`m.s1_res`** shows the calculation time in milliseconds for one monitor picture.

Change the data reduction slider and watch the time **`m.s1_res`**.
On my system, m.s1_res is approximately 15 msec to 20 msec.

A system with Riva TNT2 AGP, Riva TNT2 PCI ,Pentium 4 1200 MHz and a display card setting 640*480*32 is the minimum PC power for this Monitor option.

The monitor switches off automatically (that's often the case !), if a timeout is detected. **To be switched on again …**

If the monitor shows the loading new background images not smooth, there is not enough time, to send the data from the full screen display card to the other one. The full screen picture should appear smooth. This is not a severe error.
To see the monitor image smooth, set the Timetick value in the Background dialog to a higher value.

Using the monitor option like a screenshot should be possible anyway...

Errors, Remarks, Keyboard input

Keyboard input : **(Click first to main window)**

M next object	
Y Load object **Please don't use the repeat function of the keyboard !!**	X Unload object
N Next background	B One background back
Q More speed for the objects	A Less speed for the objects
W Objects bigger	S Objects smaller
E rotation faster	D rotation slower
R More input amplification	F Less input amplification
T random colors on	G random colors off
Z PaintOn	H PaintOff
U Wireallon	J Wirealloff
I next frame	K one frame back
O Automatic on	L Automatic off
C Scale mode	V Move mode
P Pause On / Off	
1 Less brightness	Shift + 1 More brightness
2 Less highness for objects **exponential !**	Shift + 2 Objects more highness **exponential !**
3 Rotate objects counter clockwise **exponential !**	Shift + 3 rotate objects clockwise **exponential !**
4 World height of the objects smaller **exponential !**	Shift + 4 World height of the objects bigger **exponential !**
5 Size of the background object smaller	Shift + 5 Size of the background object bigger
6 Background object keystone	Shift + 6 Background object keystone
7 Video audiostream less volume	Shift + 7 Video audiostream more volume
8 Restore framerate	

The digital audio output on the sound card is very high dynamic, as each digital signal is. There is no warranty for the loud speaker.

Sizing the window ,while a movie plays, may cause the show to terminate. This error doesn't occur in fullscreen mode.

When the show is in fullscreen mode, and another program opens a window and makes it the foreground window, the show may block. A reboot is necessary. The show is not fully compatible. Scheduled programs, scanners, virus updaters may open such windows.

The program locks the media slots when it is started, and unlocks when it terminates. If the show terminates via error, the slots may stay locked. Starting and closing show.exe unlocks.

Switching on an automatic option, while the the automatic is switched on, does not always work. The values for duration and delay will have values, which were set, when the automatic was switched on. Switching the automatic button off and on additionally sets these time values.

The monitor with the number 1 in the display panel of windows should be set as the primary display. See the display panel in the windows system settings.

The objects are always set to white, before the texture is applied. Coloring an object by vertices or faces is not supported in this version. Not each object has nice colored vertices or faces.

Setting the brightness,contrast and gamma fails on many PC systems, for values "very bright" and "very dark". Many PC systems don't support screens which are dark black or bright white and have good reasons to do so.

Too many objects, loaded at show start may overload the system, depending on the size of the x-files and the system. The program becomes uncontrollable.

If the program becomes uncontrollable, press **Ctrl + Alt + Del** and stop the task “show.exe” manually. Maybe, it is necessary, to repeat. Maybe it is necessary to reboot.

2 Displaycards :
Show on second screen, Dialogs on primary screen : Okay
Show on primary screen, Dialogs on second screen screen : The dialogs have a wrong position, if the width and height of the “Show screen” are not the same as the settings in the operating system (Windows®).

Each application defined error message is written to the file Show_Err.txt. It is allowed, to delete this file.

Small fonts (96 dpi) should be selected in the windows display settings, to display the dialogs nice …

Energy options:

When the timeout interval of an energy option elapses, the operating system sends a message to the window, which has the focus. The windows of the show program throw the SC_SCREENSAVER and SC_MONITORPOWEROFF messages away. The messages are not passed back to the operating system, for beeing finally proceeded. The SC_MONITORPOWEROFF message includes “ monitor shut off ” and “ monitor going to low power ”.

Window mode : It is the users choice, weather a show window has the focus, or another program ...
Full screen : The program has always the focus, exceptionally another program catches the focus, while full screen is on, what should rarely happen.

Harddiscoff, stand by, sleep mode and any other energy option is not explicitely managed by the program.
When an energy option is finally proceeded, a fatal error may occur, depending on the state of the program.
To do very save, switch off these screensaver, poweroff and energy options, when running the show for a long time.