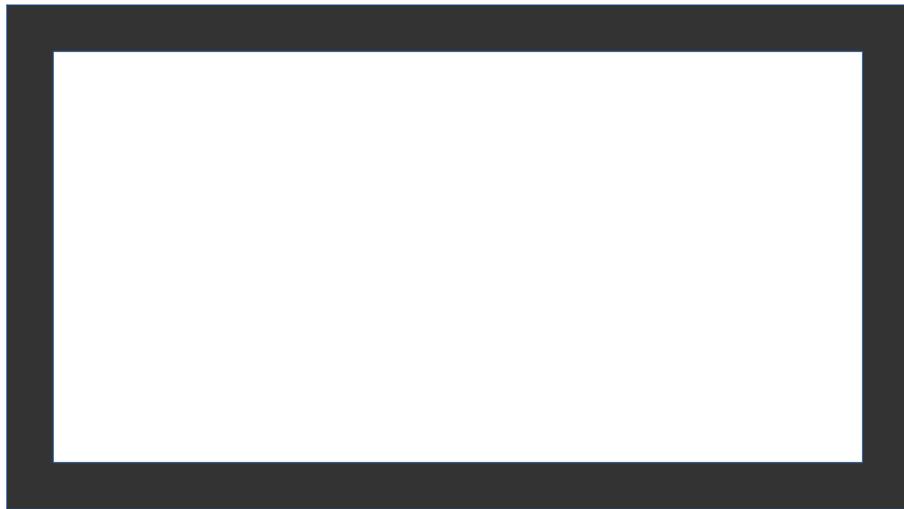


GIMP-GAP Documentation

Project gap "GIMP Animation Package"
10. Oct. 2011 (pre)release 2.6.1



Notes

This document is a compilation of all the [GNOME/gimp-gap](#) / docs available at GitHub. The text and layout has been lightly edited with all preformatted text as is from the manpages of the original author(s).

The order of the GIMP-GAP plug-ins section is based on the Internal PDB and menu names table.

Your version of GAP may not have the latest updates which include [New] plug-ins (some noted), from the development version.

--

Odinbc ([Gimp Chat](#))

2016-06-04

Quick Contents

Introduction.....	5
How to use.....	8
Internal PDB and menu names of GAP plug-ins.....	11
GIMP-GAP plug-ins.....	14
STORYBOARD_FILES.....	96
How To.....	121
Video Configuration (gimprc).....	127
About GIMP-GAP.....	134

Table of Contents

Introduction.....	5
Idea.....	5
General.....	5
Input from videofile:.....	6
Output to videofile:.....	6
The Storyboard:.....	7
Morphing.....	7
Onionskin Layers.....	8
How to use.....	8
Creating multiple frames (video frames).....	8
Navigation (Goto).....	8
Main render features.....	9
NO UNDO:.....	9
About Locks.....	9
Internal PDB and menu names of GAP plug-ins.....	11
<Image>/Filters/.....	11
<Image>/Video/.....	11
<Image>/Video/Layer/.....	12
<Image>/Video/Encode/.....	12
<Image>/Video/Go To/.....	12
<Image>/Video/Onionskin/.....	12
<Image>/Video/Split Video into Frames/.....	12
<Toolbox>/Xtns/Split Video into Frames/.....	13
<Image>->Script-Fu->Animators-> **.....	13
NON_INTERACTIVE batchscript.....	13
NON_INTERACTIVE video encoder.....	13
GIMP-GAP plug-ins.....	14
Filters/Filter all Layers.....	14
Script-Fu/Animators/Sel To AnimImage.....	14
Filters/Filtermacro.....	16
Video/Bluebox.....	18
Video/Frames Crop.....	19
Video/Frames Resize.....	20
Video/Frames Scale.....	21
Video/Delete Frames.....	21
Video/Frames Density.....	22
Video/Duplicate Frames.....	23
Video/Exchange Frame.....	24
Video/Filename to Layer.....	24
Video/Frames Modify.....	25
Video/Morph.....	29

Video/Move Path.....	29
Video/VCR Navigator.....	39
Video/Frames Convert.....	42
Video/Frames Flatten.....	43
Video/Frames LayerDel.....	43
Video/Frames to Image.....	43
Video/Frames Renumber.....	45
Video/Frame Sequence Reverse.....	46
Video/Frame Sequence Shift.....	47
Video/Split Img to Frames.....	48
Video/Storyboard.....	49
Master Properties (Storyboard).....	53
Clip Properties (Storyboard).....	54
Transition Attributes (Storyboard).....	58
Generate original tone audiotrack (Storyboard).....	59
Storyboard Player.....	60
Video/Playback.....	60
Video/Layer/Attributes/.....	66
[NEW] Plug-in-layer-set-alpha-by-colormask.....	66
Plug-in-foreground-extract-matting.....	66
Video/Encode/Master Videoencoder.....	68
Video/Encode/MPEG1 (mpeg_encode).....	72
Video/Encode/MPEG2.....	74
Video/Go To.....	76
Video/Onionskin/Configuration.....	76
Split Video into Frames/MPlayer based extract.....	80
Split Video into Frames/XANIM based extract.....	82
Split Video into Frames/Extract Videorange.....	83
Video/Layer/Colors/Levels.....	86
Video/Layer/Colors/CurvesFile.....	87
Layer/Colors/Hue-Saturation.....	88
Videoindex creation.....	88
[New] Video/Layer/Render/Fire Pattern.....	89
[New] Video/Layer/Render/Water Pattern.....	93
[New] Layer/Transparency/Foreground Extract Via Selection.....	95
STORYBOARD_FILES.....	96
General.....	96
Syntax:.....	97
Overview of supported RECORD TYPES:.....	97
RECORD Types Details:.....	99
Filenames:.....	116
Videotracks:.....	116
Sections:.....	117
Framerange:.....	118
Audiorange:.....	118

Macrofiles:.....	118
Apply Macrofiles with varying parameters:.....	118
Macrofiles for global usage:.....	120
What are Macrofiles and how to create them ?.....	120
Supported video input.....	120
LIMITS:.....	121
How To.....	121
HOW TO-do-lossless-MPEG-cut.....	121
a) lossless MPEG1 / MPEG2 cut.....	122
b) lossless MJPG cut.....	122
How to make a GIMP Plug-In an "Animated" one.....	123
Video Configuration (gimprc).....	127
About GIMP-GAP.....	134
MAINTAINERS.....	134
NEWS.....	135
README.....	142
XchangeLog.....	145

Introduction

GIMP is a great program for creating and manipulating pixel based images of many types.

The plug-in concept and the procedural database allows programmers to extend the GIMP's functions in many ways.

In GIMP there are some plug-in's supporting animation features, based on layers where each layer of the image is considered as one frame of the animation.

GIMP-GAP is a collection of plug-ins that extends GIMP's animation capabilities by supporting the creation of more complex animations.

Idea

With the help of the GIMP-GAP plug-in collection GIMP can operate on a series of images as if they were a single one.

Each frame of an animation can have multiple layers. The user can step from frame to frame by pressing an accelerator key or can do image manipulations that are propagated automatically to a selected range of frames.

Layer animated images can be combined with frames (multiple images) as you can see in the "Move Path" plug-in or converted to frames and vice-versa.

General

An animation consists of a series of images of the same size & type (frames).

The concept of GIMP-GAP is to store each frame separate on disk, using a filename convention that includes the frame number and extension.

Example: film_000001.xcf
 film_000002.xcf
 film_000012.xcf

GIMP's xcf file format should be used to store the frames, so that each frame can have more layers.

Sound is not supported in this concept, but will be handled as sperate audiofiles at encoding time or in the playback module.

The playback framerate and other common information about the animation are stored in an extra videoinfo textfile film_vin.gap

This requires a lot of diskspace but offers much more flexibility while working on the animation.

To save diskspace you may optional use gzipped xcf frames (takes extra time to un/compress) by using the extension.xcfgz (gzip has to be installed on your system to do that). If you can accept lossy compression, you may also use the xjt fileformat to store your frames on jpeg based compression. The curent implementation of the xjt fileformat is restricted to UNIX operating systems.

Input from videofile:

GIMP-GAP has features to access both audio and frames in already encoded videofiles.

One way to deal with such videofiles is to extract the frames. Each frame can be stored as numbered frame image on disk in that case. GIMP-GAP has a built in video extracting plug-in (based on ffmpeg and libmegg3).

refer to ([plug-in-gap-extract-video.txt](#))

On UNIX platforms GIMP-GAP also provides frontends for the videoplayers,

- mplayer
- xanim

those players are capable to extract frames from common videofile formats (avi, mpeg, quicktime,).

Please note that xanim is old unmaintained software.

Input from videofile:

The frontend for xanim is still there and may even work if you still have xanim exporting edition. But mplayer is the better choice.

refer to ([plug-in-gap-mplayer-decode.txt](#))

refer to ([plug-in-gap-xanim-decode.txt](#))

The other method can read the frames from one or more videofiles and pass them to one of the GIMP-GAP video encoders without the need to store the frames separately.

This method requires a storyboard file.

The storyboard file describes how to assemble videoclips, images and audiofiles to one resulting output video.

Output to videofile:

GIMP-GAP provides video encoder plug-ins to save a series of input frame images or videoclips as videofile.

This process is called encoding and usually does compresses the processed frames and audiodata.

GIMP-GAP encoder plug-ins can operate on a series of numbered frame images or storyboard files as input (encoding on the fly).

Another option is to convert the frames to one multilayered image that can be saved as MNG animation, animated-GIF (or, in the future in other animation formats when other load/save modules were added to GIMP).

GIMP-GAP provides automated file format conversions for the frames of an animation.

These conversions offer a way to use external video encoder programs.

All GIMP-supported image file formats may be used.

On UNIX Platforms GIMP-GAP provides frontend dialog interfaces for external software MPEG video encoder programs.

(mpeg_encode and mpeg2encode)

This method and both encoders are old.

External encoders can NOT directly encode storyboard files, and typically want input as series of single frames in PPM format.

This need much disk space and extra time for converting frames to PPM.

refer to ([plug-in-gap-mpeg-encode.txt](#))

refer to ([plug-in-gap-mpeg2encode.txt](#))

The Storyboard:

Storyboards are useful for the creation and cutting of longer videos. GIMP-GAP provides the storyboard dialog to create and edit simple storyboard files.

Storyboard files are textfiles and can be compared with playlists for video clips, images and audio clips.

In this release the storyboard dialog is limited to simple storyboard features without sound support.

For encoding videos via storyboard files the sound support is already implemented and there are some more features available.

See docs/reference/txt/[STORYBOARD_FILE_DOC.txt](#) for the complete Syntax description.

Morphing

Morphing can be used to render frames between 2 scenes.

** Morphing is still an experimental feature and not fully documented yet **

refer to ([plug-in-gap-morph.txt](#))

Onionskin Layers

GIMP-GAP does support onionskin layers.

refer to ([plug-in-gap-onionskin-configuration.txt](#))

How to use

Creating multiple frames (video frames)

- from an existing single image

Save your image as XCF file. (<Image>/File/Save as) use a name that ends up in _000001.xcf (or _000001.xcf.gz)

Then duplicate your image (<Image>/Video/Duplicate)

You'll be asked how many copies you need.

(**Note:** all copies are stored on disk immediate without explicit save)

- from an existing layer animated multilayer image use

<Image>/Video/Split Img to frames

This will create frames, a series of images on disk, with a name that ends up in (_000001.xcf).

Optional you may use other extensions. (.xcfz, .jpg ...)

WARNING:

The extension defines the file format of the frames. Most of the other formats (than GIMP's .xcf format) can not save multilayer frames, or frames with alpha channels.

- from outside GIMP

You may rename and copy existing XCF images according to the frame naming conventions.

000001.xcf

000002.xcf ...

Then load (only one of them) into GIMP.

Navigation (Goto)

It is recommended to define some accelerator keys for quick walk through the frames.

Here are my settings (excerpt from my \$HOME/.gimp/menurc)

(menu-path "<Image>/Video/Goto/First Frame" "<control><alt>1")

(menu-path "<Image>/Video/Goto/Prev Frame" "<alt>1")

(menu-path "<Image>/Video/Goto/Next Frame" "<alt>2")

(menu-path "<Image>/Video/Goto/Any Frame" "<alt>3")

(menu-path "<Image>/Video/Goto/Last Frame" "<control><alt>2")

Alternatives:

- You may open the GIMP-GAP videoplayer (menu-path: <Image>/Video/Playback...) and use the player for visual positioning (use the FrameNr Button or the GO-Button array for that task)

- You may open the GIMP-GAP video navigator dialog (menu-path: <Image>/Video/VCR Navigator)

Main render features

GIMP-GAP provides features for rendering moving objects and for automated modifying of multiple frames.

refer to ([plug-in-gap-move-path.txt](#))

refer to ([plug-in-gap-modify.txt](#))

NO UNDO:

There is no Undo for the video frame based GIMP-GAP video plug-ins. If you step to the next frame (menu: <image>/Video/Goto Next) the current frame is saved to disk and the next frame is loaded into the image. All undo steps were cleared at this time.

About Locks

All the GIMP-GAP plug-ins are using a lock.

This lock disables to run other GIMP-GAP plug-ins (or the same plug-in twice) on the same image at the same time, while the current GIMP-GAP plug-in is working on that frame.

GIMP-GAP keeps a locktable (for the current session) that contains the image_id's of all images currently locked by GIMP-GAP plug-ins.

If you cancel a running GIMP-GAP plug-in with the cancel button in the progress window, or kill it from your operating system the lock may remain.

(This should only happen to WINDOWS users, for UNIX users locks are checked for dead process ids)

To unlock you can save the current frame image, close all views to that image, then reload the frame image. (or quit and restart GIMP)

WARNING:

It is not recommended to open more than one frame of an animation at the same time.

(menu: File/Open "img_000001.xcf"

menu: File/Open "img_000002.xcf")

In this constellation both gap plug-ins may concurrent in save/load to/from the same file.

Example:

call from "img_000001.xcf" Video/Delete Frames (range 000001 to 000003)

now

img_000004.xcf is renamed to img_000001.xcf

img_000005.xcf is renamed to img_000002.xcf

then

call from "img_000002.xcf"

Video/Goto First the img_000002.xcf (its old content) is saved, overwiting which was img_000004.xcf before.

If you make the 2nd call while the 1st one is in progress, you may trash your frames (2 writers on one file) and/or crash your GIMP session.

Internal PDB and menu names of GAP plug-ins

<Image>/Filters/

plug_in_gap_layers_run_animfilter	<Image>/Filters/ Filter all Layers
plug_in_filter_macro	<Image>/Filters/ Filtermacro

<Image>/Video/

plug_in_bluebox	<Image>/Video/ Bluebox
plug_in_gap_anim_crop	<Image>/Video/ Frames Crop
plug_in_gap_anim_resize	<Image>/Video/ Frames Resize
plug_in_gap_anim_scale	<Image>/Video/ Frames Scale
plug_in_gap_del	<Image>/Video/ Delete Frames
plug_in_gap_density	<Image>/Video/ Frames Density
plug_in_gap_dup	<Image>/Video/ Duplicate Frames
plug_in_gap_exchg	<Image>/Video/ Exchange Frame
plug_in_gap_modify	<Image>/Video/ Frames Modify
plug-in-gap-morph-layers	<Image>/Video/ Morph
plug_in_gap_morph_tween	<Image>/Video/ Morph Tweenframes
plug_in_gap_morph_one_tween	<Image>/Video/ Morph One Tween
plug_in_gap_morph_workpoints	<Image>/Video/ Morph Workpoint Generator
plug_in_gap_move	<Image>/Video/ Move Path
plug_in_gap_navigator	<Image>/Video/ VCR Navigator
plug_in_gap_range_convert	<Image>/Video/ Frames Convert
plug_in_gap_range_flatten	<Image>/Video/ Frames Flatten
plug_in_gap_range_layer_del	<Image>/Video/ Frames LayerDel

<Image>/Video/

plug_in_gap_range_to_multilayer	<Image>/Video/ Frames to Image
plug_in_gap_renumber	<Image>/Video/ Frames Renumber
plug_in_gap_reverse	<Image>/Video/ Frame Sequence Reverse
plug_in_gap_shift	<Image>/Video/ Frame Sequence Shift
plug_in_gap_split	<Image>/Video/ Split Img to Frames
plug-in-gap-storyboard-edit	<Image>/Video/ Storyboard
plug_in_gap_videoframes_player	<Image>/Video/ Playback

<Image>/Video/Layer/

plug-in-layer-set-alpha-by-colormask	<Image>/Video/Layer/ Attributes/
plug-in-foreground-extract-matting	<Image>/Video/Layer/ Attributes/

<Image>/Video/Encode/

plug-in-gap-vid-encode-master	<Image>/Video/Encode/ Master Videoencoder
plug_in_gap_mpeg2encode	<Image>/Video/Encode/MPEG2 (mpeg2encode)
plug_in_gap_mpeg_encode	<Image>/Video/Encode/MPEG1 (mpeg_encode)

<Image>/Video/[Go To/](#)

plug_in_gap_goto	<Image>/Video/Go To/Any Frame
plug_in_gap_first	<Image>/Video/Go To/First Frame
plug_in_gap_last	<Image>/Video/Go To/Last Frame
plug_in_gap_next	<Image>/Video/Go To/Next Frame
plug_in_gap_prev	<Image>/Video/Go To/Previous Frame

<Image>/Video/Onionskin/

plug_in_gap_onionskin_configuration	<Image>/Video/Onionskin/ Configuration
-------------------------------------	--------------------------------------------------------

<Image>/Video/Split Video into Frames/

<Image>/Video/Split Video into Frames/

plug_in_gap_mplayer_decode	<Image>/Video/Split Video into Frames/ MPlayer based extract
plug_in_gap_xanim_decode	<Image>/Video/Split Video into Frames/ XANIM based extract
plug_in_gap_extract_video	<Image>/Video/Split Video into Frames/ Extract Videorange

<Toolbox>/Xtns/Split Video into Frames/

plug_in_gap_mplayer_decode_toolbox	<Toolbox>/Xtns/Split Video into Frames/ MPlayer based extract
plug_in_gap_xanim_decode_toolbox	<Toolbox>/Xtns/Split Video into Frames/ XANIM based extract
plug_in_gap_extract_video_toolbox	<Toolbox>/Xtns/Split Video into Frames/ Extract Videorange
Plug-in-gap-video-index-creator	<Toolbox>/Xtns/ Videoindex creation

<Image>->Script-Fu->Animators-> **

script-fu-selection-to-anim-image	<Image>->Script-Fu->Animators-> Sel To AnimImage
-----------------------------------	------------------------------------------------------------------

NON_INTERACTIVE batchscript

plug_in_gap_get_animinfo	** NON_INTERACTIVE batchscript API
plug_in_gap_move_path_ext	** NON_INTERACTIVE batchscript API
plug_in_gap_move_path_ext2	** NON_INTERACTIVE batchscript API (using a controlpoint file)
plug_in_gap_set_framerate	** NON_INTERACTIVE batchscript API
plug_in_gap_video_edit_clear	** NON_INTERACTIVE batchscript API
plug_in_gap_video_edit_copy	** NON_INTERACTIVE batchscript API

plug_in_gap_video_edit_paste	** NON_INTERACTIVE batchscript API
------------------------------	------------------------------------

NON_INTERACTIVE video encoder

plug-in-gap-enc-avi1	** NON_INTERACTIVE video encoder
plug-in-gap-enc-ffmpeg	** NON_INTERACTIVE video encoder
plug-in-gap-enc-singleframes	** NON_INTERACTIVE video encoder

GIMP-GAP plug-ins

Filters/Filter all Layers

Script-Fu/Animators/Sel To AnimImage

"gap-filterall-db-browser"

Animated calls of plug-in filters:

Start from Menu:

<Image>/Filters/Filter all Layers

<Image>/Script-Fu/Animators/Sel To AnimImage

First of all, you need a multilayered Image. You can use "<Image>/Video/Frames to Image" to create one from a series of frames, or duplicate the background layer of a single layered image (Press Ctrl-C within the layers dialog window N-times)

Then call "Filter all Layers" from within the multilayer image. You'll get a window similar to the PDB-Browser, that shows all available plug-ins in a listbox (and informations about the selected plug-in on the right side).

Select one of the listed plug-ins, then select acceleration characteristic and press the Apply button to continue with the next dialog step.

Note: The listbox does not show the full PDB, but only those plug-ins that are capable to run under control of the "Filter all Layers" plug-in. Some of them are restricted for apply with a fixed setting on all handled layers. For Those plug-ins the Acceleration Characteristic spinbutton widget is disabled. The other plug-ins are capable to be applied with varying values and enable the spinbutton.

"Acceleration Characteristic"

can be set with a spinbutton where you can enter a value. A graph next to the spinbutton shows the selected characteristic. The acceleration characteristic can also be changed by clicking on the graph and dragging vertically with the mouse.

A straight line from left bottom to right top corner of the graph is drawn for value 1 that represents constant speed. (This is the same behavior as the "Apply Varying" button worked in older GIMP-GAP releases 2.6 and earlier) Positive values greater than 1 result in an accelerated change of the settings, negative values result in decelerated change of the settings.

With Acceleration Characteristic value other than 0 the selected

plug-in is called 2 times in interactive mode,

1. for the background layer
2. for the top layer.

This defines the settings at begin and at end.

For all further layers, the plug-in will work non-interactive with the iterated inbetween values that were internally stored in memory as settings for begin and end, respecting the acceleration characteristic. (constant speed, acceleration or deceleration)

Note:

It is possible to iterate values of the type PARAM_DRAWABLE, (such as the BumpMap in plug_in_bump_map: You may use a layerstack-animatd Image as animatd Bump-map) This will only work, if from and to values are layers within the same image.

The iteration is done on the layerstackindex in that case. and the inbetween values for the non-interactive filtercalls will be the layers between the from and to layerstackindex. (an animation sequence as the user might expect)

With Acceleration Characteristic value 0 varying feature is turned off, and the graph is rendered as empty rectangle. In this case selected plug-in is called only once in interactive mode, For all further layers, the plug-in will work non-interactive using the last stored values.

(This is the same behavior as the "Apply Constant" button worked in older GIMP-GAP releases 2.6 and earlier)

The script sel-to-anim-img.scm simplifies the creation of animated images. Invoke the script from:

```
"<Image>/Script-Fu/Animators/Sel To AnimImage"
```

The script creates a new image with n copies of the current selection. Then it invokes the animated call of plug-in filters (optional) on the generated new image.

Notes:

- when new plug-ins are installed, they are not automatically capable to be applying with varying values, unless the plug-in implements an "_Iterator" procedure, (see docs/howto/[HOWTO-write-animatd-plug-ins.txt](#)) or registers in the GIMP_GAP configuration file

```
lastval_descriptions.txt
```

located in the gimp configuration directory
in your \$HOME (uniX) or Documents and Settings (WindowsXP)

Here is an example entry for the wideangle 3rd party plug-in:

```
# -----  
"plug_in_wideangle" #- added manually
```

```
STRUCT_BEGIN;0;48;1;  
  gdouble;0;8;1;centre_x  
  gdouble;8;8;1;centre_y  
  gdouble;16;8;1;square_a  
  gdouble;24;8;1;quad_a  
  gdouble;32;8;1;scale_a  
  gdouble;40;8;1;brighten  
STRUCT_END;0;4;1;
```

- some plug-ins may not work correct or crash when called in NON_INTERACTIVE mode. (see TESTPROT_iter_ALT)
- some plug-ins have options to create new layers or images. This does not work when they are invoked under control of the "Filter all Layers" plug-in. You should turn such options off in that case.
- some plug-in's in your PDB can have earlier or later versions. If their interfaces were changed and does not match with the interface version of the iterator procedure you get the message:(in the shell window)

```
ERROR: xxxx_Iterator stored Data mismatch in size N != M
```

when using the "Apply Varying" button.
In that case you can try to generate the needed _Iterator procedure for your plug-ins current interface by yourself.
(see file: [README_developers](#))

Filters/Filtermacro

"plug-in-filter-macro"

Filtermacro Script:

Start from Menu:

<Image>/Filters/Filtermacro

WARNINGS:

filtermacro scripts are a temporary solution. Support may be removed in future releases of gimp-gap.
filtermacros are restricted to plug-ins that are able to run with LAST_VALUES.
filtermacro scriptfiles are machine dependent, plug-in version dependent, and may not work in the expected way or even crash at execution on other machines or on execution with newer versions of the recorded filter plug-ins.

Filename:

Name of the filtermacro scriptfile.

Delete All button:

Delete the filtermacro scriptfile.

Delete button:

Delete the currently selected filtercall from the filtermacro scriptfile.

Add

Add a new filtercall to the filtermacro scriptfile. This button opens a PDB browser dialog window, where you can select filters that have been already used within the current GIMP-session. The PDB browser will show only filters that can operate on a drawable and have stored the parameters of the last run (in the current Session)

Cancel

Close the window without any further action.

OK

Close the window and apply all the filtercalls that are listed in the filtermacro scriptfile with their recorded parameters to the current drawable. (this is the drawable from where the 'Filter Macro' dialog was invoked from)

Recording of a filtermacro will automatically create an additional filtermacro reference file in case one of the recorded filters has references to additional drawables (e.g. layers). One such filter example is the `plug_in_bump_map` that uses the `bumpmap DRAWABLE` in addition to the processed input drawable.

A filtermacro reference file has the same name as the filtermacro scriptfile with extension

`.fmref`

This file records filename, type and position of such additional drawables to allow applying the filtermacro in another gimp session.

Currently supported types are:

- 3: `GAP_AINFO_ANIMIMAGE` layer in a multilayer image,
- 4: `GAP_AINFO_FRAMES`: frame (flattened representation in a sequence of numbered images)
- 5: `GAP_AINFO_MOVIE`: and movie (frame extracted from a videofile)

Where the type is automatically detected at recording like this:

- In case the drawable is the layer of an image that was extracted from a videofile within the same GIMP-session (by click on the GAP Playback dialog preview) the type 5: `GAP_AINFO_MOVIE` is recorded. Such layers are marked (with a non-persistent layer parasite) at extraction time to provide videofilename and position in the video.
- In case the drawable is a layer in an image with GIMP-GAP typical number part and another frame image with next or previous number is already existent the type 4: `GAP_AINFO_FRAMES` is assumed.
- All other cases use type 3: `GAP_AINFO_ANIMIMAGE`.

Restrictions:

Note that persistent drawable references will NOT work or give unexpected

results in case the image that contains the referred drawable was not saved, or was changed and saved after the filtermacro was recorded. Furthermore the filter must use the GIMP-GAP standard iterator implementation that supports the recording of persistent drawable id's for its additional drawable parameters.

Tip:

Filtermacro execution can be used in the same way as a single filtercall, together with the 'Filter All Layers' and the 'Frames Modify' features of gimp-gap. This way you can apply a set of filtercalls on all layers of a multilyer image (or on all selected layers in multiple frames) with one call.

Filtermacro file execution is also available in Storyboard scripts.

See [STORYBOARD_FILE_DOC.txt](#) chapter Macrofiles for more details.

Video/Bluebox

"plug-in-bluebox"

Blubox Filter

Start from Menu:

<Image>/Video/Bluebox

The bluebox filter makes the selected keycolor transparent. The bluebox filter operates on a single layer that must be of RGB(A) type. (INDEXED and GRAY images are not supported)

Keycolor:

This colorbutton shows the wanted keycolor. Pixels that are similar to this color can be set more or less transparent by the bluebox filter. You can define the detection of similar colors by Threshold(s) in 4 different ways depending on threshold mode:

Threshold Mode:

HSV	has 3 separate thresholds for hue, saturation and value and operates in the HSV colormodel.
RGB	has 3 separate thresholds for red, green and blue channels and operates in the RGB colormodel
VALUE	has only one single threshold for the maximum difference of red, green, and blue channels and operates in the RGB colormodel.
ALL	has 6 separate thresholds for HSV and RGB

Thresholds

All thresholds have the range from 0.0 to 1.0 where 0.0 has no tolerance the compared channel (requires exact match) and 1.0 allows the maximum tolerance (all values will match)

Alpha Tolerance:

Alpha tolerance value 0.0 upto 1.0
where 0.0 makes hard pixel selection by color threshold(s)
greater values allow more or less variable transparency (alpha channel values)
for the selected pixels within the threshold(s)
depending on their difference to the keycolor.
Pixels with color equal or near the keycolor appear more transparent than pixels with more color difference.
Pixels with color differences bigger than the threshold(s) are not affected by the "Alpha Tolerance" setting.

Source Alpha:

Select only pixels with alpha channel \geq source_alpha
where 0.0 is full transparent, 1.0 is full opaque.
Normally this is set to 0.2 to protect the alpha_channel of pixels that are already transparent against changes by the bluebox filter.

Target Alpha:

Control the minimum alpha value for the selected pixels.
where 0.0 sets a full selected pixel ($==$ matching with keycolor) to full transparent,
0.5 sets a full selected pixel to half transparent
and 1.0 to full opaque.

Feather Edges:

Enable smoothing the selection using the feather radius.

Feather Radius:

Feather radius in pixels, makes the selection smooth.
(this is done internally by applying a blur filter on the selection that was built by keycolor, threshold(s) alpha tolerance, shrink/grow and target alpha settings)

Shrink/Grow:

Grow or shrink the selection in pixels,
where negative values are for shrinking the selection
(that was built by keycolor, threshold(s) alpha tolerance, and target alpha settings)

The bluebox filter creates an image for preview of the effect.
This image contains an (optional scaled) copy of the original layer.

Automatic Preview:

When checked perform automatical update of the preview image on all changes of the parameter settings.

Preview Button:

Create or update the preview image.

Previewsize:

Set size of the preview image in percent of the original.

Video/Frames Crop

"plug-in-gap-anim-crop"

Crop Video Frames

Start from Menu:

<Image>/Video/Frames Crop

With the video crop feature you can crop all frame images in a sequence of numbered frames. Load one of the frames and invoke this plug-in from the loaded frame, adjust the settings as you like and the plug-in will crop all other frames down to the selected size when OK is pressed. (the parts of the frames outside the cropping rectangles are will be lost after this operation)

Current width and height are displayed as information. With "New Width" and "New Height" you can specify the desired new size in pixels.

When the "Reset" button is pressed, the new width and new height values are reset to current values.

With X/Y ratio values you can specify new size as factor, A value of 0.5 crops down to half size.

The chain button keeps image proportions constant when active. With inactive chain button (symbol changes to broken chain) width and height may be changed independent.

Offset

X and Y offset values (in pixels) for placing the image within the new canvas size. there are buttons for independent centering horizontal and vertical.

There is also a visual control widget where you can arrange the position of the current image size within the new image size via dragging with the mouse. This updates the offsets according to the desired position. The new (reduced) image size is shown as smaller black outlined rectangle.

Video/Frames Resize

"plug-in-gap-anim-resize"

Resize Video Frames

Start from Menu:

<Image>/Video/Frames Resize

With the video resize feature you can change the canvas size for all frame images in a sequence of numbered frames. Load one of the frames and invoke this plug-in from the loaded frame, adjust the settings as you like and the plug-in will apply the new canvas size to all other frames when OK is pressed.

Current width and height are displayed as information. With "New Width" and "New Height" you can specify the desired new size in pixels.

When the "Reset" button is pressed, the new width and new height values are reset to current values.

With X/Y ratio values you can specify new size as factor. A value of 2 gives double size, 0.5 half size.

The chain button keeps canvas proportions constant when active. With inactive chain button (symbol changes to broken chain) width and height may be changed independent.

Offset

X and Y offset values (in pixels) for placing the image within the new canvas size. there are buttons for independent centering horizontal and vertical.

There is also a visual control widget where you can arrange the position of the image on the new canvas size via dragging with the mouse and updates the offsets according to the desired position.

Video/Frames Scale

"plug-in-gap-anim-scale"

Scale Video Frames

Start from Menu:

<Image>/Video/Frames Scale

With the video scale feature you can scale all frame images in a sequence of numbered frames.

Load one of the frames and invoke this plug-in from the loaded frame, adjust the settings as you like and the plug-in will scale all other frames to the selected size when OK is pressed.

Current width and height are displayed as information. With "New Width" and "New Height" you can specify the desired new size in pixels.

When the "Reset" button is pressed, the new width and new height values are reset to current values.

With X/Y ratio values you can specify new size as factor,
A value of 2.0 results in double saize while 0.5 sacles down
to half size.

The chain button keeps image proportions constant when active.
With inactive chain button (symbol changes to broken chain)
width and height may be changed independent.

Note:

GIMP provides different methods for scaling images.
You can configure the desired method in the GIMP
Preferences dialog at "Tool Options"
where you can select different methods for scaling.

- None (fastest)
- Linear
- Cubic (best)

This plug-in uses the configured method from the
preferences menu.

Video/Delete Frames

"plug-in-gap-del"

Delete Frames

Start from Menu:

<Image>/Video/Delete Frames

This feature deletes frames.
this is done starting at the current frame number
(inclusive) until the specified frame number (inclusive).

The current frame number is shown in the headline
text "Delete frames from <current framennr> to (number)

There is no undo for this operation, so please use
it with care.

Notes:

Normally there is a confirmation dialog before
the frames are deleted.

The confirmation dialog is configuration dependent
and can be suppressed by adding this configuration
parameter to your gimprc file:

```
# the video-confirm-frame-delete can have values "yes" or "no"  
# where "no" does not show the confirm dialog windows when the user  
# tries to delete video frames of the video.  
# (default is "yes")  
(video-confirm-frame-delete "no")
```

Video/Frames Density

"plug-in-gap-density"

Change Frame Density

Start from Menu:

<Image>/Video/Frames Density

Changing the frame density results in duplicating frames for increasing frame density, and will delete frames for decreasing the density.

This feature is used to change the target framerate and can be useful for mixing videos with different framerates without changes of the playback speed.

Example:

If you have created an animation with framerate of 8 frames/sec and want to change to a target framerate of 24 frames/sec. you should call the "Frame Density" plug-in with a density factor of 3 and increasing density mode.

This makes 2 copies for each of the frames in the selected range. If you use playback at the old 8 frames/sec rate the motion is 3 times slower now. With the new Rate of 24 frames/sec the motion is the same speed as before.

The animation before the frame density change (frames 1-9):

```
01 02 03 04 05 06 07 08 09    (original framenumner)
```

The animation after frame density change at growing mode and factor 3 and limited to a frame range from frame 3 to frame 8:

```

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
(new framenumner)
01 02 03 03 03 04 04 04 05 05 05 06 06 06 07 07 07 08 08 08 09
(original framenumner)
          + +      + +      + +      + +      + +      + +

```

Tip:

If you use storyboard based encoding you can set the stepsize parameter for clips that should be encoded with other target framerates. That way you dont need to duplicate or drop your source frames.

Tip:

To get smooth motion in the resulting frames you may create 2 onionskin layers per frame (for the range "FromFrame" 03 "ToFrame" 20) using a setting that blends in the next 2 frames ("FrameReference" +1) at stackposition 0 FromTop of the layerstack with opacity of 66% and 50% and without ignoring any background layer(s).

That way you get a cross-fading effect with smoother motion when played with 24 frames/sec

You can find onionskin in this menu:
<Image>/Video/Onionskin/Configuration

Density factors are limited to values from 1.0 upto 100.0. float values are supported.

If you want to change from 30 frames/sec down to 15 frames/sec then use a density factor of 2 and decreasing density mode (increasing/growing mode switched off).

This will delete every 2.nd frame.

There is **no undo**, so use this feature very carefully.

Video/Duplicate Frames

"plug-in-gap-dup"

Make Duplicates of Frame Range

Start from Menu:

<Image>/Video/Duplicate Frames

This feature duplicates the specified range of frames N-times.

The duplicated frames are inserted after the current frame (the frame from where this plug-in was invoked)

When this plug-in is started, the settings "From Frame", "To Frame" and "N times" are initialized in a way that it will duplicate the current frame once.

Example:

when the current frame is 2 out of 5 frames and we apply the settings:

```
From Frame: 2
To Frame:   4
N times:    2
```

As result 6 frames will be inserted after frame 2. The inserted frame sequence is made as copy of the frames (referring to the old frame numbers):

```
pic_000001.xcf
pic_000002.xcf

/* the newly inserted frames */
pic_000003.xcf (copy of old pic_000002.xcf)
pic_000004.xcf (copy of old pic_000003.xcf)
pic_000005.xcf (copy of old pic_000004.xcf)
```

```

pic_000006.xcf  (copy of old pic_000002.xcf)
pic_000007.xcf  (copy of old pic_000003.xcf)
pic_000008.xcf  (copy of old pic_000004.xcf)

/* the renamed remaining frames */
pic_000009.xcf  (renamed src was frame number 3)
pic_000010.xcf  (renamed src was frame number 4)
pic_000011.xcf  (renamed src was frame number 5)

```

Video/Exchange Frame

"plug-in-gap-exchg"

Exchange Current Frame

Start from Menu:

<Image>/Video/Exchange Frame

This feature exchanges the current frame with the frame specified by "Number".

Video/Filename to Layer

"plug-in-name2layer"

Render Filename to Layer

Start from Menu:

<Image>/Video/Filename to Layer...

This plug-in renders the filename of the image or just the number part of the filename onto the image. If the parameter `create_new_layer` is 0, the filename is rendered to the active drawable (the layer that was active when this plug-in was invoked). Otherwise a new textlayer is created.

Mode:

```

"Number Only"    ... render just the number part
"Filename"       ... render the filename without directory path
"Path/Filename" ... render the filename including the directory path

```

Fontname:

Entry for the Fontname

Fontsize:

Fontsize in pixels

X:

Position x offset in pixels

Y:

Position y offset in pixels

Antialias:

ON: Use Antialiasing
OFF: No Antialiasing

Create Layer:

ON: Create a new layer,
OFF: Render on the active drawable

Tip:

To create framenumbers in all of your frames you can use the "Modify Frames" (<Image>/Video/Frames Modify) feature and choose "Apply filter on Layer(s)" as function. In the next dialog step, the GIMP-GAP DB-Browser window select "plug_in_gap_renumber" as filter name. (this is the internal PDB-Name of the "Render Filename to Layer" plug-in)

Then choose "Apply constant" in the DB-Browser window. This should open the "Render Filename to Layer" dialog window (only for the first frame) where you can adjust the settings. Choose "Number Only" mode for rendering only the framenumbers part, and "Create Layer" checkbox turned on if you want the number to be created as separate layer.

With "Create Layer" turned off the number will be rendered on (all) the layer(s) that were selected in the initial "Modify Frames" dialog window.

Video/Frames Modify

"plug-in-gap-modify"

Modify Frames

Start from Menu:

<Image>/Video/Frames Modify

The GIMP-GAP plug-in 'Frames Modify' provides the feature to perform functions on one or more selected layer(s) in all frames of the selected frame range.

Layer Selection:

The layers can be selected by (parts of) their name, or by their layerstack numbers.

The selection is controlled via the LayerPattern entry, the case sensitivity toggle and the radio buttons.

The toggle button "Invert Layer Selection" inverts the selection.

Some of the available functions affect the image and do not allow layer selection.

The Function menu:

Layer Attributes
"Set Layer(s) visible",

```

"Set Layer(s) invisible",
"Set Layer(s) linked",
"Set Layer(s) unlinked),
Layer Modes
  "Set Layer(s) mode: Normal",
  "Set Layer(s) mode: Dissolve",
  "Set Layer(s) mode: Multiply",
  "Set Layer(s) mode: Divide",
  "Set Layer(s) mode: Screen",
  "Set Layer(s) mode: Overlay",
  "Set Layer(s) mode: Difference",
  "Set Layer(s) mode: Addition",
  "Set Layer(s) mode: Subtract",
  "Set Layer(s) mode: Darken only",
  "Set Layer(s) mode: Lighten only",
  "Set Layer(s) mode: Dodge",
  "Set Layer(s) mode: Burn",
  "Set Layer(s) mode: Hardlight",
  "Set Layer(s) mode: Softlight",
  "Set Layer(s) mode: Color erase",
  "Set Layer(s) mode: Grain extract",
  "Set Layer(s) mode: Grain merge",
  "Set Layer(s) mode: Hue",
  "Set Layer(s) mode: Saturation",
  "Set Layer(s) mode: Color",
  "Set Layer(s) mode: Value",
Layer Stackposition
"Raise Layer(s)",
"Lower Layer(s)",
Merge Layers
"Merge Layer(s) expand as necessary",
"Merge Layer(s) clipped to image",
"Merge Layer(s) clipped to bg-layer",
Selection
"Replace selection (source is the active frame)",
"Add selection (source is the active frame)",
"Subtract selection (source is the active frame)",
"Intersect selection (source is the active frame)",
"Selection none",
"Selection all",
"Selection invert",
  "Selection from alpha channel (individual per frame)",
"Save selection to channel (individual per frame)",
"Load selection from channel (individual per frame)",
"Delete channel (by name)"
Layer Mask
"Add white layermask (opaque)",
"Add black layermask (transparent)",
"Add layermask from alpha",
"Add layermask transfer from alpha",
"Add layermask from selection",
"Add layermask from bw copy",
"Invert existing layermask",
"Apply filter on layermask",

```

```
"Delete layermask",
"Apply layermask"
"Copy layermask from layer above"
"Copy layermask from layer below"
```

```
"Apply filter on Layer(s)",
"Duplicate Layer(s)",
"Delete Layer(s)",
"Rename Layer(s)",
"Add alpha channel",
```

The Functions

```
"Replace Selection",
"Add Selection",
"Subtract Selection",
"Intersect Selection",
```

take the Selection from the initial (invoking) frame and combines it with existing selections in all the other handled frames.

The Functions

```
"Copy layermask from layer above"
"Copy layermask from layer below"
```

create Layermasks in the selected layer(s) in all handled frames by copying from the neighbour layer above (or below) the specified layer(s). If there is no such neighbour layer, a black (full transparent) layermask will be created for the selected layer(s). If the neighbour layer has no layermask, the mask will be create as black and white copy of the neighbour layer. The layermask is placed at the center of the selected (destination) layer. If the neighbour layer is smaller than the selected layer, the uncovered border of the created layermask is created full transparent.

The Functions:

```
"Duplicate Layer(s)"
"Rename Layer(s)"
"Save Selection to Channel"
"Load Selection from Channel"
"Delete Channel (by Name)"
```

require a Name in the Entry field "New Layer Name" (the name of this entry changes to "New Channel Name" when dealing with channels)

The function

```
"Apply filter on Layer(s)"
```

brings up a dialog window when started via pressing the OK button. this dialog is similar to the PDB-Browser, where you can select one of the available filters.

If you use the PDB-Browser's button "Apply Varying", the filterparameters will slightly change in each handled frame from start to end.

- If there is more than one selected layer in a frame each of the selected layers within the same frame will be processed with the same filter parameter values.

(for more information see chapter 'Animated calls of Plug-In Filters' below)

The function
"Apply filter on layermask"

works with same additional dialog steps as described for function "Apply filter on Layer(s)", but applies the selected filter to the layermask instead of the layer itself.

If the first selected layer in the first handled frame image has no layermask, this function will terminate with an appropriate error message.

If the filter shall be applied with varying values, it is also required that the first selected layer of the last handled frame image has a layermask.

If the selected layer(s) of the frame images of the selected frame range between first and last handled frame have no layermask, the filter processing is skipped on those layers, but processing will continue.

It is a good idea to use the same layerstack structure in all your frames. Another hint is that you assign useful names to your layers. that should be done consequent for all frames.

Example 1 (useful Layerstack structure/names)

```

                film_000001.xcf    film_000002.xcf    ....    film_000010.xcf
-----
layerstack [0]  mouse_01          mouse_02          mouse_10
layerstack [1]  cat_01             cat_02            cat_10
layerstack [2]  tree_01            tree_02           tree_10
layerstack [3]  background         background        background

```

Example 2 (not recommended)

```

                film_000001.xcf    film_000002.xcf    ....    film_000010.xcf
-----
layerstack [0]  tree              layer             mouse_10
layerstack [1]  bg                layer             cat_10
layerstack [2]  layer              layer             background
layerstack [3]  background         background

```

Video/Morph`"plug-in-gap-morph"`

Morph /Warp

Start from Menu:

<Image>/Video/Morph

Video/Morph Tweenframes

Video/Morph One Tween

[New] Video/Morph Workpoint Generator

This plug-in provides metamorphose transformation and warp deformations, depending on the selected render mode.

Warning: this plug-in is still in experimental state.
it may change in future releases.

Morphing:

Morphing takes a source layer and a destination layer as input and creates the steps inbetween (as newly added layers) to transform the source layer into the destination layer.

The warp transformation is controlled by a set of movement vectors. A movement vector is shown in form of 2 corresponding points (start point in the source layer, the corresponding end point in the destination layer)
When the checkbutton "lines" is ON, vectors are displayed as lines in the destination layer.

Points are created via mouse click in the source or destination layer preview and can be picked and dragged with the mouse.

documentation to be continued

Video/Move Path`"plug-in-gap-move-path"`

Move Path (make things move)

Start from Menu:

(<Image>/Video/Move Path)

For this plug-in you need a series of frames (destination video frames) and one single image or another series of frames. (source object or moving object)

If you use a single image as source object all the layers in it can be used as source frames.

The source frames are copied into the destination frames. Those copies can be transformed in many ways, using controlpoints and appears as one new layer in each handled destination frame. The X/Y controlpoint coordinates affects the offset of the copied new layer in the destination frame(s). All controlpoints build up a path that defines how the copied new layer moves within the destination frames.

Preconditions:

- The source image must be opened in GIMP
- The source image must be another Image than the destination frame (if you really want to copy layers from destination frame to destination frame(s) you have to duplicate it first)
- The source image must be from the same type (RGB, INDEXED ..) as the destination frame.

In other words: To run the 'Move Path' plug-in you have to open at least 2 images of the same type.

The selected layer(s) of the source image or frame is (are) copied into the selected range of frames. Each handled frame receives exactly one copy of the selected layer from the source image, where the copy will be modified by transformations such as scaling, changing transparency and more, according to the controlpoint settings.

If you use stepmodes "Loop", "Loop Reverse", "Once", "OnceReverse" or "PingPong"

the layers of the source image are stepped through, and the next handled frame receives the next layer from the source image's layerstack.

For the frame based stepmodes ("Frame Loop", "Frame PingPong" ...) the selected source(layer) should be a layer of another video frame. In the frame based stepmodes the source object is considered as video frames and stepping is done from frame to frame (and not from layer to layer)

The inserted layer is always a copy of one frame of the source object animation (no matter which source layer was selected) where all the visible source layers are merged to build up the moving object.

The frame based modes can be used to mix videos with many frames, because there is no need to convert one of the videos to a multilayer image.

If you use a normal image (that is not a video frame) as source for the frame based stepmodes it acts like an animation with

only one frame.

The "Step Speed" factor

decides how fast to step through the source object,
in relation to the target frames.
With factor 1.0 source and target will step synchron.
Factor 0.5 slows down the source steps to half speed.
The same source layer is copied into 2 frames,
before the advance to the next source layer is done.

The "Step Speed" factor is not relevant for the
stepmodes "None" and "FrameNone".

Selection Handling

The "Move Path" plug-in can ignore or respect selections (selected Areas)
in the source image (or source frame(s))

The Optionmenu Modes are:.

- "Ignore Selection (in all Source Images)"
Ignores all selections.
- "Use Selection (from Initial Source Image)"
Takes the selection from the initial source image (or frame)
and applies it to all handled copies.
This makes all unselected pixels of the moving object
transparent. You can get smooth edges
depending on the feather_radius settings in the controlpoints.
- "Use Selections (from all Source Images)"
Uses all selections in all source frames.
The source frames without selections are handled
as if all pixels were selected.
This mode is only relevant for the frame based stepmodes,
"Frame Loop", "Frame Once", "Frame Loop Reverse",
"Frame OnceReverse", "Frame PingPong"

The copies of the source layer(s) were modified by transitions
with varying parameters.

Parameters:

- SourceLayer (depends on Stepmode)
- Position (X/Y)
- Size Scaling (Width/Height)
- Opacity
- Rotation (angle from -360 to +360 degrees)
- Selection Feather Radius (in Pixels)
the feather radius controls how much to smooth the edges of the

selections

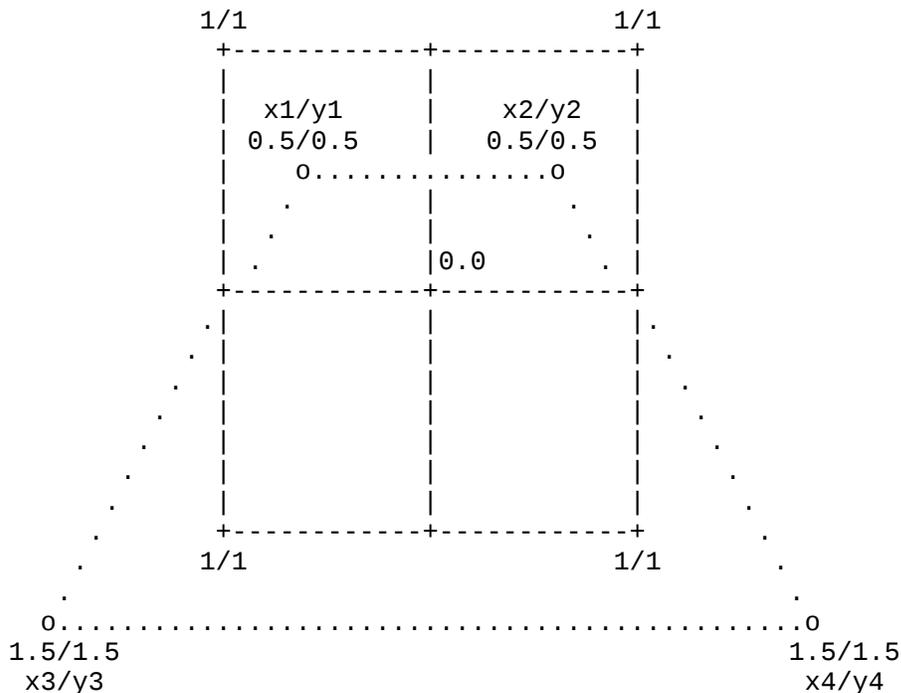
(this is only relevant for source selection handling modes other than
"Ignore Selection")

- Perspective Transformation (transformation factors for X/Y of the 4
Corner Points)

Perspective transformation factors are scaling factors
that are applied to the coordinates of the 4 corners.
If all 8 Coordinates have the value 1.0 no scaling is done
and the result is the same as the original.

Scaling with factor 0 moves the affected (x or y) coordinate to the middle.

Factor 2.0 applied to Coordinate moves it outwards by half of the original width (for X coordinates) or half height (for Y coordinates).



The controlpoint parameters were changed linear from one starting controlpoint to the next controlpoint.

Per default the move path has only only 1 controlpoint.

(So the source layers(s) are copied to all frames of the framerate at constant position, size and opacity)

If you want your source layers to move, grow, rotate or to fade (in or out) you have to add one more controlpoints (limited to 1024) to define a path.

The affected range is selected by start frame - end frame.

Each affected frame receives exactly one copy of the (current) source layer adjusted to the current controlpoint parameters.

The layerstack defines if the pasted copy appears in the foreground (0 == on top) or below other layers that are already in the frame.

With the toggle button 'Clip To Frame' the the copied layer is clipped to the destination frames image width and height.

With the frame number in the preview frame you can select the frame number to display in the preview.

You have to press the "UpdPreview" button for explicit update of the preview.

Controlpoints:

The move path is defined by controlpoints.

Only the current controlpoint is displayed with all its values.

If checkbox "Show Path" is on, all the controlpoints are shown in the preview window, connected with pathlines. Further it enables picking controlpoints and dragging controlpoint coordinates (X/Y) in the preview with the left mousebutton. With the right mousebutton you always drag the coordinates of the current controlpoint (without picking other controlpoints)

There are controlpoint edit buttons to
"Add Point"
"Insert Point" "Delete Point"

The Button "Grab Path"
deletes all controlpoints and replaces them with all anchor points of the vector based gimp path in the invoker image (the image from where the MovePath was started).

If you hold down the SHIFT key while you press the "Grab Path" button, the conversion to controlpoints tries to follow the vector based path along the bezier curve. This is done by creating a controlpoint per handled frame. The more controlpoints are used, the closer you get to the original vector based bezier path with the moving object.

With the Buttons
"Prev Point" "Next Point"
"First Point" "Last Point"
you can step from controlpoint to controlpoint, and make other controlpoint to the current controlpoint.

"Reset Point" "Reset All Points"
does reset width, height and opacity of the controlpoint to 100% , perspective factors to 1.0 (no perspective transformation) and rotation to 0 degree, but leaves the path (X/Y values) unchanged.

The "Reset All Points" button has multiple functions controlled by holding down a modifier key:

- Holding down the shift key copies the setting of point1 into all other points. (except X/Y values)
- Holding down the ctrl key spreads a mix of the settings of point1 and the last point into the other points inbetween. (except X/Y values)

"Delete All Points"
removes all controlpoints.

"Rotate Follow"

Calculate rotate values for all controlpoints to follow the path.

An object moving along a horizontal line from left to right results in an angle of 0 degree. (or a multiple of 360 degrees if the path builds circular loops)

A vertical Move from top to bottom gives 90 degrees.

SHIFT: If this button is clicked while the Shift Key is pressed, a fix rotation offset is added to all the calculated rotation values. The rotation offset is taken from the current rotate value of controlpoint 1.

If an object moves from right to left the calculated angle is 180 degree and the object appears upside down.

With a startoffset of 180 (or -180) you can compensate this effect.

"Save Points"

saves your controlpoints to file

"Load Points"

loads controlpoints from file

The "Instant Apply" checkbox

does automatic update of the preview when checked.

The automatic update needs much CPU and IO power especially when big images are used as source and/or destination or those images have many layers.

(dont use the instant_apply on slow machines.)

Tips:

- with the UpdPreview Button (or "Instant Apply" checkbox) you can blend in the selected layer of the source image. If you want to adjust position it may be useful to see the background. Therefore you can make the source image transparent (modify the opacity value) or put the sourcelayer below the background (set the layerstack to higher value) Then pres UpdPreview button again.
- Whats wrong if the "UpdPreview" was pressed, but the preview does not show the source object ?
 - maybe the source object is an invisible layer. (if the "Force Visibility" checkbox is turned ON invisible Layers should become visible).
 - maybe the current opacity setting is 0% (or nearly 0%)
 - maybe the current X and/or Y position are outside

- the image
- maybe the current scaling factors are 0% (or nearly 0%) and the result is only 1x1 pixel or smaller
- maybe the source object has only fully transparent pixels

- If you let your objects (source layers) rotate, perform perspective transformations or change their size, set handle mode to 'Center'.

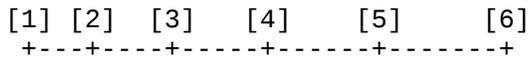
If you use another handle mode you may get unwanted moves of your object, caused by resizing.

- Speed:

If no keyframes are set, and acceleration characteristic value 0 is specified the "Move Path" plug-in alternates the settings linear from controlpoint to controlpoint, so things move (or happen) in constant speed between 2 controlpoints.

If you want to make accelerated moving objects, you can set more controlpoints with growing distances.

Example:



The affected range has 25 frames, and you have set 6 points with growing distances in one straight line and without specifying keyframes for those points. That gives 5 frames (== equal time) for each part of the path, but each part has another length. This results in different (growing) speeds for each part of the path.

Acceleration characteristic:

Since GIMP-GAP-2.7 speed can be controlled with acceleration characteristic presets. Those presets are integer values where value 1 defines constant speed along a path segment, independent to the distance between the controlpoints of the path segment. (How to define path segments see chapter Keyframes below) positive values result in acceleration, negative values in deceleration.

- 0 ... implicate speed control, compatible to GAP 2.6.x and older releases (This is the default setting for all acceleration characteristic values)
- Each line between 2 controlpoints gets an equal time slot (e.g number of frames)
- The speed between 2 controlpoints is constant, but depends on the length of the line, where short lines result in low speed and long lines result in high speed.

For all other Acceleration characteristic values than 0 the line length between controlpoints does not affect the speed of the moving object.

1 ... Constant Speed
 object moves through a path segment with constant speed.
 A path segment includes all path lines between two keyframes
 (note that first and last controlpoint are considered as
 keyframes without explicit keyframe value)
 In this mode the line length between controlpoints does not
 affect the speed of the moving object.

positive value ...Acceleration
 object moves through a path segment with increasing speed.
 higher values result in slower speed at start and higher speed
 at end of the path segment

positive value ...Deceleration
 object moves through a path segment with decreasing speed.

Acceleration characteristic values can be specified independent for

- o) Movement ... Movement of the object
- o) Opacity ... To control speed of the opacity changes
- o) Size ... To control speed of size changes (e.g. zooming)
- o) Rotation ... To control speed of object rotation
- o) Perspective ... To control speed of object perspective transformations
- o) FeatherRadius... To control speed of selection feather radius changes

Each of those values can be changed in the Acceleration Tab with a spinbutton where you can enter a value.

A graph next to the spinbutton shows the selected acceleration characteristic curve.

The acceleration characteristic can also be changed by clicking on the graph and dragging vertically with the mouse.

A straight line from left bottom to right top corner of the graph is drawn for value 1 that represents constant speed.

Acceleration characteristics are only relevant on the 1st controlpoint, and controlpoints that are marked as KEYFRAME but not on other controlpoints.

For controlpoints that are not relevant, the spinbutton widgets in the acceleration characteristic tab are disabled.

In case acceleration characteristic values $\neq 0$ are used for any other settings than movement

all controlpoint settings other than position (X/Y) of NON-keyframes are ignored.

(note that first and last controlpoint are implicit keyframes and therefore always relevant)

Example:

have a path with 10 controlpoints, none of them marked as keyframe
specify rotation 0 for first controlpoint, rotation 360 for the last
controlpoint and select both Acceleration characteristic for Movement
and Rotation value 1 (for constant speed)

This settings move the object along the path of 10 points and rotate it
with constant speed from 0 to 360 degree, regardless what rotation
settings are define in controlpoints 2,3,4,5,6,7,8,9.

Speed of the moving object can be controled for each segment of the path,
where a segment is the distance to the next KEYFRAME. In case there are no
explicite Keyframes the path has only one segment with full length
from the first to the last controlpoint.

When acceleration characteristic other than 0 is used for movement,
but not for other transitions (rotation, opacity ...), then the move path
tool automatically synchronizes frame timing for those other transitions
with the frametiming of the movement.

Example:

Have a path with 5 controlpoints and no explicite keyframe set.
At the first controlpoint specify acceleration characteristic 12
for movement to start slow and increase speed of the moving
object while moving along the path.
Further let rotation of the object follow the path
(generate the rotation settings by pressing the "Rotate Follow" button)

In this example the rotation follows the current position of the
moving object with the same acceleration as the movement.

- Keyframes

Keyframes can be used optional, to fix a controlpoint
to the given frame number. The first and last controlpoints
are implicite keyframes, always fixed to start or end
frame number. (The keyframe entry is set insensitive
on the first and last controlpoint)
With the help of keyframes you can control exactly
when things should happen.

Use a value of 0 in the keyframe entry if you dont want
to fix a controlpoint to a keyframe.

Keyframes are shown as absolute frame number
in the "Move Path" dialog window, but they
are saved as relative values in the
controlpointfile.
(if start frame = 5
and a keyframe is displayed as 7
the keyframe is internally stored as 2 (7 - 5)

Keyframes are also used to define path segments.
Where a Path segment includes all following controlpoints
until the next Keyframe.

Independent acceleration characteristics can be set at
those controlpoints that are the begin of a path segment.
(e.g the 1st controlpoint and all keyframes)

The current Segment Number, current segment length (in pixels)
and minimal / maximal speed (in pixels per tween) are displayed
below the preview.

The current Segment is the segment that includes the current
displayed controlpoint.

Example:

You have 500 frames and want to render an object
that stands still at coordinate 320/200 in the 1st 50 frames,
than accelerates until frame 150, moves with constant speed
until frame 400, decelerates until frame 450 and stays still
at coordinate 470/150 until the end frame 500.

Tip:

To avoid unwanted speed jumps in this multi segmented
example, the controlpoints should be set in a way
that max speed value of segment B is nearly equal
to the max speed value of Segment C and segment D.
Note that changes of control point coordinates affect the
length of the path segment, acceleration characteristic,
and number of involved frames (or tweens) also have
influence on the speed of the object within a segment.

#Segment A (standstill in Segment Number 0)

[0] x:320 y:200

#Segment B (increasing speed due to acceleration value 20 for Movement)

[1] x:320 y:200 keyframe:50 acceleration Movement: 20

[2] x:330 y:210

[3] x:350 y:200

[4] x:360 y:190

#Segment C (move with constant speed due to acceleration value 1 for

Movement)

[5] x:362 y:170 keyframe:150 acceleration Movement: 1

[6] ...

[7] ...

[8] ...

[9] ...

[10] ...

#Segment D (decreasing speed due to acceleration value -15 for Movement)

[11] x:500 y:150 keyframe:400 acceleration Movement: -15

[12] x:490 y:160

[13] x:475 y:155

[14] x:475 y:155

```
#Segment E (standstill)
[15] x:470 y:150 keyframe:450
[16] x:470 y:150
```

- Checking Controlpoints
 - The check is done at "OK", and "Anim Preview" button,
If errors are detected, they are shown in a pop-up window and the action is not performed.
 - The number of controlpoints is now checked against the number of affected frames.
(You can't have more controlpoints than frames)
 - If keyframes are used, they must be in (ascending or descending) sequence
- AnimPreview
With this button you can generate an animated preview to get an idea how the inserted moving object will look like. The animated preview is generated as multilayer image and the Filter/Animation/Playback plug-in is started on that multilayer image.

The button opens a dialog window, where you can enter options for the animated preview.

- Object on empty frames
renders quick on empty frames (filled with background color)
(scale down speeds up rendering time)
- Object on one frame
renders quick on one frame (preview frame)
(scale down speeds up rendering time)
- Exact Object on frames
renders slow, but exactly on the selected framerate.
(scale down increases rendering time)
- Scale Preview
Optional you can scale down the animated preview size (100% down to 5%)
- Framerate
The framerate is used in the generated multilayer image only.
- Copy to Video Buffer
Optional you can copy the preview frames to the video buffer
(can be pasted in the GIMP-GAP VCR Navigator)
- Force visibility
If this checkbox is set, all source layerobjects are set visible when they are copied into frames.

"Move Path" advanced settings

- Bluebox
With the bluebox check_button you can apply the bluebox filter effect on the moving object, that makes the keycolor transparent.
- The Keycolor button
Opens a dialog window where you can set all the parameters of the bluebox filter.
If clicked with the right mouse button you can grab the keycolor from the current FG/BG color (in the GIMP main window)
- Tracelayer
a trace layer shows all positions of the moving object from begin until the position of the previous frame. Positions in virtual frames (tweens) are also included in the trace layer.
If descending Opacity is used, the trace is fading out at the older (previous) Positions of the moving object.
- Tweensteps
This feature is for rendering fast moving Objects.

The "MovePath" plug-in can calculate the moving object for virtual frames (tweens) between real frames. You can control this by setting tween_steps to a value greater than 0. In that case the "MovePath" plug-in creates an additional layer (the "Tweenlayer") and inserts that tween layer below the stackposition of the current moving object in the next real frame. The tween layer shows the moving object at the positions of all virtual frames between two real frames.

Example: The selected frame range is 10 frames and the tweensteps value is 2. In this case 28 steps are processed internal. 10 real frames + 18 virtual frames (2 virtual between each real frame)

Use descending opacity to fade out tweens.

The tween nearest to the real frame is drawn with the initial opacity value, the other tweens are reduced more and more if descending opacity is less than 100%.

With tweens and opacity settings you can produce motion-blur effects for fast moving objects.

If both tween processing and trace layer are active, the "Tweenlayer" is added, but set invisible. This is done because the trace layer contains already all the tween steps. A visible tween layer would produce unwanted increase of the total opacity of the moving object in the composite view.

Video/VCR Navigator`"plug-in-gap-navigator"`

VCR Navigator

Start from menu:

`<Image>/Video/VCR Navigator`

The VCR Navigator dialog shows a thumbnail view, frame_number and time (mm:ss:msec) of the frames in a scrollable listbox.

AL-Tracking:

Tracking of the active layer can enhance the workflow when stepping from frame to frame. This feature tries to find a layer in the newly loaded frame that matches with the active layer of the previously displayed frame (by name or stackposition) and set the active layer to that matching layer automatically.

Radio buttons

`"OFF" ... disable tracking of active layer`

`"Name" ... enable tracking of active layer by name.`
 Layernames are compared from left to right and the layer with the most matching layername characters is selected.

If there are more layers with same amount of matching chars case sensitivity, stringlength and stackposition are taken into account too.

`"Pos" ... enable tracking of active layer by stackposition`
 where positions of onionskin layers are not counted.

framerate:

You can set the global framerate for all frames here.

Note: changing the framerate in the VCR Navigator dialog window does not change the number of frames. Therefore the motion does speed up or slow down at playback. If you want to keep your motion speed constant at increased framerate use the "Frames Density" tool to create (or delete) the requested additional frames.

timezoom:

with timezoom you tell the VCR Navigator dialog window to display every n-th frame only, to get an overview if there are many frames.

Framerate and timezoom values are stored in a gap video info file named:
`<framebasename>_vin.gap`

Double click with left mousebutton on a frame in the listbox performs a goto operation, which loads the clicked frame as current frame.

Single click with left mousebutton

selects one frame (deselecting all others)

Ctrl click with left mousebutton

selects one frame (additional to the current selection)

Shift click with left mousebutton

selects a range of frames.

Click with right mousebutton

brings up a pop-up menu

Pop-up menu

copy,

cut

paste before

paste after

paste replace

clear pastebuffer

select all

select none

cut and copy

is sensitive if any frame is selected,

paste and clear pastebuffer

is sensitive if video paste buffer contains

at least one frame.

This menu enables you to cut, copy and paste the selected frames even between different videos.

Size and type of the handled frames are converted at pasting if needed.

The palette of the current frame is used for the pasted frames when the destination type is INDEXED.

There are 3 types of paste:

before:

Insert frames before the current frame.

Use this mode if you want to insert frames

before the first frame.

after:

Insert frames after the current frame.

Use this mode if you want to insert frames

after the last frame.

replace:

Replace frames beginning at the current frame

with the frames from the video paste buffer,

and does overwrite existing frames.

If there are selected frames in the VCR Navigator dialog window, the current frame is set to

the first selected frame and the paste operation is related to the first selected frame

The paste buffer is located in the filesystem and can be configured in the gimprc file:

```
(video-paste-dir "/home/hof/gap_video_paste_dir")
(video-paste-basename "gap_video_paste_")
```

The duplicate button in the VCR Navigator dialog duplicates the selected frames immediately into the current video, without writing to the video paste buffer.

The delete button in the VCR Navigator dialog deletes the selected frames without writing to the video paste buffer.

IMPORTANT NOTE:

There is no undo for GIMP-GAP operations, and the deleted frames can not be restored.

Video/Frames Convert

"plug-in-gap-range-convert"

Frames Convert

Start from Menu:

<Image>/Video/Frames Convert

Converts multiple frames from one fileformat to another.

Basename:

Name of one destination frame without the frame number part and without the extension.

Extension:

Extension of the destination frames. The extension also defines the fileformat.

Imagetype:

- Keep Type
- Convert to RGB
- Convert to Gray
- Convert to Indexed

Flatten:

Merge all layers if checked. This check_button should be turned on for destination fileformats that can not store multiple layers.

Example1:

If your source frames are XCF-images of imagetype RGB with the names:

```
img_000001.xcf  
img_000002.xcf  
img_000003.xcf
```

and you want to convert to JPEG's named

```
/tmp/scratch_000001.jpg  
/tmp/scratch_000002.jpg  
/tmp/scratch_000003.jpg
```

you need the following convert settings:

```
From Frame: 1  
To Frame: 3  
Basename: /tmp/scratch_  
Extension: .jpg  
Imagetype: Keep Type  
Flatten: ON (checked)
```

Example2:

To convert from RGB JPEG fileformat (.jpg) to GIF (.gif)
you have to set the destination imagetype "Convert to Indexed",
because the GIF fileformat can not handle RGB, only GRAY and INDEXED)

Convert to INDEXED reduces the number of colors
down to 256 (or less).

Video/Frames Flatten

"plug-in-gap-range-flatten"

Flatten Frames

Start from Menu:

<Image>/Video/Frames Flatten

This feature flattens all the frames in the range
specified via "From Frame" and "To Frames".

Warning:

Flattening melts down all layers to one resulting
background layer and replaces transparent regions
with the background color.
(because flattening removes the alpha channel)
this is done in all handled frames.

Video/Frames LayerDel

"plug-in-gap-range-layer-del"

Delete Layers in Frames

Start from Menu:

<Image>/Video/Frames Layer Delete

This feature deletes one layer that is specified by its layerstack number where 0 refers to top of the layerstack.

This is done in all handled frame images specified by "From Frame" and "To Frame".

Note:

The "Frames Modify" feature provides a more sophisticated way to pick and delete layers in a specified framerate.

refer to ([plug-in-gap-modify.txt](#))

Video/Frames to Image

"plug-in-gap-range-to-multilayer"

Frames to Image

Start from Menu:

<image>/Video/Frames to Image

Convert frames to one multilayered image.

The selected source range of frames is copied into one new created multilayered destination Image. Each frame results in one layer in the destination image. (or nothing if the source frame has no selected layer)

With the layer basename you can choose a name for the resulting layers in the destination Image. The string [#####] is replaced by the frame number.

Example: my_layer_[##] results in: my_layer_01, my_layer_02

Layer Mergemode:

- Extend as necessary Build a destination layer by merging selected layers of one source frame. The destination layer's size will be

the outline-rectangle of all selected layers.

- Clipped to image Build a destination layer by merging selected layers of one source frame. The destination layer's size will be the image size.
- Clipped to bottom layer Build a destination layer by merging selected layers of one sourceframe. The destination layer's size will be the size of the lowest selected layer.
- Flattened image Build destination layer by flattening the source layers copied from one source frame. There will be no transparent parts in the destination layers. The destination layer's size will be the image size.

With the 'Exclude BG-Layer' checkbutton pressed, the background layers of all the source frames are excluded from the copy, regardless if they are selected or not. Otherwise background layers are handled like all other layers.

(If you are using flatten mode 'Flatten' the background color will fill all transparent parts of the resulting destination layer(s).)

Layer Selection:

Here you can select which layer(s) of a frame is(are) used to build the destination layer.

Select Layer(s):

Pattern is equal to layer name

Pattern is start of layer name

Pattern is end of layer name

Pattern is a part of layer name

With these settings you can select layers by their layername.

(All layers with a layer name matching the select pattern are selected).

Pattern is layerstack number list

Layers are selected by their layerstack position, where 0 is the top layer.

The pattern is a list of layerstack numbers or number ranges. (0-3 matches to the upper 4 layers)

Pattern is reverse-stack list"

Here you can specify layerstack numbers in reverse order. (where 0 is the background layer)

All Visible (ignore pattern)

All visible layers are selected.

(The select pattern is ignored)

Select Pattern:

String to identify a layer. It can be a part of the layer name or a list of layerstack numbers (like this one: "0, 2-5, 9") depending on your choice in select layer(s) above.

Case Sensitive:

Lowercase and uppercase letters are considered as equal if this checkbox is set to off.

Invert Selection:

Select all unselected layers.

Pixel Selection:

How to handle selected areas in the handled frame range.

"Ignore"

Ignore all selections in all handled frames

"Initial Frame"

Use only one selection from the first handled frame, All unselected areas are set to transparent in the resulting multilayer image.

All the resulting layers in the generated multilayer image are shaped by the initial selection.

"Frame specific"

Use selections from all handled frames.

(Frames without any selection are handled as if all pixels were

selected)

All unselected areas are set to transparent in the resulting multilayer image.

Video/Frames Renumber

"plug-in-gap-renumber"

Renumber Frames

Start from Menu:

<Image>/Video/Frames Renumber

This plug-in rennumbers all frames (discfiles) of a video.

First Frame Number:

Enter the new number for the first frame in this entry field.

All frames of video are renumbered, starting at this number.

Digits:

Here you can enter how many digits (1 up to 8) to use for your frame numbers. Framenumbers are filled up with leading zeros to the specified number of digits.

Framenumbers that are greater than the specified number of digits will use more digits than specified, but are built without leading zeroes.

If you choose 1 your framenumbers are built without any leading zeroes at all.

If the frames are numbered in a non-continous way, you can make them continous with this plug-ins.
(Most other GIMP-GAP features do not accept non-continous numbered frames)

Example:

you have the frames:
film_0001.xcf
film_0007.xcf
film_0008.xcf

After running the renumber feature with settings
First Frame Number: 4
Digits: 6

the frames will be renumbered as:

```
film_000004.xcf    # old name was film_0001.xcf
film_000005.xcf    # old name was film_0007.xcf
film_000006.xcf    # old name was film_0008.xcf
```

Video/Frame Sequence Reverse

"plug-in-gap-reverse"

Frame Sequence Reverse

Start from Menu:

<Image>/Video/Frame Sequence Reverse

This feature reverses the specified sequence of frames in the affected range.
The affected range is specified with "From Frame" and "To Frame".

Example:

Assume there are 10 frames:
pic_000001.xcf
pic_000002.xcf

pic_000010.xcf

After applying the Frame Sequence Reverse with settings:

From Frame: 5
To Frame: 8

the affected frames are renumbered as shown here:

old name		new name
pic_000005.xcf	==>	pic_000008.xcf
pic_000006.xcf	==>	pic_000007.xcf
pic_000007.xcf	==>	pic_000006.xcf
pic_000008.xcf	==>	pic_000005.xcf

Video/Frame Sequence Shift

"plug-in-gap-shift"

Frame Sequence Shift

Start from Menu:

<Image>/Video/Frame Sequence Shift

This feature does cyclic reordering the sequence of the frames in the affected range. The affected range is specified with "From Frame" and "To Frame".

The "N-Shift" specifies positive or negative offsets for the number part in the frame names.

Example:

Assume there are 20 frames:

```
pic_000001.xcf
pic_000002.xcf

pic_000020.xcf
```

After applying the Frame Sequence Shift" with settings:

```
From Frame: 5
To Frame:   15
N-Shift:    2
```

the affected frames are renumbered as shown here:

old name		new name
pic_000005.xcf	==>	pic_000007.xcf
pic_000006.xcf	==>	pic_000008.xcf
pic_000007.xcf	==>	pic_000009.xcf
pic_000008.xcf	==>	pic_000010.xcf
pic_000009.xcf	==>	pic_000011.xcf
pic_000010.xcf	==>	pic_000012.xcf
pic_000011.xcf	==>	pic_000013.xcf
pic_000012.xcf	==>	pic_000014.xcf

```
pic_000013.xcf ==> pic_000015.xcf
pic_000014.xcf ==> pic_000005.xcf
pic_000015.xcf ==> pic_000006.xcf
```

Video/Split Img to Frames

"plug-in-gap-split"

Split Image into Frames

Start from Menu:

<Image>/Video/Split Img to Frames

This feature splits a multilayer image into frames. Usually each one of the layers will be stored as separate frame image on disk. The name of the frame images on disk is built from the image name, the framenumbers and the extension.

Note:

This feature does not operate on images that have already a numberpart in the imagename. If you want to split such images, you should make a duplicate (Menu: <Image>/Image/Duplicate) or rename the image and then try again on the duplicate (or the renamed image)

Extension:

The extension defines the last part of the resulting framenames. Further it defines the image fileformat. Normally .xcf is used to store the frames in GIMP's native fileformat XCF.

Inverse Order:

ON: start with frame 000001 at top layer.
OFF: start with frame 000001 at background layer.

Flatten:

ON: Flatten all resulting frames. Flattening replaces transparent parts by the background color. (flattening removes the alpha channel that is normally used to store transparency)
OFF: keep layers 1:1 as they are in the original image.

Only Visible:

ON: affect only the visible layers and ignore invisible layers.
OFF: affect all layers.

Digits:

How many digits to use for the numberpart of the resulting frames.

Video/Storyboard

"plug-in-gap-storyboard-edit"

Storyboard Dialog

Start from Menu:

<Image>/Video/Storyboard (has HelpMenu, no Button)

A storyboard file is a textfile that can be used to assemble many videoclips, single images and frame sequences to one resulting video.

The file contains lists of references to both video and audio sourcefiles and is organized in tracks.

(see [STORYBOARD_FILE_DOC.txt](#) for more information about storyboard file syntax and usage)

The storyboard dialog is a visual tool that allows creation and editing of simple storyboard files.

the storyboard editor window is organized into 3 regions

- cliplist
- storyboard
- playback preview

Cliplist and storyboard have the same capabilities to edit storyboard files. The general idea is to have 2 files open at the same time, where the cliplist usually acts as collection of available clips, and the storyboard is the destination where the resulting video is assembled.

Copy, cut and paste of Clips between cliplist and storyboard is supported. There are corresponding buttons and menu entries both for the storyboard and the cliplist.

There are playback buttons in the storyboard and the cliplist. Those buttons play the entire list in the playback motion preview. When the SHIFT key is pressed while those buttons are clicked, the playback is restricted to the selected clips of the list.

Creating new storyboard file and clips:

- 1.) Use the menu item "New" in the storyboard menu to create a new storyboard file.

This opens the popup dialog window for the storyboard master properties, where you specify a filename, size in pixels and framerate settings.

Decoder: here you can specify preferred decoder library (libavformat or libmeg3) for videofile read access.

(you may change those properties later any time too, using storyboard menu "Properties")

1b.) Creation of a storyboard file can also be done via drag&drop. Drag a clip (image, frames or videofile) from another application (e.g. a filemanager supporting Drag&Drop) and drop onto one the button row. (any of the buttons can act as drop destination in the same way) This will add the dropped clips at end of an already existing storyboard (or cliplist) but also automatically create a new storyboard (or cliplist) in case if
if the destination is NOT already existing.

2.) Create a new clip.

Use the storyboard menu "Create Clip" to add a new clip. This opens the "Clip Properties" popup dialog window, where you can specify a filename.

The Clip Type is detected automatically (depending on the file) as:

MOVIE

FRAME-IMAGES

SINGLE-IMAGE

refer to description of the "Clip Properties" popup dialog window.

([plug-in-gap-storyboard-clip-prop.txt](#))

If you want to add clips that are parts of videos or frameranges that are already used in the storyboard you can select the clip (start and end frame) in the playback motion preview and add it from there by pressing the buttons 'From Frame' or 'To Frame'

2b.) Create a new clip via Drag&Drop.

Select one or more frames, single images or video clips in another application (e.g. a filemanager supporting Drag&Drop) and drag them with the mouse onto a thumbnail of an existing clip in the storyboard or cliplist window and drop them via releasing the mousebutton.

If the mouse pointer is within the left half of the destination thumbnail, the dropped clips will be inserted before the destination thumbnail. If you drop them over the right half they are inserted after the destination thumbnail.

Following functions on the thumbnails in the storyboard and cliplist can be activated via mouse clicks:

Single Click on a clip thumbnail toggles selection state of the clip.

All other clips are deselected.

With the modifier Keys CTRL and SHIFT you can keep and extend existing selections.

Double click on a clip thumbnail starts the playback of this clip.

Click with the right mousebutton on a clip thumbnail to open the properties popup dialog for this clip.

The Playback Preview

This area of the window provides a motion preview for a single clip or for parts or the entire clip sequence of the storyboard.

The headline of the playback area shows

```
"STB: <name of the storyboard file>"
  to indicate storyboard playback, the Prefix changes to
```

```
"VIDEO:" or
"FRAMES:"
```

when single clips are played. Depending on the type of the clip, that may be a sequence of frames or a selected range out of an already encoded videofile.

Creating Clips via Playback

Select the desired range of frame in the spinbutton widgets right next to the "From Frame" and "To Frame" buttons in the Playback preview. Then press "From Frame:" button to add the selected range as new clip.

Pressing the "To Frame:" button adds the clip with inverse range.

new clips are added after the last clip in the cliplist. (if no cliplist is opened, or the storyboard is set as default clip target, the clip will be added to the storyboard)

Restrictions: Creating Clips via Playback
is not supported when playback is in storyboard mode.
(STB: name of the storyboard file)

For more details on the playback related widgets see the chapter

refer to Player description ([plug-in-gap-videoframes-player.txt](#))

Creating clips via automatic scene change detection.

You may add the full range of a longer video as one single clip and then automatically split this long clip into more short clips. Therefore click on the thumbnail of the long clip to open the Clip Properties dialog.

In the Clip Properties dialog you find the button "Auto Scene Split". This will scann the selected range, and create new clips on every significant scene change.

```
### refer to chapter Clip Properties dialog #####
## (plug-in-gap-storyboard-clip-prop.txt) ##
```

The Global Menu:

- Properties

Opens a popup dialog window where you can specify layout properties (e.g. thumbnailsize, rows and columns) of the storyboard editor main window.

- Videothumbnails
Render videothumbnails when checked.
Videothumbnails are built at runtime in memory for all frames that are startframes of clips from type Movie.
Using Videoindexes is strongly recommended for fast frame access.
Thumbnails for frame images are not affected here.
- Exit
Close the storyboard edit window.
This will close both storyboard and cliplist.

The Cliplist and Storyboard Menu
(Both Menus include the same features)

- New
Create a new storyboard file. This opens a popup dialog where you can enter filename and properties of the new storyboard file.
- Open
Open an existing storyboard file.
- Save
Save the storyboard (or cliplist) as textfile to disc.an existing storyboard file.
- Save As
Save the storyboard (or cliplist) as new file.
- Playback
Playback motion preview of entire
- Properties
Open the properties pop-up dialog window, where you can specify master properties (e.g. master size and framerate) of the storyboard (cliplist)
- Create Clip
Create a new clip. This opens a popup dialog where you can enter the properties of the clip.
- Toggle Unit
Defines the small description text below the thumbnail of a clip.
The toggle changes between the following settings:
 - Clip Number only
 - Clip Number + Framenumber
 - Clip Number + Timestamp (mm:ss:msec)
- Add Audio Otone
This feature generates an audiotrack in the storyboard. This audiotrack adds the audio original tone references for all the videoclips of type MOVIE that are found in the storyboard.
(except those clips that are played backwards, pingpong mode)

or at other stepsize than 1.0)

A pop-up dialog is opened where you can specify:

Input Audiotrack:

This refers to the audiotrack number of (all used) inputvideoclips.

Most videofiles have just one track.

DVD's typically store multiple language versions on multiple audotracks in their .vob files.

Output Audiotrack:

This is the storyboard specific audiotrack number.

Replace Audiotrack

This flag allows replace of the specified audiotrack in the storyboard.

Note:

the generated audiotrack is just a list of references to the audiopart of the used videoclips.

Processing of this list is done as audio preprocessing step at encoding in the Master Videoencoder.

(see Master Videoencoder, Create Composite Audio for more information)

refer to chapter [Master Videoencoder](#) dialog

- Encode
Call the master videoencoder dialog for storyboard based encoding.
(this menu entry is insensitive if there are unsaved changes)
- Close
Close the storyboard (Clist).

Restrictions:

In GIMP-GAP release 2.4 the storyboard dialog does not support the full storyboard syntax.

- audiotracks are not supported.

Keep in mind that the backend processing can already handle all specified storyboard features.

If they are added via a text editor

all features inclusive audio processing are supported for rendering the resulting video.

Master Properties (Storyboard)

"plug-in-gap-storyboard-master-prop"

Master Properties (Storyboard)

The master properties dialog window is used to create new storyboard files and to view and update properties of an existing storyboard file. It is a popup dialog of the storyboard editor

refer to Storyboard Dialog
([plug-in-gap-storyboard-edit.txt](#))

File:

The name of the storyboard file.
If the dialog was opened via menu "properties" the filename is read only information.
If invoked via "new" you can specify the filename here and the button "..." opens a filebrowser dialog window where you can select the filename.

Width:

Height:

Master size of the target videofile in pixels.
At encoding time all input frames are scaled to this master size.
(For manually edited storyboard files there you can control how to scale the input frames individually per clip using records of type VID_FIT_SIZE see storyboard syntax description for more details.
[STORYBOARD_FILE_DOC.txt](#)

Framerate:

Master framerate of the resulting videofile in frames per second.

Decoder:

Selection of a preferred video decoder software library.
If nothing is specified the decoder is selected automatically.
The decoder setting is only relevant for the clip references to videofiles, to specify what software to use for readaccess to the videofile.

Samplerate:

Master samplerate in 1000 samples per second.
The master samplerate is the target samplerate for the resulting audiotrack when storyboard file is encoded to a videofile.
GIMP-GAP creates this track in a pre-processing step (in the Master Videoencoder module) as composite audiofile by mixing down all audiotracks of a storyboard file.

Volume:

Master volume of the resulting audiotrack, where values greater than 1.0 will amplify the volume. (with the risk of producing noise on overflow)

AreaFormat:

This format string triggers automatical logo insertion for all handled clips of type MOVIE.

The area format string shall contain the placeholder %s that is replaced by the basename of the currently processed videoclip. The placeholder %06d is replaced by the current framenummer.

The storyboard processing builds the filename of a logo image whenever a frame is fetched from a movie clip and pastes the logo into the frame in case the logo image exists. If the format does not contain any placeholder, the same logo will be used in all handled movie clips.

Frame specific Example:

```
AreaFormat: /logo_frames/%s/logo_frame_%06d.xcf
```

```
Processing of frame 7 of movie clip /videos/MY_VIDEO.AVI
will paste logo image /logo_frames/MY_VIDEO.AVI/logo_frame_000007.xcf
```

```
Processing of frame 22 of movie clip /videos/YOUR_VIDEO.MPEG
will paste logo image /logo_frames/YOUR_VIDEO.MPEG/logo_frame_000022.xcf
```

Clip Properties (Storyboard)

"plug-in-gap-storyboard-clip-prop"

Clip Properties (Storyboard)

The clip properties dialog window is used to create new clips and to view and update properties of existing clips in a storyboard file.

It is a popup dialog of the storyboard editor

```
## refer to Storyboard Dialog
## (plug-in-gap-storyboard-edit.txt)
```

File:

Clips use a filename to refer to a videofile or to a frame imagefile on disk.

The button "... " opens a filebrowser dialog window where you can select the filename.

The Clip Type is detected automatically (depending on the file) as:

```
MOVIE
FRAME-IMAGES
SINGLE-IMAGE
```

From:**To:**

Here you can enter the selected frame range for the clip. (those widgets are disabled when file referes to a single image)

Loops:

Here you can specify the number of loops, where a value of 2 shows the selected range of frames two times in a loop.

Pingpong:

The Pingpong checkbutton plays the selected range forward and backwards.

If the selected range is from 1 to 5
the clip is played as sequence 1, 2, 3, 4, 5, 4, 3, 2

Stepsize:

With the stepsize density you can affect the playback speed individual per clip without changing the master framerate of the storyboard file.

Values greater than one results in skipping input frames and will speed up the clip.

Values less than one does show some frames twice and does slow down the clip.

For normal playback of the clip use stepsize value 1.0.

Videotrack:

Here you can select the number of the input videotrack.

Most videos have just one track.

Some DVD videos have multiple angle view where the different angles are represented by 2 or more videotracks.

Note:

the videotrack is only available for clip type MOVIE.

Deinterlace:

Deinterlacing is selected via radio buttons.

Fetching frames out of already encoded videofiles has a built in deinterlace filter.

Interlaced frames are built up of 2 half pictures, where the even lines build the one half and the odd lines build the other half picture.

The deinterlace filter builds the extracted frame by picking only the lines of one of the half pictures and replaceng the lines of the other half picture by more or less smooth mixing the upper and lower neighbour lines.

The mix is controlled with the spinbutton value where 1.0 is a smooth mix, and 0.0 takes just one of the neighbour lines without mixing.

That way the deinterlaced frames have same height as the originals.

The deinterlace radio buttons are:

"None"

Turn off the deinterlace filter and read frames 1:1 as they are found in the videofile.

"Odd"

Extract the half-picture that is represented by odd lines

"Even"

Extract the half-picture that is represented by even lines

Note:

the deinterlace filter is only available for clip type MOVIE.

Transform:

Provides frame orientation transformations that are applied on frames of this clip.

The transform radio buttons are:

"None"

keep the original frame orientation

"Rotate 180"

rotate the frames of this clip by 180 degree

"Flip Horizontally"

horizontally mirror the frames of this clip

"Flip Vertically"

vertically mirror the frames of this clip

Duration:

Information about the duration of the clip in frames and playtime in format mm:ss:msec

Mask:

Individual transparency for each frame of the clip can be added by selecting a mask.

Important:

You must define (at least one) Mask clip in the Mask Section of the Storyboard before you can assign a mask to a clip.

The mask can be assigned to clip or master.

o) Clip ... the mask is scaled to the size of the fetched frame of the clip and attached as additional transparency before transitions (move and scale) are applied. Eg. the mask moves and scales with the clip.

o) Master ... the mask is scaled to the size of the resulting frame and attached after transitions (move and scale) were already applied. E.g. the mask is fixed in size and position.

Mask Stepsize:

In case the assigned mask clip is an animation (and not a single image), the mask stepsize controls how to advance to the next mask frame. Where a value of 2.0 speeds up the mask effect (by skipping

every 2nd frame in the assigned mask clip)
 A value of 0.5 will slow down the mask effect.
 Use value 1.0 for normal speed for mask effect.

Filtermacro:

Here you can specify the filename of a GIMP-GAP filtermacro that shall be applied for each frame of this clip.

see [plug-in-filter-macro.txt](#) how to record filtermacros and [STORYBOARD_FILE_DOC.txt](#) chapter Macrofiles for more details.

Filtermacro Steps:

Here you can enter the duration in number of frames for the transition of settings from filtermacro to settings of a 2nd filtermacro (in case a 2nd filtermacro is available)

Use value 1 to apply just constant settings of the 1st filtermacro (if a 2nd filtermacro is available it will be ignored and its name will be displayed with the prefix "OFF")

Note that filtermacro effects are not automatically visible in standard playback of the Storyboard dialog. You must use the playback based on full rendering to preview those effects.

Filtermacro2:

For applying filtermacros with varying values you need a 2nd filtermacro file.

The 2nd filtermacro file is automatically identified via naming convention and can NOT be specified explicite. Its name is displayed read only when such a file is available, and prefixed with "ON" when active or "OFF" when disabled (e.g duration of 1 frame is specified)

see [STORYBOARD_FILE_DOC.txt](#) chapter Macrofiles for details about applying filtermacros with varying values.

Acceleration Characteristic for applying filtermacro with varying values:

A Spinbutton and a graph widget

is available to enter and display the acceleration characteristic value

The graph next to the spinbutton shows the selected acceleration

characteristic curve.

The acceleration characteristic can also be changed by clicking on the graph and dragging vertically with the mouse.

A straight line from left bottom to right top corner of the graph is drawn for value 1 that represents constant speed. other positive values result in increasing speed of the transition, negative values result in decreasing speed.

Comment:

An optional comment text on the clip.

Find Scene End

This button checks the frames in the defined range

between "From" and "To" for significant changes and sets the "To" value to the first detected scene change.

Auto Scene Split

The button "Auto Scene Split" will scan all the frames in the selected range and create new clips on every significant scene change. The new clips are added at end of the storyboard. (or cliplist)
You may stop the scan any time with the "Close" button.

Transition Attributes (Storyboard)

"plug-in-gap-storyboard-clip-prop"

Transition Attributes (Storyboard)

The transition attributes dialog window is used to control transitions that shall be performed on the following clips in the same storyboard video track.
It is a popup dialog of the storyboard editor

refer to Storyboard Dialog
([plug-in-gap-storyboard-edit.txt](#))

supported transitions are:

- Opacity changes
- Moves related to the center position 0/0 (Scrolling)
- Scaling (Zoom)

The dialog has 2 previews:

The left one shows the transition effect

start settings applied on the first affected frame.

The right one shows the transition end settings applied on the last affected frame.

The previews show the size of the resulting video (master size) in a centered green frame and the environment around the master size with solid grey background.

Note: the last affected frame can be different for each single transition. In this case the preview shows only one frame as 'last affected' using the first enabled transition.

FitSize:

Width:

Enabled: Scale width of input clips to fit into the resulting master width.

Height

Enabled: Scale height of input clips to fit into the resulting master

height.

Keep Proportion

Enabled: Keep proportions of the input frame at scaling.

Opacity:

Settings for the opacity where 0 is fully transparent and 100 is fully opaque.

Move X:

Scroll settings, where 0.0 is horizontal centered, 100.0 moves the frame entirely outside the right border of the master width.

Move Y:

Scroll settings, where 0.0 is vertical centered, 100.0 moves the frame entirely outside the lower border of the master height.

Scale Width:

Zoom settings, where 100.0 specifies no scaling, 200.0 is scaling to double width, 50.0 is scaling down to half width.

Scale Height:

Zoom settings, where 100.0 specifies no scaling, 200.0 is scaling to double height, 50.0 is scaling down to half height.

Frames:

Duration of the transition effect measured in frames.

Acceleration Characteristic:

A Spinbutton and a graph widget is available to enter and display the acceleration characteristic value in each row (for Opacity, MoveX, MoveY, ScaleWidth and ScaleHeight)

The graph next to the spinbutton shows the selected acceleration characteristic curve.

The acceleration characteristic can also be changed by clicking on the graph and dragging vertically with the mouse.

A straight line from left bottom to right top corner of the graph is drawn for value 1 that represents constant speed. other positive values result in increasing speed of the transition, negative values result in decreasing speed.

Comment:

An optional comment text on the transition.

Reset

This button resets all settings to the values at opening time of this dialog window.

Generate original tone audiotrack (Storyboard)

"plug-in-gap-storyboard-gen-otone"

Generate original tone audiotrack (Storyboard)

This feature generates an audiotrack in the storyboard. This audiotrack adds the audio original tone references

for all the videoclips of type MOVIE that are found
in the storyboard.
(except those clips that are played backwards, pingpong mode
or at other stepsize than 1.0)

It is a popup dialog of the storyboard editor.

refer to Storyboard Dialog
([plug-in-gap-storyboard-edit.txt](#))

File:

Input Audiobook:

This refers to the audiobook number of (all used) inputvideoclips.
Most videofiles have just one track.
DVD's typically store multiple language versions on
multiple audiotracks in their .vob files.

Output Audiobook:

This is the storyboard specific audiobook number.

Replace Audiobook

This flag allows replace of the specified audiobook
in the storyboard.

Notes:

the generated audiobook is just a list of references
to the audiobook part of the used videoclips.
Processing of this list is done as audio preprocessing step
at encoding in the Master Videoencoder.
(see Master Videoencoder, Create Composite Audio for more
information)

refer to chapter [Master Videoencoder](#) dialog

Storyboard Player

"plug-in-gap-storyboard-player"

Player invoked from storyboard dialog editor plug-in

(the player hides the HELP Button in case when
invoked from the storyboard editor dialog,
because the player is part of the storyboard editor window
and has its own help menu item)

For more details on the playback related widgets
see the chapter

refer to Player description ([plug-in-gap-videoframes-player.txt](#))

Video/Playback

"plug-in-gap-videoframes-player"

Playback

Start from Menu:

<Image>/Video/Playback...

This player does playback of your video frames.
 Along with the video frames you can also do playback of an audiofile.
 The player is a tool that should help to give a motionpreview
 and allows visible positioning for the animation designer.

There is NO direct support for playback of videoformats (like MPEG, AVI,)
 Internally the player can read the frames (but no audio) from videofiles
 when invoked from the video extract dialog or from the storyboard dialog.

Video Options:

- Play (>) button
 Start the playback,
 or switch to forward play if already playing reversed.
 - PlayReverse (<) button
 Start the reversed playback,
 or switch to reversed play if already playing forward.
- SHIFT click on the "Play" or "PlayReverse" button creates
 a snapshot image and adds all the frames that are played
 as new layers to this image to trace the playback.
 (see chapter Creating a snapshot image below)
- Pause button (||)
 Stop playback,
 or go to first frame of the selected frame range if not playing.
 Go to the end of selected frame range if right mousebutton was used.
 or go to the active frame number if the middle mouse button was used.
 (the active frame number is the one that is currently
 loaded in the GIMP image window from where you invoked the player)
 - The GO button array
 the 50 blank buttons above the preview build up the GO button array,
 This tool is for positioning and manual speedcontrol.
- Move the mousepointer over those buttons to display
 frames in sequence as fast as you move the mouse.
 Moving right is forward, left is backward.

Click on one of the GO-buttons loads the current frame
 into the invoking frame image.

CTRL-Click sets the current frame as start of the range,
ALT-Click sets the current frame as end of the range.

Tip:

The positioning is done relative to the point where the mouse entered the GO button array.
If you reached the leftmost GO button at framennr 50 and want to go more backwards (to framennr 35), just move the Mouse outside the GO button array (up or down) and then enter again from the right side, and continue moving to the left, but now keeping the mouse inside the GO button array until the desired frame (Nr. 35) is reached.

Tip: For efficient use of the GO buttons the "Use Thumbnails" checkbox should be turned ON.
Further all of the frames should have thumbnail files.

Even fast machines may have problems to deliver full sized multilayer frames in time.
(depending on size of the frames, number of layers and IO performance of your hardware)

- "Frame" "Nr" buttons and spinbutton

The current framenummer position is shown in the spinbutton right to those buttons.

The timepos: mm:ss:sec displays below shows the same current position as Minutes:Seconds:Microseconds

(The time calculation is always based on the original speed.
This is the framerate of your animation as found in the _vin.gap file.
You can change the framerate in the VCR Navigator window)

A mouseclick on the "Frame" button sets the start of the range to the current position.

A click on the "Nr" button sets the end of the range to the current position.

You can load the frame (the number that is currently on the players display) into the active image window when you SHIFT click on the "Frame" or on the "Nr" button.

Alternative you can also click on one of the GO buttons, or on the players display to reload the active image with the desired frame.

- "From Frame" and "Frame To" buttons and spinbuttons:

Here you can select a framerange by entering the begin and end framenummer.

The buttons to the left of the spinbuttons print

out the description of the selected range in storyboard file syntax notation.

Note:

When the player is invoked via the storyboard editor (it usually becomes part of the storyboard editors dialog window) you can add the selected range as new clip at end of the cliplist by pressing those "From Frame" or "To Frame" buttons, where "To Frame" means backward playback of the clip.

- Speed button
 - Reset Speed to the original (the framerate of your Animation)
 - (a 2.nd click goes back to the previous enters Speed)
- Speed spinbutton
 - Here you can enter the desired playback speed.
 - The available range is 1 upto 250 frames/sec.

 - Please note that 250 frames/sec may not be reached on most machines.
 - If the exact_timing checkbutton is ON, the 250.th frame will be reached in time (after 1 second), but if your machine can handle only 25 frames in one second you will see just frame 10, 20, 30, 40, , 230, 240, 250
- Size button
 - Set the standard thumbail size
 - There are two standard sizes
 - "normal" 128 Pixels
 - "large" 256 Pixels
 - this button toggles between both sizes.
 - If you hold the SHIFT key wihle clicking the size button, the preview is scaled to 1:1 original framesize.
- Loop checkbutton:
 - Plays in loop if ON, otherwise play once and then stop.
- Selection only
 - Play only frames within the selected range.
 - if not checked (OFF) play all frames.
- PingPong
 - Play alternating backward/forward
- Use Thumbnails
 - use thumbnail files if available.
 - Both the PNG based open thumbnails standard and the older .xvpics based thumbnail files are supported.

 - Thumbnail based playback has reduced quality.
 - If there are just .xvpics thumbnail files the quality is very miserable.

The full original frame images are read in case when no thumbnail file is available,

(**Note:**

It also does create the thumbnail file in such a case, but this is only done, if the preferences setting permits creation of normal (or large) thumbnail files)

No thumbnail data is written when the playback reads the frames directly from a videofile.

)

If the "Use Thumbnails" checkbutton is switched OFF the player always reads the full original sized frame images

Playback with full frame image read mode is very slow, even on fast machines, but gives the original quality.

- Exact Timing

if this checkbutton is ON (the checked state) the player tries to keep the frametiming as exact as possible. if your machine is not fast enough one or more frames of the sequence are not displayed (dropped) and the next frame is displayed in time.

if this checkbutton is OFF all frames are displayed. this leads to more and more growing delay depending on the the power of your machine.

Creating a snapshot image:

A click on the displayed frame creates an image and adds the current frame as new layer to this snapshot image. The size of the snapshot image depends on the "Use thumbnails" checkbutton.
OFF: Original framesize is used.
ON: thumbnail size is used.

if the snapshot image is already there (on subsequent clicks on the displayed frame) the displayed frame is added to the top of the layerstack in this snapshot image. (it will be scaled to snapshot image size if necessary).

you can close the snapshot image any time you want, to force recreation at next time.

The snapshot image can also be created and automatically filled with layers that are snapshots of all the frames that are shown while playback is running. This motion snapshot tracing feature is activated when you hold the SHIFT key while you click on the "Play" (>) or "PlayReverse" (<) button.

Warning:

It is recommended to turn on the "Play selection only" checkbox, and to turn off both the "Loop" checkbox and the "Exact timing" checkbox in that case.

Exact timing would skip most of the frames, and loop produces endless number of frames and may lead to strange behaviour or crashes when no more memory is available.

Note: this feature is also available when the player is invoked from the video extract plug-in or from the storyboard editor dialog and may then be used to pick images out of videofiles.

Audio Options:

Note: The audio options are not existent in the WINDOWS version (eg. when compiled with --disable-audiosupport.)

- Audiofile:

Here you can specify the audiofile. (you may also use the "FileBrowser" button)

- Volume:

*** Volume control is not implemented yet ***

The "Enable" checkbox can turn ON/OFF audio playback.

- Offset

Here you can shift the audiostart against the video startframe. An offset value 0 will start audio at the first frame. Negative offsets will start audio with a delay of n frames. Positive offsets will skip the beginning of the audiofile. The skipped duration is equal to the time it takes to play n frames at original framerate.

The "Offset" label shows the audio shift in unit Minutes:Secs:Millisecs. The "Audiotime" label shows the total playing time of the audiofile (without taking care of the offset shifts)

The "Audioframes" label shows the total playing time of the audiofile in unit frames where frames are played at the original framerate of the video frames.

The other labels show informations about the audiofile.

- Nuber of Samples
- Samplerate (in Samples / Sec)
- Channles (usual 2 for stereo)
- Bits/Sample

If no valid audiofile was specified, all values are filled with 0.

- Copy As Wavfile

This button provides conversion of audiofile formats to RIFF WAV. It opens a subdialog where the name of the wavfile can be entered and where you can optional specify a new samplerate if you want

to resample down to lower samplerates.

IMPORTANT NOTE:

- You need the external programs sox and lame to run that feature, because the provided converter script `audioconvert_to_wav.sh` is based on that programs.
- If you have another external audio converter program, you can configure GIMP-GAP to use that program rather than the standard converter script `audioconvert_to_wav.sh`

The preferred way is to edit your `gimprc` file and add a line like this on:
`(audioconvert_program "/usr/local/bin/my_private_audioconverter")`

The 2nd way is to set an Environmentvariable:
`export AUDIOCONVERT_TO_WAV=/usr/local/bin/my_private_audioconverter`

Some notes on the audiosupport:

- The audio playback checks for the existence of the `wavplay` audioserver executable file in your `gimprc` file
`(wavplaypath "/usr/X11R6/bin/wavplay")`
then checks the environment variable
`WAVPLAYPATH=/usr/X11R6/bin/wavplay`
and finally searches an executable named `wavplay` in all directories defined by your `PATH` environment variable.
- Audiosupport can be completely turned off at compiletime. If Audiosupport was enabled at compiletime, The playback dialog window contains a notebook widget tab named "Audio Options"
- Audio is always turned off, while frames are played backwards.
- The audioplayback samplerate tries to follow the videospeed. If videoframes are played at the original speed, audio is played at the normal samplerate. If video speed is too fast audio may not follow synchron.
- If the "Exact Timing" checkbox is off (unchecked) video and audio will not play synchron !

Tip:

If you want to create thumbnails from frames under visual control you can use the player to do that. Therefore you have to enable image thumbnails in the general GIMP preferences environment settings.

Then simply playback from 1.st until last frame with "Use thumbnails" checkbutton turned on and "exact timing" checkbutton turned off. This creates the thumbnails while the frames are displayed.

Video/Layer/Attributes/

[NEW] Plug-in-layer-set-alpha-by-colormask

"plug-in-colormask"

Colormask Filter

Start from Menu:

<Image>/Video/Layer/Attributes/Apply Colormask

The colormask filter can apply transparency for pixels matching the colormask image.

It is intended for processing frames where moving objects can be isolated from the non-moving constant background by applying the constant background as colormask on all frames.

The colormask provides more flexibility than the bluebox filter because it can operate on frames where the background is not just made of shades

of the same color.

But for good results the moving object shall not contain the same colors as the background.

In case the colormask layer has an alpha channel it is used to define protected areas, e.g the alpha channel of the colormask controls where the filter shall be applied (opaque) or not (transparent pixels are protected).

Pixels where the corresponding alpha channel in the colormask is below the specified

triggerAlpha value are not affected by the colormask filter.

This is useful when the colormask filter runs non-interactive

under control of storyboard render processing

on frame images that are created internally where the image has no selection.

The colormask filter is not yet fully implemented.

TODO: add more detailed feature description here when implementation is finished.

Plug-in-foreground-extract-matting

"plug-in-foreground-extract-matting"

Foreground Extract Filter

Start from Menu:

<Image>/Layer/Transparency/Foreground Extract

<Image>/Video/Layer/Attributes/Foreground Extract

How to use:

Have a image with a foreground object loaded in GIMP
(for instance a person) that shall be
separated from the background.
The image shall be of RGB color

Add a layermask to the layer that shall be processed
and fill the layermask with a medium Gray color
now the complete layer looks semi-transparent.

Paint on the layermask with WHITE color to mark foreground areas.
the painted areas appear fully opaque.

Paint on the layermask with BLACK color to mark background areas.
the painted areas disappear, e.g get fully transparent.

Take care that you DO NOT mark background areas as foreground
and vice versa. (e.g. your brush strokes
shall not overlap the borders of the foreground object.

When finished with that (rough) selection call the
Foreground Extract Filter
select "Layermask" in the TRI-MAP combo box

use unchecked state for the checkbuttons
Create Layermask and Lock Colors

This will create a new layer as copy of your input layer
with calculated transparency for the UNDEFINED areas.
This copy is placed above your input layer.

Set your original layer invisible
to check the result.
(click the Eye-Icon in the GIMP Layers dialog
to toggle visibility)

Note that the quality of the result depends on the provided TRI-MAP.
In many cases you can improve the results by tuning the TRI-MAP
on those areas where the filter algorithm did not deliver acceptable results
and run the filter again.

For good results approximately 70% of the area shall be marked
as either FORGROUND or BACKGROUND. Further note that processing time
depends on the size of the UNDEFINED areas.

Options:

Create Layermask

ON: Render opacity by creating a new layer mask for the resulting layer
OFF: Apply rendered opacity to the alpha channel

Lock Colors

ON: Keep RGB channels of the input layer
OFF: Allow Background color removal in processed undefined regions

How it Works:

The foreground extract filter calculates opacity and color based on a TRI-MAP.

The TRI-MAP is a special kind of user selection where the user marks areas as FOREGROUND (white)

BACKGROUND (black) and UNDEFINED (gray). The filter can calculate opacity and modify color for the

UNDEFINED areas in the corresponding layer.

Typically a layermask will be used as TRI-MAP.

Optionally you may select another layer (with same size) as TRI-MAP input.

Note that RGB TR-MAP input is implicate converted to GRAYSCALE representation where values ≥ 240 are considered as FOREGROUND, value 0 marks BACKGROUND and all other

values are considered as UNDEFINED.

In case the processed layer already has an alpha channel, fully transparent areas are also marked as BACKGROUND. (e.g. fully transparent pixels keep their transparency

and overrule the value in the corresponding TRI-MAP pixel (even if the TRI-MAP marks those pixels as foreground)

The foreground extraction is based on the alpha matting algorithm.

(http://www.alphamatting.com/eval_25.php)

Whats the difference to the foreground extraction tool?

The old algorithm, implemented after an older paper (<http://www.siox.org/>), has the main problem that it generates only a binary alpha matte.

The problems with that is that you can have severe "color bleeding" of the background color

into the foreground at borders. For example, fine black hair on a green background gets darkish green.

Also, you don't have any transparency or antialiasing at borders, each pixel is either foreground or background.

The new algorithm, on the other hand, tries to find out the foreground AND the background color of each pixel,

and with that, it can calculate a semitransparent alpha value.

This also means that the background color can be removed from the final extracted pixel,

what gives, in general, much better results.

Video/Encode/Master Videoencoder

"plug-in-gap-vid-encode-master"

Master Videoencoder

Start from Menu:

<Image>/Video/Encode/Master Encoder

The master videoencoder dialog provides videofile encoding features for:

- Frame sequences
- Multilayer Images
- Storyboard files

The master videoencoder is the dialog window from where all available GIMP-GAP video encoders are started. GIMP-GAP video encoders are implemented as GIMP plug-ins.

If GIMP-GAP-2.2 was configured fully and was successfully built it provides the FFMPEG encoder, the AVI1 encoder and the SINGLEFRAMES encoder.

Output

In the "Video" entry you specify the filename for the the videofile that should be written.

Video Options:

Input Mode radio buttons

here you can select the type of input that should be encoded as videofile.

Frames

Encode a Framesequence, where one of the frames is the current image where the master video encoder ws invoked from.

Layers

Encode layers of the current image.

Storyboard

Encode clips that are defined via a storyboard file. Therefore the name of the storyboard file must be specified (in the "Extras" tab)

If the master videoencoder is started from the storyboard editor, the name of the storyboard_file is already selected.

Important:

=====

If the storyboard describes audio tracks it is required to perform the audio mixdown processing to create a composite audiofile. This composite audiofile can be passed as audio input to the video encoder plug-ins.

The video-encoder plug-ins are accepting only one simple RIFF-WAV file as audio input and ignore all audio records in the storyboard file. Therefore all audio mixing and converting tasks are handled in the Master Videoencoder itself.

(see description Extras Tab below in this chapter,

Storyboard Audio: Button "Create Composite Audio")

From Frame (Layer/Storyframe)

To Frame (Layer/Storyframe)

with these spinbuttons you can specify the range of frames (or layers or storyboard frames) that should be included in the resulting video.

Depending on the input mode the name changes to "Frame", "Layer" or "Storyframe".

Width and

Height

with these spinbuttons you can specify the size of the resulting videoframes in pixels.

The combo box on the right provides some commonly used videosizes as preset value pairs for both the width and the height widgets.

Framerate

Here you can specify the framerate in frames per second.

The framerate controls the playback speed of the resulting videofile.

The combo box on the right provides some common used framerate values.

Videonorm combobox.

Set the videonorm for the resulting video.

NTSC

PAL

Encoder "Parameters"

The encoder parameters button invokes the encoder specific parameter popup dialog window.

Encoder combo box

Here you can select one of the registered videoencoder plug-ins.

Audio Options:

Audiofile:

Specify the name of an audiofile.

- Files in the RIFF WAVE format are preferred, but you may use other audiofile formats if you have installed sox or configured another external audio converter program.

- Encoding of multiple audio tracks:

Instead of an audiofile you can specify an audio playlist file. This is a textfile that contains filenames of audifiles, one filename per line. The filenames can include absolute path or may use relative path that is relative to the audio playlist file. Empty lines and comment lines (starting with #) are allowed.

For each of this audiofiles the encoder will create a separated audio track.

Restrictions:

- The selected video encoder and selected output videofile format must support multiple audio tracks. Currently only the FFMPEG based encoder provides this support for the most videoformats (.mpg .vob, .avi ...)
- Each of those filenames must refer to an audiofile in the RIFF WAVE fileformat. (automatic convert to RIFF WAVE is not supported inside of audio playlist files)
- The audio samplerate in all referenced audiofiles must exactly match the desired samplerate.
- The samples must be encoded with 16 bits per sample in all referenced audiofiles.

Samplerate:

Here you can specify the audio target samplerate that should be used in the resulting videofile. (samplerate is measured in in samples per second)

The combo box provides some common used samplerate values.

If the target samplerate is different form the original samplerate in the specified audiofile a audiofile conversion must be done.

Press the button "Audioconvert" to perform this conversion.

This may take a while and then displays the name of the generated audiofile as Tempfile.

The line below the tempfile shows information about the generated temporary audiofile and the audioplaytime in mm:ss:msec format.

Audio Tool Configuration

All the video encoder plug-ins are designed to accept 16-Bit mono or stereo WAV Files as audioinput.

The master videoencoder tries to convert other audio file formats and samplerates before it passes the audiodata to one of the video encoder plug-ins. This converion is handled via an external audiotool program.

The name of this program and its options can be configured here.

Audiotool:

By default sox is used as the audiotool to perform audio conversions.

sox can handle many audioformats, but can not read MP3 audiofiles.

Options:

```
"$IN" -w -r $RATE "$OUT" resample
```

Here you can specify the options that will be passed to the external audiotool program.

Save button

Save current audiotool configuration as parameters in your gimprc file.

Load button

Load audiotool configuration from gimprc parameters.

Default button

Restore to hardcoded audiotool configuration presets.

Extras

Macrofile:

Here you can specify a filtermacro file, that should be performed on each handled frame at encoding time on the fly.

(**Note:** filtermacros are limited and not real macros. the GIMP-GAP specific filtermacros are recorded filternames and parametervalues restricted to the supported filters, such as color curves, sharpen, blur and many more ...
)

Storyboard File:

Here you can specify the name of a storyboard file. When a valid storyboard file is entered the file will be checked. If errors are detected they are reported as text below the storyboard file entry widget.

If the storyboard file check is ok, the text will be a report about the properties and tracks found in the file.

Further the Input Mode (in the Video Options notebook tab) is set to "Storyboard".

Stroybard Audio:

The Button "Create Composite Audiofile" starts the audio mixdown processing for all included audiotracks of the storyboard and creates a composite audiofile in RIFF Wave fileformat.

The name of this resulting file will be inserted in the "Audiofile:" entry at the Audio Options notebook tab.

The "Create Composite Audiofile" button is set insensitive when there is no valid storyboard specified, or when the specified storyboard file has no audiopart.

Monitor Frames while Encoding checkbutton

will create an image during the encoding process and replaces its content each time a new frame is passed to the encoding engine.

(you may close that image any time you want, to stop monitoring)

Debug Flat File:
Debug Multilayer File:

Video/Encode/MPEG1 (mpeg_encode)

"plug-in-gap-mpeg-encode"

MPEG1 Encode

Start from Menu:

<Image>/Video/Encode/MPEG1...

Encode frames as MPEG1 videofile.
This feature is only available on UNIX systems and is
a frontend for the Berkeley MPEG-1 videoencoder
mpeg_encode 1.5.
For operation mpeg_encode 1.5 must be installed on your system
at runtime.

<ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/mpeg/bmt1r1.tar.gz>

Warning:

this is old software.
You may want to use the Master Videoencoder instead.

Limits:

Your frames must be stored in PPM or JPEG image fileformat.
There is no support for storyboard based encoding
and /or filtermacros at encoding time.
There is no audiosupport.

FromFrame:

ToFrame:

Here you can specify the range of frames that should
be included in the resulting MPEG encoded videofile.

Framerate:

The framerate in frames per second.

Bitrate:

Bitrate value.

Constant Bitrate:

ON: use constant bitrate.
OFF: use variable bitrate.

Pattern:

A stringpattern to specify the sequence of frames.
valid characters are
I ... for initial frame type (pattern should always start with I)

Video/Encode/MPEG1 (mpeg_encode)

P ... for predictable frames
B ... for bidirectional predictable frames

IQSCALE

PQSCALE

BQSCALE

Quality scale for I-frames, P-frames and B-frames
where 0 is best quality, 31 high compressin low quality)

P-Search

Select algorithm for P-frame calculation.

B-Search

Select algorithm for B-frame calculation.

Outputfile:

Name of the resulting MPEG videofile.

GenParams

this button generates a shellsript (using UNIX bourne shell syntax) with the required calls to run the external mpeg encoder program. An additional Parameterfile is generated along with the shellsript. It contains the parameter settings that are required to call the external mpeg encoder program and is referred in the generated shellsript.

The generated shellsript is written to the filename specified by the "Startscript" entry.

The entry "Parameterfile:" specifies the name of the parameterfile.

Gen + Encode

this button does the same as "GenParams" but also starts the generated script.

To prepare for MPEG encoding:

- Use 'Frames Convert' from the video menu to convert your video frames to .ppm (or another fileformat that can be read directly by your encoder)
- If you have a layerstack-animated multilayer image you can use 'Split Img to Frames' from the video menu.
- If height or width are not multiples of 16: Use 'Frames Scale' or 'Frames Crop' from the video menu on the newly converted/created video frames.
- Then use 'Encode/MPEG1...' from the video menu. (Invoked from one of the prepared video frames)

The encoder offers a lot of parameter settings for the experienced User. The parameters are passed to the encoder in a parameterfile.

The frontend dialog shows only the most important parameters, but generates a documented parameterfile with default values for the other parameters.

You can generate the parameterfile with or without starting the encoder.

(You may start the encoder later by executing the startscript from a Unixshell. The startscript is generated by the frontend dialog)

For more information on the parameters please refer to the encoders documentation.

Video/Encode/MPEG2

"plug-in-gap-mpeg2encode"

MPEG2 Encode

Start from Menu:

<Image>/Video/Encode/MPEG2...

Encode frames as MPEG2 or MPEG2 videofile.
This feature is only available on UNIX systems and is a frontend for the videoencoder mpeg2encode 1.2 (MPEG Software Simulation Group)

Web: <http://www.mpeg.org/MSSG>
FTP: <ftp://ftp.mpeg.org/pub/mpeg/mssg>
E-mail: mssg@mpeg.org (author contact)

For operation mpeg2encode 1.2 must be installed on your system at runtime.

Warning:

this is old software.
You may want to use the Master Videoencoder instead.

Limits:

Your frames must be stored in PPM image fileformat.
There is no support for storyboard based encoding and /or filtermacros at encoding time.
There is no audiosupport.

FromFrame:

ToFrame:

Here you can specify the range of frames that should be included in the resulting MPEG encoded videofile.

Framerate:

The framerate in frames per second.

Bitrate:

Bitrate value.

MPEG-type

Radio button to select MPEG1 or MPEG2 type.

Videoformat:

Combo box to select videonorm (NTSC, PAL or SECAM)

Outputfile:

Name of the resulting MPEG videofile.

GenParams

this button generates a shellscript (using UNIX bourne shell syntax) with the required calls to run the external mpeg encoder program. An additional Parameterfile is generated along with the shellscript. It contains the parameter settings that are required to call the external mpeg encoder program and is referred in the generated shellscript.

The generated shellscript is written to the filename specified by the "Startscript" entry.

The entry "Parameterfile:" specifies the name of the parameterfile.

Gen + Encode

this button does the same as "GenParams" but also starts the generated script.

To prepare for MPEG encoding:

- Use 'Frames Convert' from the video menu to convert your video frames to .ppm (or another fileformat that can be read directly by your encoder)
- If you have a layerstack-animated multilayer image you can use 'Split Img to Frames' from the video menu.
- Then use 'Encode/MPEG2..' from the video menu. (Invoked from one of the prepared video frames)

The encoder offers a lot of parameter settings for the experienced User. The parameters are passed to the encoder in a parameterfile.

The frontend dialog shows only the most important parameters, but generates a documented parameterfile with default values for the other parameters.

You can generate the parameterfile with or without starting the encoder.

(You may start the encoder later by executing the startscript from a Unixshell. The startscript is generated by the frontend dialog)

For more information on the parameters please refer to the encoders documentation.

Video/Go To

GAP controlled Framechanges are GoTo Operations in the Video Menu or in The Video/VCR Navigator Dialog or in the Video/Playback dialogs.

```
plug_in_gap_goto "<Image>/Video/Go To/Any Frame"
plug_in_gap_first "<Image>/Video/Go To/First Frame"
plug_in_gap_last "<Image>/Video/Go To/Last Frame"
plug_in_gap_next "<Image>/Video/Go To/Next Frame"
plug_in_gap_prev "<Image>/Video/Go To/Previous Frame"
```

** ([see Introduction](#)) **

Video/Onionskin/Configuration

"plug-in-gap-onionskin-configuration"

Onionskin Layers:

Onionskin Layer(s) usually do show previous (or next) frame(s) of the video in the current frame.

The GIMP Video Menu has a submenu named "Onionskin" that provides functions to deal with onionskin layers

```
Start from Menu:
<Image>/Video/Onionskin/Configuration...
<Image>/Video/Onionskin/Create or Replace
<Image>/Video/Onionskin/Delete
<Image>/Video/Onionskin/Toggle Visibility
```

Onionskin layers are copies of previous or next frames of the video.

By default they are built by merging the visible layers of the previous frame but without the BG-Layer (and without other onionlayers).

Onionskin layers are usually placed above the background layer in the layerstack of the current frame.

Onion Layers are normally used for display of the previous scene when painting moving characters.

They behave like a normal layer, but you can toggle visibility or delete onionskin layers with Functions provided by this module.

Configuration:

In the onionskin configuration dialog window you can

- edit the settings for creation of onionskin layers
- apply the settings to a selected range of frames

- delete onionskin layers in a selected range of frames.

Reference Mode:

```
NORMAL          /* default */
    The Frame Reference Parameter is used as delta
    in just one direction.
    Negative values refer to previous frames only,
    positive values refer to next frames only.
BIDIRECTIONAL_SINGLE
    The Frame Reference Parameter is used as
    bidirectional delta in both directions
    With an initial Frame Reference offset -1
    the reference sequence is
    -1, +2, -3, +4, -5, +6, ...

BIDIRECTIONAL_DOUBLE
    The Frame Reference Parameter is used as
    bidirectional delta in both directions
    With an initial Frame Reference offset -1
    the reference sequence is
    -1, +1 -2, +2, -3, +3, ...
```

Onionskin Layers:

Defines how many onionlayers will be created in the current image.

Frame Reference:

Stepsize to reference the source framenummer from where we copy the onionlayer.

Examples:

```
-1 .. copy 1.st onionlayer from current framenummer -1
    copy 2.nd onionlayer from current framenummer -2

+2 .. copy 1.st onionlayer from current framenummer +2
    copy 2.nd onionlayer from current framenummer +4
```

The example assumes Reference Mode NORMAL.

Cyclic (Togglebutton):

Define fold back behavior when referencing previous or next frames:

ON: last frame has frame 0 as next frame.

OFF: last frame has NO next frame.

Stackposition:

Position where to place onionlayer(s) in the layerstack.

From TOP (Togglebutton):

ON: Stackposition is counted from top (0 is on top)

OFF: Stackposition is counted from bottom (0 is on bottom)

Opacity:

The 1.st onionlayer is created with this opacity seting (100 is full opaque, 0 is full transparent)

Opacity Delta:

- 100 : The 2.nd onionlayer is created with same opacity as the 1.st onionlayer.
- 80 : The 2.nd onionlayer is created with 80% of the opacity of the 1.st onionlayer, The 3.rd with 80% of the 2.nd and so on...

Ascending Opacity (Togglebutton):

ON: the farrest handled Neighbour Frame has the highest Opacity (the Primary Opacity defined above) This may be used to create cross-fading effects. (*) see cross-fading example settings below.

OFF: the nearest handled Neighbour Frame has the highest Opacity, Opacity is descending for the other frames at farer distance. This should be used for regular Onionskin usage.

Options how to merge the previous Layer into one Onionlayer:

Restrict by Layernames: Default '*' all Names. Here you can specify a layernamepattern. only layers with matching names are included in the onionlayer.

Ignore BottomLayer(s): Default 1 Here you can exclude the background layer (1) or n layers counted from the bottom of the layerstack from the merged copy of the frame that makes up the onionlayer.

Auto Create after Load

ON: Force automatic Creation (or Replacement) of Onionskin Layer(s) on GAP controlled Framechanges.

Auto Delete before Save

ON: Force automatic Delete of Onionskin Layer(s) on GAP controlled Framechanges.

GAP controlled Framechanges are GoTo Operations in the Video Menu or in The Video/VCR Navigator Dialog or in the Video/Playback dialogs.

The Buttons:

APPLY:

The apply button creates onion layers in the selected range of frames and sets the values for the handled Animation.

DELETE:

The delete button removes onion layers in the selected range of frames and sets the values for the handled Animation.

CLOSE:

This button Closes the window and sets the values for the current handled Animation, without changing any frame.

DEFAULT:

Load hardcoded defaults (does not set anything)

SET for <name_of_the_current_animation>:

This button sets the values for the current handled Animation.

Restrictions:

- The Onionskin automatic does NOT work if you load a frame via the File Menu, or othe implicate file load/save operations.
- Onionskin Layers are restricted to Imageformats that can store multiple layers, layer parasites and tattoos.

GIMP's native format XCF is rcommanded.

Onionskin Functions:

The other functions in the Video/OnionskinLayer menu do not open dialog windows. They are intended for quick use, and it is recommended to assign key-shortcuts if you want to use them frequently.

here we should refer to gimp docs howto assign key-shortcuts

Create or Update Onion Layers:

Create onionlayers in the current frame, according to the configurated settings.
(see above)

Toggle Visibility:

Turn visibilty on (or off) for all onionlayers.

Delete Onion Layers:

Remove all onionlayers from the current frame.

Note: if you made a copy of an onionlayer manually the copy should not be deleted. If you have gimp-1.2.2 the copy will be deleted too, because there is a bug in the gimp video base functions.

Tips: Cross-Fading Example

With special Onionskin Settings you can get Cross Fading Effects.

An animation was created with a Framerate of 8 f/sec and should playback on a target rate of 24 f/sec

As 1.st step you need to duplicate all the frames in a way like this:

```

01 02 03 04 ...          original squence

01 02 03 04 05 06 07 08 09 10 11 12 .... new numbers
01 01 01 02 02 02 03 03 03 04 04 04 .... (old original numbers)

```

Use the Menu <Image>/Video/Frames Density... for this task.

To to improve smoothness, you can use Onionskin Layers to show 1 and 2 frames ahead on top of the Layerstack.

In The 2.nd Step You create Onionskin Layers in all resulting Frames with the following Settings:

```

Onionskin Layers: 2      Ascending Opacity: ON
Frame Reference: 1      Cyclic:                OFF
Stackposition: 0        From Top:              ON
Opacity:             66.6  50.0

```

```
Ignore BG-layer(s): 0      // do not ignore any Layer
```

```
From Frame: 1
To Frame : the last frame of your animation
```

When you wan equally balanced Opacity
you should use Opacity 1/3 for each frame
33.3 %, 100%

Split Video into Frames/MPlayer based extract

"plug-in-gap-mplayer-decode"

MPLayer based extract

Start from Menu:

```
<Image>/Video/Split Video to Frames/MPLayer based extract
```

```
<Toolbox>/Xtns/Split Video to Frames/MPLayer based extract
```

Extract frames and/or audio from a videofile.
This feature is only available on UNIX systems and is
a frontend for the mediaplayer program mplayer.

For that feature GIMP-GAP must be configured and compiled
with the configure option:
--enable-unix-frontends

Further it is required that mplayer is installed on your system
at run time.
(GIMP_GAP was tested with mplayer-1.0pre5).

Split Video into Frames/MPlayer based extract

MPlayer does support many videofileformats amd codecs and offers access to more vidoefiles than the other GIMP-GAP video extract features. But it does not provide frame exact positioning.

Input Video:

Here you can enter the name of an input videofile.
The "..." button opens a filebrowser popup dialog where you can select a filename.

Start Time:

Start time of the selected range in format HH:MM:SS.

Note:

MPlayer always extracts the 1.st frame of the video too regardless if you start at 00:00:00 or not.

Frames:

Number of frames to extract. If the video has less frames than specified here, extract will stop at the last available frame.

Videotrack:

Here you can select the number of the input videotrack.
Most videos have just one videotrack.
Some DVD videos have multiple angle view where the different angles are represented by 2 or more videotracks.
Use a videotrack value 0 to ignore videoframes.

Audiotrack:

Here you can select the number of the input audiotrack.
Most videos have just one audiotrack.
Some DVD videos have multiple language support where the different languages are represented by 2 or more audiotracks.
Use a audiotrack value 0 to ignore audiotracks.

Output Audio:

Filename for the extracted audio wave file to write.
(ignored if audiotrack is 0, or if the input video has no audiopart)

Framenames:

Basenamepart for the extracted frames to write.
The 6-digit numberpart and the extension (.png, .jpeg .xcf) are added for each extracted frame.

Format:

Imagefileformat for the extracted frames.
JPEG
PNG
XCF

Note: XCF fileformat is not directly supported by mplayer.
In this case the ectracted frames are created in PNG imagefileformat and converted to GIMP's native XCF fileformat automatically.

Split Video into Frames/MPlayer based extract

You should use Png Compression 0 in that case for faster operation.

Png Compression:

Compression for extracted PNG frames where 0 is uncompressed (fastest) and 9 is best compression. Png Compression is ignored if format JPEG is used.

Jpeg Quality:

Quality of the frames where 100 is best quality (but largest filesize)

Jpeg Optimize:

Optimization factor [0-100]

Jpeg Smooth:

Smooth factor [0-100]

Jpeg Progressive:

Specify standard or progressive JPEG.

Jpeg Baseline:

Specify use of baseline or not.

The Jpeg options are ignored if another format than JPEG is selected.

Silent:

ON: tell mplayer to ignore video audio when no track is selected.

OFF: mplayer plays unselected video /audio (where track is 0) on standard devices.

E.g. if audiotrack is 0, you will hear the audio track, if videotrack is 0, the corresponding frames will be visible at audioextract.

Open:

Open the 1.st extracted frame when extracting is done.

Asynchron:

ON: Run mplayer as asynchron process and try to show progress on base of the already extracted frames.

OFF: call mplayer as synchron process.

(progress for the extracting step is not possible in that case)

Split Video into Frames/XANIM based extract

"plug-in-gap-xanim-decode"

XANIM based extract

Start from Menu:

<Image>/Video/Split Video to Frames/XANIM based extract

<Toolbox>/Xtns/Split Video to Frames/XANIM based extract

Split Video into Frames/XANIM based extract

WARNING:

This feature is deprecated, because XANIM is old unmaintained software. If you still have XANIM 2.80 exporting edition it may still work. If not it is not worth to make efforts to get it working because GIMP-GAP has frontend support for MPlayer and built-in support for video extract based on FFMPEG / libmpeg3.

Extract frames and/or audio from a videofile.
This feature is only available on UNIX systems and is a frontend for the xanim player.

For that feature GIMP-GAP must be configured and compiled with the configure option:
--enable-unix-frontends

Further it is required that xanim is installed on your system at run time.
(GIMP_GAP was tested with xanim-2.80 exporting edition).

Video:

Here you can enter the name of an input videofile.
The "... " button opens a filebrowser popup dialog where you can select a filename.

From Frame:

Framenumber of the 1.st frame to extract.

To Frame:

Framenumber of the last frame to extract.

Framenames:

Basenamepart for the extracted frames to write.
The 6-digit numberpart and the extension (.ppm, .jpeg .xcf) are added for each extracted frame.

Format:

Imagefileformat for the extracted frames.
JPEG
PPM
XCF

Note: XCF fileformat is not directly supported by xanim.
In this case the ectracted frames are created in PPM imagefileformat and converted to GIMP's native XCF fileformat automatically.

Extract Frames:

ON: extract frames.
OFF: dont extract frames.

Extract Audio:

ON: extract audiopart.

Split Video into Frames/XANIM based extract

OFF: dont extract audiopart if there is one.

Jpeg Quality:

Quality of the frames where 100 is best quality (but largest filesize)

Open:

Open the 1.st extracted frame when extracting is done.

Asynchron:

xanim is not able to start extracting at the specified "From Frame" offset. Therefore this frontend automatically deletes unwanted frames that are extracted by xanim per default.

ON: Run xanim as asynchron process and try to delete unwanted frames while xanim is running and kill the xanim process after the last wanted "To Frame" framenummer was extracted.

OFF: call xanim as synchron process and delete unwanted frames when xanim process has finished.

Split Video into Frames/Extract Videorange

"plug-in-gap-extract-video"

Extract Videorange

Start from Menu:

<Image>/Video/Split Video to Frames/Extract Videorange

The extract videorange dialog provides features to extract a selected range of frames and/or audio from a videofile. It offers frame exact positioning and visual controlled range picking.

Input Video Selection:

The widgets in this area are used to specify the source videofile.

Videofilename entry:

Here you can enter the name of your source videofile.

Some of the common used videofileformats are supported. Typical extensions for videofiles are

- .avi
- .mov
- .mpg
- .mpeg
- .vob
- .ifo

Split Video into Frames/Extract Videorange

For a detailed list of supported videofileformats / codecs see ffmpeg and libmpeg3 documentation.

From Frame:

To Frame:

Here you can specify the range of frames that should be extracted.

If you want to extract all frames of the video and dont know the total number of frames, use 999999 in the "To Frame" widget.

The "Video Range" button opens a playback preview where you can view the video and set the range under visual control.

Videotrack:

Here you can select the video track, if the specified videofile has more than one videotrack.
(such as multiangle DVD stuff)

Select videotrack 0 to ignore video.
(in this case no frames are extracted)

Audiotrack:

Here you can select the audio track, if the specified videofile has more than one audiotrack.
(such as multilingual DVD stuff)

Select audiotrack 0 to ignore audio.
(in this case no audiodata is extracted)

Decoder:

In this widget you can specify a preferred videodecoder.

Use one of the values

"(none, automatic)"

"libavformat"

"libmpeg3"

provided in the combo-box to the right.

Normally you dont need this widget.

But some videofiles may cause problems (even crashes at open attempt) with one of the decoders.

In such cases it could help to set another decoder that may be able to read the same videofile.

In case where the specified decoder does not recognize your videofile as supported fileformat, another decoders will be tried

Split Video into Frames/Extract Videorange

automatically.

If a compatible decoder for the videofile was found its name is displayed as the Active Decoder.

Aspect Ratio:

Displays the aspect ratio information of the frames in the videofile. Aspect Ratio is defined as width / height. Commonly used aspect ratio values (4:3) or (16:9) will be detected and shown as additional information to the ratio value.

If the videofile contains no aspect ratio information, or the active decoder is not able to access this information the string "unknown" is displayed instead of the ratio value. (libavformat works well, libmpeg3 does not recognize the aspect ratio)

Output:

Mode:

If the "Create only one multilayer image" checkbox is ON only one GIMP-Image will be created in RAM. All the extracted frames are added as layers to this newly created image.

(Dont use this option if you want extract a huge amount of frames)

In the other mode (OFF) the extracted frames are written as sequence of numbered frame images in GIMP-GAP typical style.

The Name of those frame images is controlled by the entry widgets:

 Basename
 Digits
 Extension

where the digits control how many digits to use for the framenummer part and the extension does automatically define the image fileformat.

Framenr 1:

Here you can specify the framenummerpart for the 1.st extracted frame. A value of 0 will use the original framenummer of the input video.

Deinterlace:

The frame extraction has a built in deinterlace filter. Interlaced frames are built up of 2 half pictures, where the even lines build the one half and the odd lines build the other half picture.

Split Video into Frames/Extract Videorange

The deinterlace filter builds the extracted frame by picking only the lines of one of the half pictures and replaceng the lines of the other half picture by more or less smooth mixing the upper and lower neighbour lines. The mix is controlled with the spinbutton value where 1.0 is a smooth mix, and 0.0 takes just one of the neighbour lines without mixing. That way the deinterlaced frames have same height as the originals.

The Deinterlace combo-box options are:

"no deinterlace"

Turn off the deinterlace filter and extract frames 1:1 as they were read from the videofile.

"deinterlace (odd lines only)"

Extract the half-picture that is represented by odd lines

"deinterlace (even lines only)"

Extract the half-picture that is represented by even lines

"deinterlace frames x 2 (odd 1st)"

Extract both half-pictures starting with the odd one.

1 frames in the video gives 2 frame images on disc in that case.

"deinterlace frames x 2 (even 1st)"

Extract both half-pictures starting with the even one.

1 frames in the video gives 2 frame images on disc in that case.

Audiofile:

The filename for the extracted audiodata.

Audiodata is always written in RIFF WAVE Fileformat.

Video/Layer/Colors/Levels

"plug-in-wr-color-levels"

Color Levels

Start from Menu:

<Image>/Video/Layer/Colors/Levels

for automated apply, this filter is started indirect via the menus:

<Image>/Filters/Filter all Layers or
<Image>/Video/Frames Modify

when its internal PDB-name:

"plug_in_wr_color_levels"

is selected as filter name.

This color levels dialog works the same way as the color levels tool that can be found in the menu: Layer/Colors/Levels

but has a less comfortable dialog without preview.

The advantage of this variante is the ability to run automatically on many layers or frames with constant value settings or as animated filter with varying value settings in combination with the GIMP-GAP plug-ins:

"Filter All Layer"
"Modify Layers"

Video/Layer/Colors/CurvesFile

"plug-in-wr-curves"

Apply Color Curves File

Start from Menu:

<Image>/Video/Layer/Colors/CurvesFile

for automated apply, this filter is started indirect via the menus:
<Image>/Filters/Filter all Layers or
<Image>/Video/Frames Modify

when its internal PDB-name:

"plug_in_wr_curves"

is selected as filter name.

This curves file dialog works the same way as the color curves tool that can be found in the menu: Layer/Colors/Curves)

but has no graphical dialog for editing the color curve settings. It needs a color curves file as input.

The advantage of this variante is the ability to run automatically on many layers or frames with constant value settings or as animated filter with varying value settings in combination with the GIMP-GAP plug-ins:

"Filter All Layer"
"Modify Layers"

For applying this filter with varying values you need 2 color curves file and provide them as input for the first and last handled layer. All the layers inbetween will be automatically handled with slightly varying the color curve settings from the start color curve to the end color curve.

Layer/Colors/Hue-Saturation

"plug-in-wr-huesat"

Hue-Saturation

Start from Menu:

<Image>/Video/Layer/Colors/Hue-Saturation

for automated apply, this filter is started indirect via the menus:
<Image>/Filters/Filter all Layers or
<Image>/Video/Frames Modify

when its internal PDB-name:

"plug_in_wr_huesat"

is selected as filter name.

This Hue-Saturation dialog works the same way as the hue-saturation tool that can be found in the menu: Layer/Colors/Hue-Saturation

but has a less comfortable dialog without preview.

The advantage of this variante is the ability to run automatically on many layers or frames with constant value settings or as animated filter with varying value settings in combination with the GIMP-GAP plug-ins:

"Filter All Layer"
"Modify Layers"

Videoindex creation

"<Toolbox>/Xtns/Videoindex creation"

See: [Video Configuration \(gimprc\)](#)

[New] Video/Layer/Render/Fire Pattern...

"plug-in-firepattern"

Fire Pattern Filter:

Start from Menu:

<Image>/Video/Layer/Render/Fire Pattern...

This Plugin generates an animated fire effect.

It can render the animation in one call, where the input drawable is copied n-times to a newly created image.

Optional this Plugin can be called to render just one frame/phase of the animated effect.

This type of animated call on multiple already existent layers (or frames) with varying shiftPhase parameter

is supported by the GIMP-GAP filter all layers or by applying as filter via GIMP-GAP modify frames feature.

Note that the render one frame per call style offers more flexibility where you can

apply the flame with varying shape, color and opacity for each rendered frame atomatically.

Note that this Plugin will create an additional image when the options for creating fire pattern or fire shape are selected.

If you intend to record the Plugin using GIMP-GAP filtermacro feature or want to be able to reproduce the exactly same same results in another GIMP session

You have to save this image to disc (using the XCF file format)

Animation options:

Create Image: (checkboxbutton)

ON: create a new image with n copies of the input drawable and render complete animation effect on those copies.

OFF: render only one frame/phase of the animation effect on the input drawable.

(This type of call must be selected in case you call this plug-in via the GIMP-GAP filter all layers feature)

N-Frames: (spinbutton)

Number of frames to be rendered as layer in the newly created image. In this mode the vertical shift per frame is calculated automatically for each frame varying from 0.0 to Phase Shift value.

Select 1.0 (or multiples of 1.0) to create an animation that can be played in seamless loop.

(disabled in case Create Image: OFF is selected)

Phase shift: (spinbutton)

Vertical phase shift (movement of the fire pattern)

where 1.0 refers to image height.

In animated calls via (filter all layers) it is recommended to set Phase shift to value 0.0 for the first and to value 1.0 for the last processed frame to create a seamless animation effect when playback of the rendered frame is done in a loop.

Pattern options:

Widgets to control the fire pattern cloud layer that is used as base for rendering the animated fire effect.

Create Pattern: (checkboxbutton)

ON: create firepattern cloud layer with tileable solid noise according to options.

(Creating a new pattern should be used only in case rendering the first frame of an animation sequence. for each further frame it is recommended to turn this option OFF)

OFF: Use external pattern layer.

Typically you may select a fire pattern layer that was created for rendering the first frame of the animation.

Stretch Height (spinbutton)

vertical stretch factor for the fire pattern.

Scale Pattern X/Y: (2 spinbuttons)

Horizontal/Vertical scaling of the random pattern that is created for rendering the fire pattern (cloud layer)

Seed Pattern: (spinbutton)

Seed for creating random pattern (cloud1 layer) use 0 for random value.

Detail: (spinbutton)

Detail level for creating random solid noise pattern (cloud layer)

Layer Pattern: (combobox)

Select an already existing pattern layer

(typically from previous run, when rendering in one frame per call

mode)

Fireshape options:

Widgets to control the fire shape layer that is used to control the affected area where to render the flames.

Create Fireshape: (checkboxbutton)

ON: create fire shape layer according to options.

OFF: Use external fire shape layer.

Trapezoid: (checkboxbutton)

ON: Render trapezoid shaped fire.

OFF: render fire at full image width.

Flame Height: (spinbutton)

Height of the flame (1.0 refers to full image height)

Flame Border: (spinbutton)
border of the flame (used for horizontally soft fade of the shape at the borders)

FlameWidth: (2 spinbuttons)
width of the flame at base line and
width of the flame at flame height
(1.0 for full image width)

Flame Center: (spinbutton)
horizontal offset of the flame center
(0 for center, -0.5 left border +0.5 at right border of the image)

Fire Shape:
Select an already existing fire shape layer
(typically from previous run, when rendering in one frame per call mode)

Render options:

Create FireLayer: (checkboxbutton)
ON: Render fire pattern effect as separate layer.
OFF: merge rendered effect onto processed layer.
(it is recommended to set the Transparent BG to ON too)

(In case this filter is called via the filter all layers feature it is recommended to turn this option OFF)

Blend Mode: (radio buttons)
Selects the blend mode to be used to combine the fire pattern (cloud) layer with the fire shape layer. Following modes are available:
"Burn"
"Subtract"
"Multiply"

Transparent BG: (checkboxbutton)
ON: Render fire layer with transparent background.
OFF: render with black background.

Opacity: (spinbutton)
The opacity of the rendered flames.

Reverse Gradient: (checkboxbutton)
ON: use reverse gradient colors.
OFF: use gradient colors.

Gradient: (gradient button)
Select the gradient that should be used to colorize the rendered flames.

Action Buttons:

Reset

Reset all parameters to default values.

Cancel

Close the window without any further action.

OK

Close the window and render the fire effect according to the selected options.

Usage Examples:

Rendering fire with varying colors:

For this example we use a multilayer image where each layer represents one frame of an animation.
(in case this multilayer image was loaded from an animated GIF the mode has to be changed from Indexed to RGB before we continue)

From this multilayer image start the menu Filter/filter all Layers. This opens a browser dialog window where the plug-in-firepattern can be selected in the list of available filters. Set acceleration characteristics to value 1 for linear varying values with constant speed and press the APPLY button in the browser dialog.

This starts the Fire-Pattern dialog window of the selected plug-in-firepattern where you can specify the options to be applied for processing the first layer of the multilayer image.

1. st dialog step

Press the reset button to init all options with default values. Make sure to set Phase Shift to value 0 and Also both the Create Pattern and the Create Fireshape checkbuttons shall be turned on.

Press OK in the Fire Pattern dialog. This renders the fire on the first (e.g. the background) layer of your multilayer image.

After rendering a dialog window with title "Animated Filter Apply" pops up. Click the "Continue" button in this dialog.

2. nd dialog step

The Fire-Pattern dialog window appears again where you can enter the options for the last layer to be processed (the top layer of your multilayer image)

Uncheck the "Create Fireshape" checkbutton to use the same Fire Shape in all further frames.

Enter the value 1.0 in the Phase shift spinbutton entry widget. Click on the gradient and select another gradient name in the dialog that opens on this click

[New] Video/Layer/Render/Fire Pattern...

Press OK in the Fire Pattern dialog. This renders the fire on the last (e.g. the top) layer of your multilayer image.

After rendering a dialog window with title "Animated Filter Apply" pops up.

Click the "Continue" button in this dialog.

This triggers rendering the fire effect for all further layers (between bg and top) of your multilayer image.

Rendering fire with varying external shape:

In case you want render the fire with another shape (than the built in trapezoid)

you can provide the fire shape layer in another image.

For varying external shape this other image should be a multilayer image where each layer represents another step of the animated fire outline shape.

(typically white at base fading to black until the desired flame height)

It is recommended to use the same number and size for the "external"

fireshape multilayer image and for the multilayer image where the fire effect shall be rendered onto.

How it works:

This filter generates a fire pattern (cloud) layer with tileable solid noise, and a fire shape layer.

The fire effect is made by placing the fire pattern above the fire shape layer using "Burn" (or "subtract")

combination mode. The animation is done by vertically shifting the fire pattern upwards slightly

on each frame.

The flames are built by merging the fire pattern at its shifted state with the fire shape.

The resulting (gray flame) layer is colored with the colors of the selected gradient.

The mapping of the colors (and optional opacity) is done by luminance.

[New] Video/Layer/Render/Water Pattern...

"plug-in-waterpattern"

Water Pattern Filter:

Start from Menu:

<Image>/Video/Layer/Render/Water Pattern...

This plug-in generates an animated water effect that looks like the bottom of a ripple tank.

It can render the animation in one call,

where the input drawable is copied n-times to a newly created image

[New] Video/Layer/Render/Water Pattern...

and renders the water pattern with slightly shifted on each copy.
(in this case the amount of phase shifting)

Note that this Plugin will create an additional image when the option for creating patterns is selected.

If you intend to record the Plugin using GIMP-GAP filtermacro feature or want to be able to reproduce the exactly same same results in another GIMP session

You have to save this image to disc (using the XCF file format)

Animation options:

Create Image: (checkboxbutton)

ON: create a new image with n copies of the input drawable and render complete animation effect on those copies.

OFF: render only one frame/phase of the animation effect on the input drawable.

(This type of call must be selected in case you call this plug-in via the GIMP-GAP filter all layers feature)

N-Frames: (spinbutton)

Number of frames to be rendered as layer in the newly created image.

In this mode the vertical and horizontal shift value per frame is calculated automatically

for each frame varying from 0.0 to Phase Shift X (Y) value.

Select 1.0 (or multiples of 1.0) to create an animation that can be played in seamless loop.

(disabled in case Create Image: OFF is selected)

Phase shift X / Y: (2 spinbuttons)

Horizontal (Vertical) shift phase where 1.0 refers to image width (height)

Pattern options:

Widgets to control the water pattern cloud layers that are used as base for rendering the animated water effect.

Create Pattern: (checkboxbutton)

ON: create waterpattern cloud layers according to pattern options.

OFF: Use (already existing) external pattern layers.

Layer Pattern 1 / 2:

Select an already existing pattern layer (from previous run)

Scale Pattern X/Y: (2 spinbuttons)

Horizontal/Vertical scaling of the random patterns that is created for rendering the water highlights and displacement effects (cloud layer1 and 2)

Seed Pattern 1 / 2: (2 spinbuttons)

Seed values for creating random patterns (cloud1 and cloud2 layers) use 0 for random value.

Render options:

Use Highlights: (checkbox)

ON: Render water pattern highlight effect

OFF: Disable highlight effect.

Opacity: (spinbutton)

The highlight strength (e.g. opacity)

Blend Mode: (radio buttons)

Selects the blend mode to be used to combine the patterns (cloud1 and cloud2 layers)

Following modes are available:

"Overlay"

"Addition"

"Screen"

"Dodge"

Use Displace Map: (checkbox)

ON: Render water pattern distortion effect by applying a displace map that is generated by combining both patterns (cloud1 and cloud2 layer)

OFF: Disable distortion rendering.

Displace Strength: (spinbutton)

The distortion displace strength.

Note that values greater than 0.02 results in heavy distortions (that does not look naturally).

How it works:

This filter generates two layers with tileable solid noise, and sets the top layer to difference mode. This creates dark "bands" where the values of each layer cross.

This result is normalized and has the curves filter applied to bring out the bands in white instead of black.

This result is blended over the base layer (the one that was active when the script is called)

with the chosen blend mode.

With create Image option switched ON this entire procedure is repeated for each frame,

with slight offsets of the solid noise (e.g. cloud 1 & 2) layers, so the bands are shifted slightly each time.

The offsets are the size of the image multiplied by phase value and divided by the number of frames.

In case phase value 1 (or multiples of 1) is used at the end (processing of last layer)

the solid noise layers are back where they started, and the entire animation loops seamlessly.

Note that phase values control the speed of the animation effect where higher values

result in faster speed.

[New] Layer/Transparency/Foreground Extract Via Selection

"plug-in-selection-to-foreground-layer"

Filter for Foreground Extraction based on Selection

Start from Menu:

<Image>/Layer/Transparency/Foreground Extract Via Selection

How to use:

Have an image with a foreground object loaded in GIMP
(for instance a person) that shall be
separated from the background.
The image shall be of RGB color

Draw a selection (for instance by using the freehand
selection tool)
The selection will be used as base for the foreground extraction
where the opacity and optional color of pixels near
the border will be trimmed automatically.

Start the script from <Image>/Layer/Duplicate Selected Foreground.
This creates a new layer with the extracted foreground object
where the selection is automatically trimmed according
to matching colors.
It also creates a tri map (that is attached as layermask
to the input layer) that is internally used by the alpha matting
algorithm.

When your selection is drawn all inside the object,
you shall use inner radius 0 and set outer radius to
the desired width where the selection shall be trimmed.

When your selection is drawn all outside the object,
you shall use outer radius 0 and set outer radius to
the desired width where the selection shall be trimmed.

Note:

If this filter is called without having a selection
it uses the alpha channel as implicit selection.

Options:

InnerRadius

Specifies how many pixels inside the borderline of the selection
are considered as undefined area.

OuterRadius

Specifies how many pixels outside the borderline of the selection
are considered as undefined area.

[New] Layer/Transparency/Foreground Extract Via Selection

Create Layermask

- ON: Render opacity by creating a new layer mask for the resulting layer
- OFF: Apply rendered opacity to the alpha channel

Lock Colors

- ON: Keep RGB channels of the input layer
- OFF: Allow Background color removal in processed undefined regions

How it Works:

A tri-map is generated based on the current selection by setting the pixels that are near the selection border to UNDEFINED state.

Pixels that are inside the selection and far (distance is > inner radius) from the border are considered as FOREGROUND.

Pixels that are far (distance is > outer radius) outside the selection are considered as BACKGROUND.

The generated tri map is attached as layermask to the input layer.

This generated tri map is used to perform a foreground extraction

based on the alpha matting algorithm (# see plug-in-foreground-extract-matting)

The foreground extraction renders a new layer where transparency (and optionally color)

is calculated for the UNDEFINED area, FOREGROUND is rendered fully opaque and BACKGROUND is rendered fully transparent.

STORYBOARD_FILES

STORYBOARD_FILE_DOC.txt 2011.04.14:

General

A STORYBOARD_FILE is a textfile that can be used to assemble many audiofiles, videoclips, single images and frame sequences to one resulting video.

The images and frame sequences may be located in different directories, and can contain frame_images of any imagefiletype (.jpg, .xcf, .gif ...) that is readable by GIMP. You may also use parts from videofiles. (.avi, .mpg ... see chapter supported video input) all frames are converted and scaled to the desired video master size on the fly at encoding time.

All frame sequences and videoclips are played at the desired master framerate. This will result in speed up or slow down in all non rate-matching sequences. You can compensate this effect by dropping/duplicating the input frames or by using individual stepsize parameters on your input videoclips.

The built in audio processing has support for uncompressed audio in the RIFF WAVE format, but you can use any other Auidoformat if you have installed an external converter tool.

sox and lame are configured as default converter tools, so it is possible to use many different audio formats (including mp3). Recoding and resampling is done automatically where needed.

Syntax:

 The STORYBOARD_FILE is organized in text lines with blank separated items. The 1st item is a Keyword that describes the RECORD Type of the line. The other items depend on the RECORD Type. Lines starting with # are Comments, Blanklines are allowed and will be ignored.

You may join 2 or more Lines to one logical Line if you set the backslash '\' as last character (immediate before newline) item values may be specified using a parametername key prefix in the style:

```
key:value
```

If no parametername key prefix is specified, the fixed order of items is assumed. items must not contain blanks. (except blanks in "quoted strings")

Note: you should not use this style without key that is just supported for backwards compatibility with old GIMP-GAP versions)

Overview of supported RECORD TYPES:

```
-----
STORYBOARDFILE    version:1.0    # comments

VID_MASTER_FRAMERATE    frames_per_sec:24.0
VID_MASTER_FRAME_ASPECT width:16 height:9
VID_MASTER_SIZE        width:320 height:200
VID_PREFERRED_DECODER  libavformat
VID_MASTER_LAYERSTACK  track1:Background
VID_MASTER_INSERT_ALPHA format:"/path/file_%s_%06d.xcf"

AUD_MASTER_VOLUME      volume:1.0
AUD_MASTER_SAMPLERATE samples_per_sec:44100

# video clip descriptions
VID_PLAY_MOVIE        track:1 file:"movie_filename"    from:0001 to:0099 \
                        mode:{pingpong|normal}\
                        nloops:1 \
                        seltrack:1 \
                        exactseek:0 \
                        deinterlace:1.0 \
                        stepsize:1.0 \
```

```

        macro:"macro_filename" \
        flip:n \
        mask_name:"maskname" \
        mask_anchor:clip \
        mask_stepsize:1.0 \
        mask_disable:n \
        macsteps:1 \
        macaccel

VID_PLAY_ANIMIMAGE track:1 file:"image_filename"      from:0001 to:0099 \
        mode:{pingpong|normal} \
        nloops:1 \
        stepsize:1.0 \
        macro:"macro_filename"
        flip:n \
        mask_name:"maskname" \
        mask_anchor:clip \
        mask_stepsize:1.0 \
        mask_disable:n \
        macsteps:1 \
        macaccel

VID_PLAY_FRAMES   track:1 base:"basename" ext:.jpg from:0001 to:0099 \
        mode:{pingpong|normal} \
        nloops:1 \
        stepsize:1.0 \
        macro:"macro_filename"
        flip:n \
        mask_name:"maskname" \
        mask_anchor:clip \
        mask_stepsize:1.0 \
        mask_disable:n \
        macsteps:1 \
        macaccel

VID_PLAY_IMAGE    track:1 file:"image_filename"      nloops:1 \
        macro:"macro_filename"
        flip:n \
        mask_name:"maskname" \
        mask_anchor:clip \
        mask_stepsize:1.0 \
        mask_disable:n \
        macsteps:1 \
        macaccel

VID_PLAY_COLOR    track:1 red:0.0 green:1.0 blue:0.0 alpha:1.0  nloops:1 \
        flip:n \
        mask_name:"maskname" \
        mask_anchor:clip \
        mask_stepsize:1.0 \
        mask_disable:n \
        macsteps:1 \
        macaccel

```

```

VID_SILENCE          track:1                               nloops:1
wait_until_sec:0.0

# video attributes
VID_ROTATE          track:1 rotate_from:0.0 rotate_to:360.0 nframes:10
VID_OPACITY         track:1 opacity_from:0.0 opacity_to:1.0 nframes:10
VID_ZOOM_X          track:1 zoom_x_from:0.1 zoom_x_to:10.0 nframes:10
VID_ZOOM_Y          track:1 zoom_y_from:0.1 zoom_y_to:10.0 nframes:10
VID_MOVE_X          track:1 move_x_from:-1.0 move_x_to:1.0 nframes:10
VID_MOVE_Y          track:1 move_y_from:-1.0 move_y_to:1.0 nframes:10
VID_FIT_SIZE        track:1 mode:{both|none|width|height} proportions:
{keep_proportions|change_proportions}
VID_OVERLAP         track:1 nframes:0

# mask definitions
MASK_IMAGE          mask_name:"m1" file:"mask_filename"
MASK_ANIMIMAGE      mask_name:"m2" file:"mask_filename" from:0001 to:0099
MASK_FRAMES         mask_name:"m3" base:"basename" ext:".jpg" from:0001 to:0099
MASK_MOVIE          mask_name:"m4" file:"mask_filename" from:0001 to:0099

# audio clip descriptions
AUD_PLAY_SOUND      track:1 file:"audiofile" start_sec:0.0 end_sec:10.2 \
                    volume:1.0 \
                    start_vol:0.0 fade_in_time:1.0 \
                    end_vol:0.0 fade_out_time:1.0 \
                    nloops:1
AUD_PLAY_MOVIE      track:1 file:"videofile" start_sec:0.0 end_sec:10.2 \
                    volume:1.0 \
                    volume:1.0 \
                    start_vol:0.0 fade_in_time:1.0 \
                    end_vol:0.0 fade_out_time:1.0 \
                    nloops:1 \
                    1=selecttrack
AUD_SILENCE         track:1 duration_sec:0.0 wait_until_sec:10.2

```

RECORD Types Details:

Here is the detailed description of the items for the supported RECORD Types. optional items have internal defaults and can be omitted.

```

required item numbers (1)
optional item numbers [7]

```

STORYBOARDFILE

File Header. This Record must be the 1.st Line in the file.

```

(1) Record Key          ... STORYBOARDFILE
(2) version             ... version string (1.0 for the current implementation)

```

VID_MASTER_FRAME_ASPECT

Specifies the relation of width / height for playback of the resulting video output. This may result in automatical scaling to the specified aspect on playback.

If nothing is specified, or one of the values width or height is 0, then playback will be done at the aspect derived from the original pixelsize.

You can encode a video at 720 x 576 pixels, but specify aspect 16:9

Note: not all video encoders do support aspect settings.

- (1) Record Key ... VID_MASTER_FRAME_ASPECT
- (2) width ... the width aspect
 (0: derive aspect from original pixelsize)
- (3) height ... the height aspect
 (0: derive aspect from original pixelsize)

VID_PREFERRED_DECODER

- (1) Record Key ... VID_PREFERRED_DECODER
- (2) decoder ... Name of the preferred Videodecoder
 The Videodecoder is relevant for all Clips specified with VID_PLAY_MOVIE or AUD_PLAY_MOVIE statements. If you set a preferred Videodecoder this one tried first. If it fails to open the videofile, all other decoders are tried next.

Currently you may choose the following decoders:

```
libavformat
libmpeg3
DEFAULT: NULL (no preferred decoder)
```

VID_MASTER_LAYERSTACK

- (1) Record Key ... VID_PREFERRED_DECODER
- (2) track1 ... control order of videotracks in the layerstack of the composite frame image.
 If track1 is placed on Top of the layerstack it appears in foreground. higher video track numbers appears behind track1, and the highest video track builds the background.
 Background tracks are only visible when all other

tracks

in the foreground have transparent areas, or do not cover the full image size (caused by ZOOM or MOVE operations)

You may choose the following options:

```
Top       ..... (track 1 is the foreground)
Background .... (track 1 is the background)
DEFAULT: Top
```

VID_MASTER_INSERT_ALPHA

Specifies a format string that refers to an image or to a series of frame images

to provide transparency for the corresponding video frame.
 This format string shall contain placeholders
 %s for the basename of video clips and
 %06d for the 6 digit framenumbers.

This format string will be checked each time when storyboard processing
 fetches a video frame (from clip type VID_PLAY_MOVIE)
 and will be evaluated to a corresponding frame/image filename.
 If a frame image with this evaluated filename exists,
 that image will be automatically loaded,
 scaled to frame size and attached as alpha channel
 to the fetched frame.
 This feature offers individual transparency per video clip and frame.

Example1:

Have a Storyboard with:

```
VID_MASTER_INSERT_ALPHA format="/home/storyboards/video_alpha_frame/alpha_
%s_%06d.png"
VID_PLAY_MOVIE track=2 file=my_clip.AVI from=12 to=20
```

When processing fetches frame number 14 of my_clip.AVI
 it evaluates the alpha imagename by substitution
 where

```
  %s    is replaced by "my_clip"
  %06d  is replaced by "000014"
```

and checks if the resulting alpha imagename exists:

```
  /home/storyboards/video_alpha_frame/alpha_my_clip_000014.png
```

If there is such an image it will be added as alpha channel
 to add transparency for the fetched video frame.

Example2:

Have a Storyboard with:

```
VID_MASTER_INSERT_ALPHA format="/home/storyboards/video_alpha_frame/alpha_
%s.png"
VID_PLAY_MOVIE track=2 file=my_clip.AVI from=12 to=20
```

In this case there is just the placeholder for the videofile but no
 placeholder for the framenumbers.

In this case the alpha imagename evaluates to

```
  /home/storyboards/video_alpha_frame/alpha_my_clip.png
```

for all frames of this video clip. If such a file exists it
 is applied as alpha channel on all frames of my_clip.AVI

- | | |
|----------------|--------------------------------------------------------------------------------------------------|
| (1) Record Key | ... VID_MASTER_INSERT_ALPHA |
| (2) format | ... the format string
shall include
%s as placeholder for
videofile basename and |

%06d as placeholder for the framenumber.

VID_PLAY_MOVIE

fetch frames from a videofile. (such as MPEG, AVI, ...
see GAP_VID_API description for Informations about
supported videoformats and codecs)

(1) Record Key	...	VID_PLAY_MOVIE
(2) track	...	integer tracknumber
(3) file	...	filename of the video.
(4) from	...	start of Range Framenumber (integer) or seconds
(float)		(where 1 is the first frame in the video)
(5) to	...	end of Range Framenumber (integer) or seconds (float)
		(see Range description for more information)
[6] mode	...	optional keyword "pingpong" or "normal"
		DEFAULT: "normal"
[7] nloops	...	optional integer of Range repetitions
		DEFAULT: 1
[8] seltrack	...	Select input videotrack (for videos with more
videotracks.		videotracks)
		use 1 for the first track, or if the video has only one
videotrack)		DEFAULT: 1
[9] exactseek	...	0 .. use faster videoindex based seek ops for
Videoframe reads		when available.
		1 .. force sequential reads for exact positioning
		in the Video Read API (GVA) even if videoindex is
available.		In larger videos this will be very very slow.
		DEFAULT: 0
[10] deinterlace	...	0.0 (upto 0.99) read videoframes 1:1 without
deinterlacing		1.0 upto 1.99 deinterlace videoframes keeping odd rows
		where 1.0 does not interpolate the even rows (hard)
		and 1.99 does soft interpolation for the even
rows.		2.0 upto 2.99 deinterlace videoframes keeping even
rows		DEFAULT: 0.0
[11] stepsize	...	0.125 (upto 8.0) step size for fetching input frames
		a value of 0.5 does fetch each input frame twice
		a value of 2.0 does skip every 2.nd input frame
		use a value of 1.0 to fetch all input frames step by step
		DEFAULT: 1.0
[12] macro	...	optional name of a filtermacro_file
		that will be execute on each imported input_layer.

If the macsteps parameter (see below) is set to a value

> 1 and the filtermacro_file has a pendant file that provides alternative parameter sets, the filtermacro will be applied with varying values until the number of defined steps (parameter macsteps) is reached or all frames of this clip are processed. (for further details see also chapter about Macrofiles below)

[13] decoder ... optional name of preferred video decoder that should be used for the videofile read access.
 "libmpeg3"
 "libavformat"

[14] flip ... h ... (Horizontal) mirror the input frames horizontally
 v ... (Vertical) mirror the input frames vertically
 b ... (Both)
 n ... (None) do not flip input frames.
 DEFAULT: n (None)

[15] mask_name ... optional maskname refers to mask definitions. the mask name is a unique identifier of a mask clip that is applied to all frames of the clip. mask definitions are mask clips that will be attached as layermask to the input frames. This allows modification of opacity for individual masked areas.

[16] mask_anchor ... Defines how to apply the layermask.
 "clip" scale mask to resulting layer size.
 E.g. the mask scales and moves according to transformations of the input frame where it is attached to.
 (default)
 "master" scale mask to master frame size and apply layer after clipping the layer to master frame image. In this mode the mask is fixed to the master frame. and does not follow transformations and movements of the input frame.

"xcolor" scale mask to resulting layer size. and apply as color mask. In this case the image that is referred via mask_name should be a color image. the mask scales and moves according to transformations of the input frame where it is attached to.

[17] mask_stepsize: ... stepsize factor for the mask-frames. a value of 2.0 speeds up the mask effect (by skipping every 2nd frame in the assigned mask clip) a value of 0.5 will slow down the mask effect. Default 1.0 (normal speed for mask effect)

[18] mask_disable: ... disable flag

- [19] macsteps: ... an integer value greater or equal than 1.
Defines the (maximal) duration in number of frames
for applying filtermacros with varying values.
- [20] macaccel:
characteristic ... an integer value specifying acceleration
for applying filtermacros with varying values.
where 1 changes filtersettings with constant speed.
other positive values result in increasing speed,
negative values result in decreasing speed.
- [21] colormask_file: ... a filename referring to a file with parameters for
applying a mask with the colormask filter.
the colormask filter makes pixels of the processed
frame transparent when they are equal or similar to the
corresponding pixel in the colormask.
typically the colormask represents the non-moving
background that can be set transparent while moving objects are
opaque.

VID_PLAY_ANIMIMAGE

fetch frames from layers of a multilayer image (such as GIF animations,
or animations in GIMP native XCF fileformat)

- (1) Record Key ... VID_PLAY_ANIMIMAGE
(2) track ... integer tracknumber
(3) file ... filename of the multilayer image.
(4) from ... start of range layernumber (integer) or seconds
(float) (where 0 is the top layer in the multilayer image)
(5) to ... end of range layernumber (integer) or seconds (float)
(see range description for more information)
[6] mode ... optional keyword "pingpong" or "normal"
DEFAULT: "normal"
[7] nloops ... optional integer of Range repetitions
DEFAULT: 1
[8] stepsize ... 0.125 (upto 8.0) step size for fetching input frames
a value of 0.5 does fetch each input layer twice
a value of 2.0 does skip every 2.nd input layer
use a value of 1.0 to fetch all input layers step by step
DEFAULT: 1.0
[9] macro ... Same as described for VID_PLAY_MOVIE (see above)
[10] flip ... Same as described for VID_PLAY_MOVIE (see above)
[11] mask_name ... Same as described for VID_PLAY_MOVIE (see above)
[12] mask_anchor ... Same as described for VID_PLAY_MOVIE (see above)
[13] mask_stepsize: ... Same as described for VID_PLAY_MOVIE (see above)
[14] mask_disable: ... Same as described for VID_PLAY_MOVIE (see above)
[15] macsteps: ... Same as described for VID_PLAY_MOVIE (see above)

- [16] macaccel: ... Same as described for VID_PLAY_MOVIE (see above)
 [17] colormask_file: ... Same as described for VID_PLAY_MOVIE (see above)

VID_PLAY_FRAMES

fetch frames from numbered imagefiles with names follow the convention

- base_filename (may include directory path)
- number
- .
- extension

such as supported by GIMP Video Menu functions.

- (1) Record Key ... VID_PLAY_FRAMES
 (2) track ... integer tracknumber
 (3) base ... filename prefix part before the framenummer.
 (4) ext ... Extension (without '.' jpg, gif xcf ...)
 (5) from ... start of Range frame number (integer) or seconds
 (float) or keyword
 (6) to ... end of Range frame number (integer) or seconds (float)
 or keyword
 (see Range description for more information)
- [7] mode ... optional keyword "pingpong" or "normal"
 DEFAULT: "normal"
- [8] nloops ... optional integer of Range repetitions
 DEFAULT: 1
- [9] stepsize ... 0.125 (upto 8.0) step size for fetching input frames
 a value of 0.5 does fetch each input frame twice
 a value of 2.0 does skip every 2.nd input frame
 use a value of 1.0 to fetch all input frames step by step
 DEFAULT: 1.0
- [10] macro ... Same as described for VID_PLAY_MOVIE (see above)
 [11] flip ... Same as described for VID_PLAY_MOVIE (see above)
 [12] mask_name ... Same as described for VID_PLAY_MOVIE (see above)
 [13] mask_anchor ... Same as described for VID_PLAY_MOVIE (see above)
 [14] mask_stepsize: ... Same as described for VID_PLAY_MOVIE (see above)
 [15] mask_disable: ... Same as described for VID_PLAY_MOVIE (see above)
 [16] macsteps: ... Same as described for VID_PLAY_MOVIE (see above)
 [17] macaccel: ... Same as described for VID_PLAY_MOVIE (see above)
 [18] colormask_file: ... Same as described for VID_PLAY_MOVIE (see above)

VID_PLAY_IMAGE

fetch frame from a single imagefile. (Multilayer Images are handled as one composite image. For animated GIF's you can use VID_PLAY_ANIMIMAGE to playback all the layers separately)

- (1) Record Key ... VID_PLAY_IMAGE
 (2) track ... integer tracknumber
 (3) file ... filename of the image.
 [4] nloops ... optional duration in number of frames (integer) or
 seconds (float)
 DEFAULT: 1
- [5] macro ... Same as described for VID_PLAY_MOVIE (see above)

[6] flip ... Same as described for VID_PLAY_MOVIE (see above)
 [7] mask_name ... Same as described for VID_PLAY_MOVIE (see above)
 [8] mask_anchor ... Same as described for VID_PLAY_MOVIE (see above)
 [9] mask_stepsize: ... Same as described for VID_PLAY_MOVIE (see above)
 [10] mask_disable: ... Same as described for VID_PLAY_MOVIE (see above)
 [11] macsteps: ... Same as described for VID_PLAY_MOVIE (see above)
 [12] macaccel: ... Same as described for VID_PLAY_MOVIE (see above)
 [13] colormask_file: ... Same as described for VID_PLAY_MOVIE (see above)

VID_PLAY_COLOR

create unicolored frame.

(1) Record Key ... VID_PLAY_COLOR
 (2) track ... integer tracknumber
 (3) red ... 0.0 upto 1.0
 (4) green ... 0.0 upto 1.0
 (5) blue ... 0.0 upto 1.0
 (6) alpha ... 0.0 upto 1.0 alphachannel
 (where 0.0 is full transparent and 1.0 is full opaque)
 [7] nloops ... duration in number of frames (integer) or seconds
 (float)
 DEFAULT: 1
 [6] macro ... Same as described for VID_PLAY_MOVIE (see above)
 [7] flip ... Same as described for VID_PLAY_MOVIE (see above)
 [8] mask_name ... Same as described for VID_PLAY_MOVIE (see above)
 [9] mask_anchor ... Same as described for VID_PLAY_MOVIE (see above)
 [10] mask_stepsize: ... Same as described for VID_PLAY_MOVIE (see above)
 [11] mask_disable: ... Same as described for VID_PLAY_MOVIE (see above)
 [12] macsteps: ... Same as described for VID_PLAY_MOVIE (see above)
 [13] macaccel: ... Same as described for VID_PLAY_MOVIE (see above)
 [14] colormask_file: ... Same as described for VID_PLAY_MOVIE (see above)

VID_SILENCE

This record is used to define Pauses in a videotrack.

(1) Record Key ... VID_SILENCE
 (2) track ... integer tracknumber
 [3] nloops ... duration of the silent pause in number of frames
 (integer)
 or seconds (float)
 DEFAULT: 1
 [4] wait_until ... wait until this timestamp is reached,
 (then wait duration time)
 timestamp can be a framenummer (integer) or seconds
 (float)
 DEFAULT: 0

VID_ROTATE

This record is used to define rotate transitions,
 by changing the rotation angle of the processed frames in the specified
 Videotrack slightly from one value
 to another.

(1) Record Key ... VID_ROTATE
 (2) track ... integer tracknumber
 (3) rotate_from ... Start rotation with this value
 where 0.0 is and 360.0 is one full rotation
 [4] rotate_to ... Change rotation to this value
 DEFAULT: use same value as rotate_from
 [5] nframes ... optional duration of the effect
 in number of frames (integer) or seconds (float)
 The duration Value 0 applies the TO value immediate.
 DEFAULT: 0
 [6] accel ... an integer value specifying acceleration
 characteristic
 for speed behavior of rotation changes.
 where 1 (and 0) rotates the object with constant speed.
 other positive values result in increasing rotation
 speed, negative values
 result in decreasing speed.
 DEFAULT: 0

VID_OPACITY

This record is used to define Fade Effects,
 by changing the Opacity of the Videotrack slightly from one value
 to another.

(1) Record Key ... VID_OPACITY
 (2) track ... integer tracknumber
 (3) opacity_from ... Start opacity with this value
 where 0.0 is full transparent and 1.0 is full opaque
 [4] opacity_to ... Change opacity to this value
 DEFAULT: use same value as opacity_from
 [5] nframes ... optional duration of the effect
 in number of frames (integer) or seconds (float)
 The duration Value 0 applies the TO value immediate.
 DEFAULT: 0
 [6] accel ... an integer value specifying acceleration
 characteristic
 for speed behavior of opacity changes.
 where 1 (and 0) changes opacity with constant speed.
 other positive values result in increasing speed,
 negative values
 result in decreasing speed.
 DEFAULT: 0

VID_ZOOM_X and VID_ZOOM_Y

These records are used to define Zooming Effects
 by scaling the Videotrack slightly from one value
 to another. Zooming can be defined independet for Width (VID_ZOOM_X)
 and Height (VID_ZOOM_Y).

(1) Record Key ... VID_ZOOM_X (or VID_ZOOM_Y)
 (2) track ... integer tracknumber
 (3) zoom_from ... Start zoom with this value

double videosize where 0.0 is 1x1 Pixel, 1.0 is fit videosize, 2.0 is

[4] zoom_to ... Change zoom to this value
 DEFAULT: use same value as zoom_from

[5] nframes ... optional duration of the effect
 in number of frames (integer 10) or seconds (float)
 The duration Value 0 applies the TO value immediate.
 DEFAULT: 0

[6] accel ... an integer value specifying acceleration
 characteristic for speed behavior of zoom changes.
 where 1 (and 0) changes size with constant speed.
 other positive values result in increasing speed,
 negative values result in decreasing speed.
 DEFAULT: 0

VID_MOVE_X and VID_MOVE_Y

These records are used to define Scroll Effects by changing the Offset(s) of the Videotrack slightly from one value to another. Scrolling can be defined independet for X (VID_MOVE_X) and Y Directions (VID_MOVE_Y).

(1) Record Key ... VID_MOVE_X (or VID_MOVE_Y)
 (2) track ... integer tracknumber
 (3) move_x_from ... Start move with this value
 where -1.0 is left (or up) outside,
 0.0 is centered,
 +1.0 is right (or bottom) outside

[4] move_x_to ... Change offset to this value
 DEFAULT: use same value as move_from

[5] nframes ... optional duration time of the effect
 in number of frames (integer 10) or seconds (float)
 The duration Value 0 applies the TO value immediate.
 DEFAULT: 0

[6] accel ... an integer value specifying acceleration
 characteristic for speed behavior of movents.
 where 1 (and 0) changes positions with constant speed.
 other positive values result in increasing speed,
 negative values result in decreasing speed.
 DEFAULT: 0

Examples:

Assume we have video master size 320x200 pixels,

```

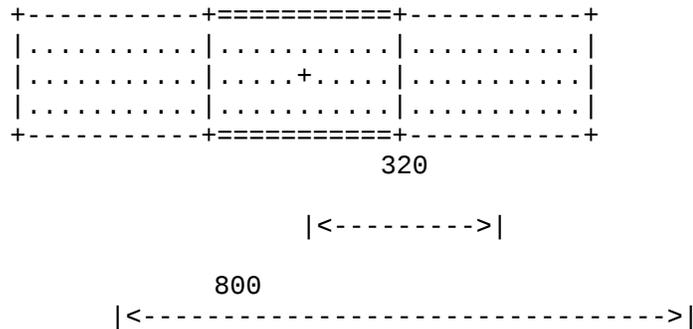
+-----+
|master  |
|width: 320|
|         |
+-----+
  
```

We use VID_ZOOM_X settings with a value of 2.0,

(also assume that automatical fit to master size is turned OFF using VID_FIT_SIZE mode=none)

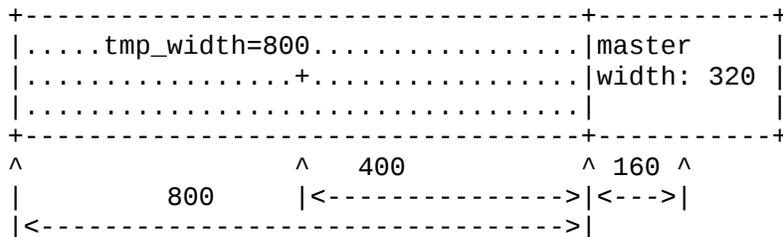
The source is an image 400 x 200 that is scaled to 800 x 200 (due to VID_ZOOM_X value of 2.0) that gives a tmp_width of 800 pixels.

Without any VID_MOVE attributes the result will be centered The normal be



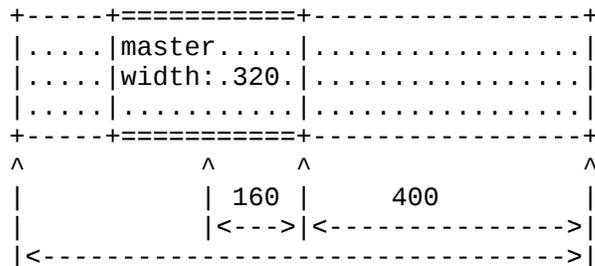
We use VID_MOVE_X setting to place the image completely outside to the left border at the first handled frame,

move_x_from = -1.0



the setting for the last handled frame places the image half outside the right border.

move_x_to = 0.5



VID_FIT_SIZE

Define how to match input framesize with the target size.
 The target size is typically the size of the resulting video (VID_MASTER_SIZE).
 except in scenarios where a movepath transition is active.
 In this special case the input frame is processed as moving object
 of this transition and the target size refers to the scaled size
 (preScaleWidth, preScaleHeight) of the recorded moving object.

Notes:

- The Input Frame is placed in the center of the composite output frame.
 (unless VID_MOVE_X/Y define other offsets than 0.0)
- Zooming may cause additional Scaling. The VID_FIT_SIZE record describes the normal Size (where zoom is set to 1.0)
- In case the input Frame is processed by a movepath transition the target size depends on the size of the moving object and the framesize in the xml_paramfile and on the VID_MASTER_SIZE.
 (see VID_MOVE_PATH below)

Examples:

in the xml_paramfile the frame size 640x400 pixels and the moving object size was 320x240 pixels.

```
<frame_description width="640" height="400"
<moving_object      width="320" height="240"
```

The actual rendering shall be done at double frame size VID_MASTER_SIZE 1280 x 800 pixels.

In this example the target size of the pre-scaled moving object is 640 x 480 pixels

target size calculation:

```
preScaleWidth = recordedMovingObjWidth * masterWidth / recordedFrameWidth
    640         = 320 * 1280 / 640
preScaleHeight = recordedMovingObjHeight * masterHeight / recordedFrameWidth
    480         = 240 * 800 / 400
```

- **WARNING:** settings that disable scaling
 (e.g. use fixed width or height at original source image size)
 do NOT follow changes of the resulting VID_MASTER_SIZE.
 In case you want to render videos in different sizes from the same storyboard
 by simply changing the VID_MASTER_SIZE sizes, you should NOT
 use such settings for your clips.

```
VID_FIT_SIZE mode=none # for fixed width
and height
VID_FIT_SIZE mode=width proportions=change_proportions # fixed width
VID_FIT_SIZE mode=height proportions=change_proportions # fixed height
```

Furthermore the fixed size also forces internal rendering for preview purpose to operate at full VID_MASTER_SIZE which may be very slow (especially for HD video)

The described behavior is valid for all fetched Frames in the same track until the next VID_FIT_SIZE record (or until end of file if there is none)

```
(1) Record Key      ... VID_FIT_SIZE
(2) track           ... integer tracknumber
[3] mode            ... One of the Keywords "width" "height" "none" or "both"
                    width:  Scale that only width does exactly
                             fit to the resulting video,
                             height is unchanged original input frame
height,
input frame
                    or adjusted to keep the proportions of the
                    (depends on the proportions setting)
                    height: Scale that only height does exactly
                             fit to the resulting video,
                             width is unchanged original input frame width,
                             or adjusted to keep the proportions of the
input frame
                    (depends on the proportions setting)
                    both:   Scale that both width and height do exactly
                             fit to the resulting video
                    none:   Do not Scale the input frame at all
                    DEFAULT "both"

[4] proportions    ... One of the Keywords "keep_proportions"
"change_proportions"
                    keep_proportions:
                    Keep proportions of the input frame at scaling.
                    This can lead to stripes on top/bottom
                    (or left/right).
                    especially if the resulting video is LANDSCAPE
oriented
                    and the input frame is PORTRAIT oriented.
                    The stripes are transparent and show the frames
                    of lower video track numbers, or the black
                    background if there no such frames.
                    change_proportions:
                    allow proportion changes at scaling.

                    DEFAULT: "change_proportions"
```

This Example shows how an Input Frame of size 400x400 pixel is automatically scaled to fit into a video at size 1280 x 800

```
+-----+
|#####|  Input Frame width:      400
|#####|  Input Frame height:   400
```

```
|#####|
|#####|
+-----+
```

==== scenarios that allow changing proportions of the moving object =====

o) VID_FIT_SIZE mode=both proportions=change_proportions

```

+-----+
|#####| scaled copy of input frame: 1280 x
800 |#####| video size: 1280 x
800 |#####|
|#####|
|#####|
|#####|
|#####|
|#####|
|#####|
|#####|
|#####|
|#####|
|#####|
+-----+
```

o) VID_FIT_SIZE mode=width proportions=change_proportions

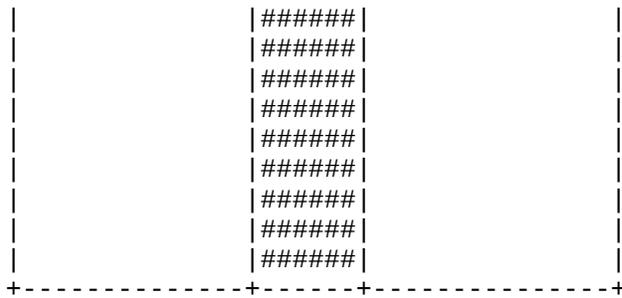
```

+-----+
|          | scaled copy of input frame: 1280 x
400 |          | video size: 1280 x
800 |          |
|          |
+-----+
|#####|
|#####|
|#####|
|#####|
+-----+
|          |
|          |
|          |
+-----+
```

o) VID_FIT_SIZE mode=height proportions=change_proportions

```

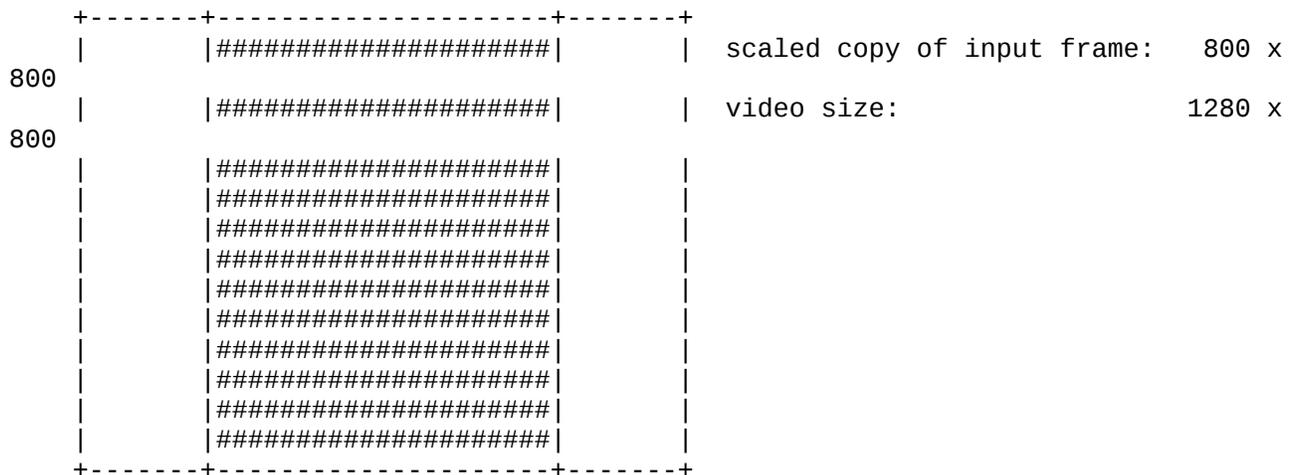
+-----+-----+-----+
|          |#####|          | scaled copy of input frame: 400 x
800 |          |#####|          | video size: 1280 x
800 |          |#####|          |
|          |          |          |
```



==== scenarios that keep proportions of the moving object =====

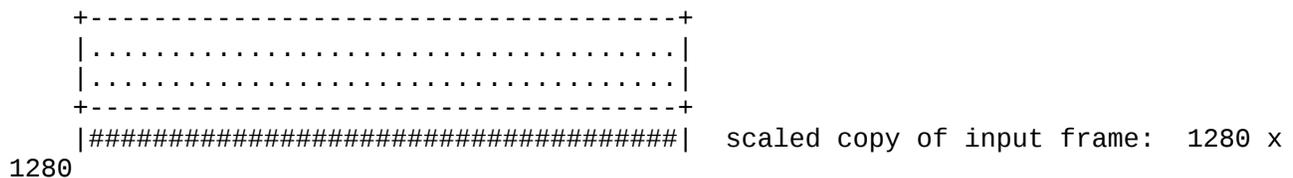
o) VID_FIT_SIZE mode=both proportions=keep_proportions

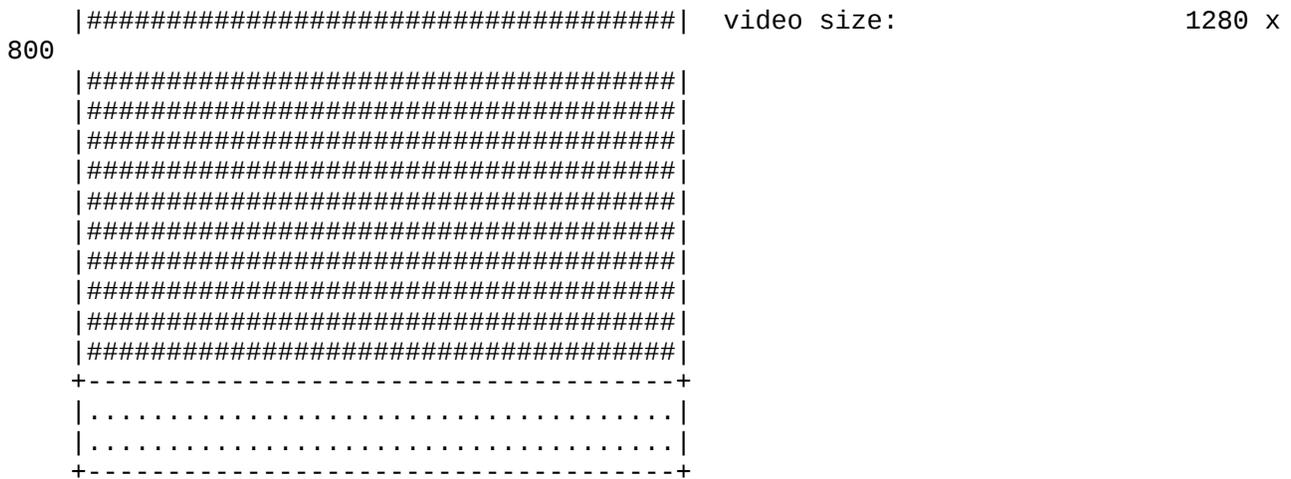
The input frame is scaled to fit into video size rectangle since the input frame has different proportions than the video size there will be transparent borders. (where the black background or clips of other tracks on lower stack position shows through)



o) VID_FIT_SIZE mode=width proportions=keep_proportions

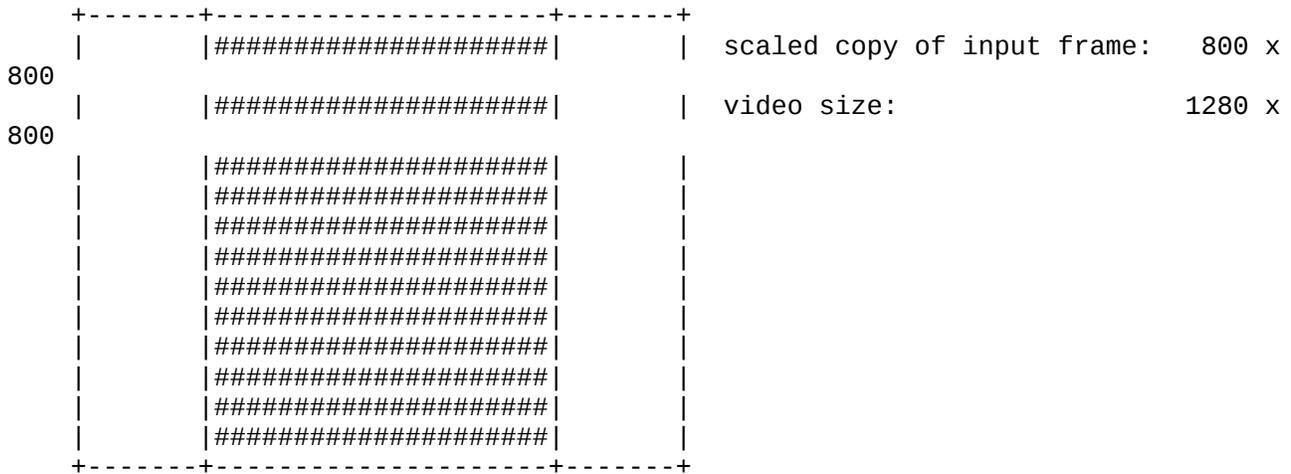
The input frame is scaled to width 1280 pixels and height 1280 pixels to keep the proportions. But this cuts off the parts that do not fit into the video height of 800 pixels.





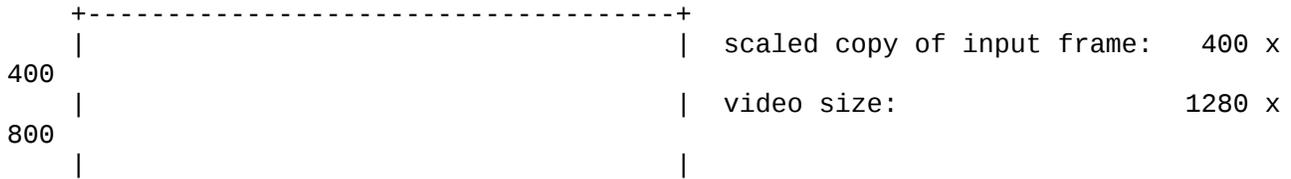
o) VID_FIT_SIZE mode=height proportions=keep_proportions

The input frame is scaled to height 800 pixels.
 keeping the proportions of the input frame results in width of 800 pixels
 that gives transparent stripes left and right.



o) VID_FIT_SIZE mode=none

No automatic scaling is done to fit into video size.
 (the proportions setting is not relevant in this case)




```

00100      A.mpg 000100  B.mpg 000005
00101      B.mpg 000006
00102      B.mpg 000007
...
00195      B.mpg 000100

```

VID_MOVE_PATH

This record is used to define a set of complex transformations on the current video track (layer)

Those transformations can include movement along a path, scaling, rotation, perspective transformation, opacity changes ... (see the GIMP-GAP MovePath feature for more details)

to another.

```

(1) Record Key      ... VID_MOVE_PATH
(2) track           ... integer tracknumber
(3) frame_from     ... Start frame number specifies the phase
                    where to start in the move path (typically start at 1)
(4) frame_to       ... End frame number specifies the phase
                    where to end in the move path
(5) nframes        ... duration of the effect
                    in number of frames (integer)
                    In case nframes is greater than total_frames
                    the transistions of the end
                    The duration Value 0 disables move transitions.
(6) accel          ... an integer value specifying acceleration
characteristic
                    DEFAULT: 0  ** other values are currently ignored

(7) xml_paramfile  ... name of the paramterfile for the move path plugin.

```

AUD_PLAY_SOUND

This record is used for playback of portions of an audiofile, with optional fade effects. Please note that all audiotracks are mixed to one composite audiotrack.

```

(1) Record Key      ... AUD_PLAY_SOUND
(2) track           ... integer tracknumber
(3) file            ... filename of soundfile (.WAV and other types).
(4) start_sec       ... start playback range at offest in seconds (float)
(5) end_sec         ... end playback range at offest in seconds (float)
                    Alternative you can specify the time in frames
                    (see from and to parameters below)
[6] volume          ... adjust the volume. (float)
                    A value of 1.0 keeps the original input volume for this
track
                    value 0.5 results in half, value 2.0 in double volume.
                    (the result may be additional affected by the

```

MASTER_VOLUME,

that is applied to the mixed composite audio)
 DEFAULT = 1.0

[7] start_volume ... Start Value of the volume. (float)
 DEFAULT: 1.0

[8] fade_in_time ... duration for the fade in. In this durationtime the
 input volume changes slightly from start_volume --> to volume.
 Duration can be specified in Seconds (float) or frames
 (int).
 With value 0 the playback starts immediate with volume,
 there will be no fade-in at all.
 DEFAULT: 0

[9] end_volume ... End Value of the volume.. (float)
 DEFAULT: 1.0

[10] fade_out_time ... duration for the fade out. In this durationtime the
 input volume changes slightly from volume --> to end_volume.
 Duration can be specified in Seconds (float) or frames
 (int).
 With value 0 there will be no fade-out at all.
 DEFAULT: 0

[11] nloops ... number of repeats
 DEFAULT = 1

[12] from ... start playback range at offest in frames
 [13] to ... end playback range at offest in frames
 [14] framerate ... framerate that is used for transformation frames to
 seconds
 (specify the original framerate of the video here
 in case when it differs from the master framerate)
 DEFAULT: the value specified by VID_MASTER_FRAMERATE

AUD_PLAY_MOVIE

This record is used for playback of portions of an audiotrack in an input
 videofile,
 with optional fade effects. Please note that all audiotracks
 are mixed to one composite audiotrack.

(1) Record Key ... AUD_PLAY_MOVIE
 (2) track ... integer tracknumber
 (3) file ... filename of a video (must contain the audiotrack).
 (4) start_sec ... start playback range at offest in seconds (float)
 (5) end_sec ... end playback range at offest in seconds (float)
 [6] volume ... adjust the volume. (float)
 A value of 1.0 keeps the original input volume for this
 track
 value 0.5 results in half, value 2.0 in double volume.
 (the result may be additional affected by the
 MASTER_VOLUME,
 that is applied to the mixed composite audio)
 DEFAULT = 1.0

[7] start_volume ... Start Value of the volume. (float)
 DEFAULT: 1.0

[8] fade_in_time ... duration for the fade in. In this durationtime the
input volume changes slightly from start_volume --> to volume.
(int). Duration can be specified in Seconds (float) or frames
With value 0 the playback starts immediate with volume,
there will be no fade-in at all.
DEFAULT: 0

[9] end_volume ... End Value of the volume.. (float)
DEFAULT: 1.0

[10] fade_out_time ... duration for the fade out. In this durationtime the
input volume changes slightly from volume --> to end_volume.
(int). Duration can be specified in Seconds (float) or frames
With value 0 there will be no fade-out at all.
DEFAULT: 0

[11] nloops ... number of repeats
DEFAULT = 1

[12] seltrack ... Select input audiotrack (for videos with more
audiotracks such as multilingual DVDs.
only use 1 for the first audiotrack, or if the video has
one audiotrack)
DEFAULT: 1

[13] decoder ... optional name of preferred audio/video decoder that
should be used for the videofile read access.
"libmpeg3"
"libavformat"

AUD_SILENCE

This record is used to define Pauses in an audiotrack.

(1) Record Key ... AUD_SILENCE
(2) track ... integer tracknumber
[3] duration_sec ... duration of the silent pause in seconds (float)
DEFAULT: 1
[4] wait_until_sec ... wait silently until this timestamp is reached,
(then do the unconditional pause for duration time)
timestamp in measured in seconds (float)
DEFAULT: 0

Examples:

a) play the the sound "hello.wav",
then wait silent until the total playtime has reached 60 secs.
then start the sound "ringing.wav".
if "hello.wav" is longer than 60 secs there will be no pause,
(and a warning is reported)

```
AUD_PLAY_SOUND track:1 hello.wav
AUD_SILENCE    track:1 delay:0.0 wait_until_sec:60.0
AUD_PLAY_SOUND track:1 ringing.wav
```

b) play the sound "hello.wav",
then wait 4 seconds and start the sound "ringing.wav" exactly at sec. 60.0

```
AUD_PLAY_SOUND track:1 hello.wav
AUD_SILENCE    track:1 delay_sec:4.0 wait_until_sec:0.0
AUD_PLAY_SOUND track:1 ringing.wav
```

Filenames:

Filenames can be prefixed by absolute pathnames,
or by relative pathnames. Relative pathnames are relative
to the location of the storyboard_file.

Both / and \ are treated as directory Separator.

Videotracks:

The Videotracks in a Storyboard_file are Input tracks
and will be merged internal to one composite Videotrack.

The composite frame image for the output Videotrack is built
as Layerstack, where each videotrack represents one Layer
(except mask track 0 and tracks where the current frame is VID_SILENCE.
in such cases no layer will be created in the composite frame image)

The tracknumber of video tracks defines its position
in the layerstack. Per default video track 1 is placed on
top of the Layerstack.

You can set VID_OPACITY to value 0.0 (or less than 1.0) to let
other tracks shine through.

If the Input Frame of a Video Track has an alpha channel,
the layers below are visible at the transparent Regions
of its alpha channel, even if VID_OPACITY is set to 1.0 (full opaque)
If there is no layer below, black pixels are shown
at full transparent regions.

Notes:

- Inverse order that places video tracknumber 1 in the background
can be specified via VID_MASTER_LAYERSTACK
- You can add a layermask to Input frames by using
masks. (If the original frame already has an alpha channel
only those pixels are visible that are enough opaque in both the mask
and the original alpha channel)
- video tracknumber 0 is reserved used for mask definitions.

Clips with tracknumber 0 therefore are not placed as layers in the composite frame image.
Those Mask clips can be refered in Video Clips (via mask_name) for adding a layermask to provide individual transparency areas.

Sections:

Storyboard files always have a main section, optionally indicated by the Record Key MAIN_SECTION and can have additional sub sections.

For the output video rendering only the clips in the main section are processed. Sub sections are definitions of clip sequences that can be refered from the main section via VID_PLAY_SECTION as if the were a simple clip.

Note that Clips in a SUB_SECTION not rendered in the output video if there is no reference via VID_PLAY_SECTION in the MAIN_SECTION.

Both main section and sub sections can have multiple tracks.

Examples:

creating a video with a complex intro scene built up

```
SUB_SECTION section_name:"Intro"      (complex multitrack)
SUB_SECTION section_name:"Finale"    (complex multitrack)
```

A sub section starts with the Record Key SUB_SECTION.
If there are no Records with Record Keys SUB_SECTION or MAIN_SECTION all clips are implicate placed in the main section.

Mask definitions have global scope for all sections.

audio tracks in SUB_SECTION are not supported,
and will be ignored.

Framerange:

A Framerange is the selected portion of Frames (or Layers).
It is specified by two numbers, FROM and TO
and the optional keywords {"pingpong" or "normal"}
where both numbers are included in the Framerange.
(example: from:3 to:6 results in playback 3,4,5,6)

FROM and TO can be specified as integer numbers (10 ==> unit is frames or layers)

If the FROM value is greater than the TO

value, the Framerange is processed backwards.

The "pingpong" mode extends the Framerange in the way that the frames are played in both directions.

(example: from:1 to:4 pingpong results in playback 1,2,3,4,3,2)

The repeatcount repeats the Framerange N times.

(from:1 to:4 pingpong repeatcount:2 results in playback 1,2,3,4,3,2, 1,2,3,4,3,2)

Audiorange:

A Audiorange is the selected portion of a soundfile (or audiotrack of a movie) that is to play.

It can be specified by

- two float numbers (1.0 = 1 second)

Macrofiles:

Macros can be used to perform a user defineable set of one or more gimp procedures (plug-ins), such as

- color correction
- sharpen / blur
- transformations
- artistic effects
- or whatever ..

on the handled frames of an input framerate in a video track.

The macro is performed before the frame is added to the composite frame.

Apply Macrofiles with varying parameters:

A single macrofile only allows applying an effect with constant settings to all processed frames.

If an effect shall be applied with variable settings this can be done by using a 2nd macrofile that describes alternate parameter sets, and by defining the duration (e.g number of frames see macsteps parameter) that shall be used to vary the iterable parameters from the values defined in the 1st set to their final alternate value as defined in the 2nd macrofile.

The 2nd macrofile is implicate referenced by naming convention via the keyword .VARYING (as suffix or before the extension)

Example:

Macrofiles:

```
VID_PLAY_MOVIE track:1 file:"/vol1/videos/A.mpg" from:000001 to:000500 \  
    macro:"/vol1/fmac/my_blur_macro.fmac" \  
    macsteps:200 \  
    macaccel:10
```

This example `implicite` refers to a second macrofile:

```
"/vol1/fmac/my_blur_macro.VARYING.fmac"
```

If such a file exists, it is used as provider of alternate parameter sets for correlating filtercalls.

Filtercalls in the alternate (2nd) macrofile where filtername and occurrence count matches to filtername and occurrence count in the first macrofile are correlating and will be used to build a mix of the iterable parameters.

Processing for frame 000001 of the input video clip A.mpg will apply the parameter values of the 1st macrofile.

As processing continues, the influence of the values defined in the 2nd macrofile grows.

At frame 000200 (macsteps:200) the final value, as defined in the 2nd macrofile is reached, and will be used for all further frames of this video clip too (201 upto 500).

The speed of the change from settings in the 1st and the 2nd macrofile depends on the specified acceleration characteristic (macaccel:10) where 10 defines increasing speed.

Use value 1 for constant speed (e.g linear changing settings)

Filtercalls in the 2nd macrofile that have no correlating pendent in the 1st macrofile are ignored. Filtercalls of the 1st macrofile that have no pendent in the 2,nd macrofile are executed with constant parameter values in all processed frames.

Example:

```
*  
* # macrofile1:      len  filter parameter data  
* -----  
* 1. plug_in_abc    3     00 01 02  
* 2. plug_in_xy     2     00 00  
* 3. plug_in_abc    3     00 08 00  
* 4. plug_in_xy     2     0A 0A  
* 5. plug_in_xy     2     02 02  
*  
* # macrofile2:      len  filter parameter data  
* -----  
* 1. plug_in_zzz    1     00  
* 2. plug_in_xy     2     00 20  
* 3. plug_in_xy     2     00 00  
*  
* In this example the plug_in_zzz is ignored (because there is no correlating  
* filtercall definition in macrofile1)  
* macrofile2:line 2.) correlates with macrofile1:line 2.)  
* macrofile2:line 3.) correlates with macrofile1:line 4.)
```

RESTRICTION:

Macrofiles:

- Applying filters (e.g plug-ins) with varying values is restricted to filters that provide the capability to access the filter specific parameter settings (typically stored as last values buffer in memory) by either implementing an iterator procedure or registering the structure of this last values buffer in the configuration file lastval_descriptions.txt

This allows GIMP-GAP to handle the mix of filter specific parameter values. (see also Filter->Filter all Layers)

Macrofiles for global usage:

Another method allows performing of a macro with global scope (e.g. affecting all frames). This can be done by specifying the macro directly in the "Master Encoder" dialog. In that case the macro is performed on the already flattened composite frames immediate before they are passed to the encoder engine.

What are Macrofiles and how to create them ?

GAP Filtermacro_files
include a user defineable list of filtercalls,
where each filtercall defines a set of filter specific parameters.

You can Record any GIMP Filter Plug-In that works on a single drawable and is able to run with LAST VALUES in a filtermacro_file.
(use the Menu Filter->Filtermacro)

NOTES:

- gimp-gap filtermacro files are machine dependent, and GIMP / FILTER Version dependent.

filtermacro support may be dropped in the future.

Supported video input

-
- libmpeg3 based MPEG1, MPEG2 (DVD) Video
libmpeg3 can read most videos from DVD.
there is support for typical .vob and .ifo files
and fast exact random access to the videoframes.
(libmpeg3 native tocfiles are used for access.
GIMP-GAP API does create such tocfiles
on demand when videoindex creation is done)

See docs of libmpeg3 decoder for detailed list.

- libavformat based MPEG1, MPEG2 (DVD), MPEG4 Video, AVI, WMV, Quicktime, ..
libavformat can read many videofileformats and codecs
but can not handle .ifo files.
Exact random access is slower than libmpeg3,
because it is handled in the GIMP-GAP API
(GIMP-GAP videoindex creation is handled
completely outside the library)

See docs of ffmpeg decoder for detailed list of
supported fileformats and codecs.

LIMITS:

- Video Tracknumbers are limited to integer numbers within
the range from 0 to 19.
- Both / and \ are treated as directory Separator in filenames.

How To

HOW TO-do-lossless-MPEG-cut

=====

The GIMP-GAP FFMPEG encoder can do (nearly) lossles MPEG cut.
This should give same quality as the original
when your input videos are already compressed MPEG videos.
(such as DVD's)

The lossless MPEG-cut is still an experimental feature,
and may sometimes produce output videos that do not
playback properly on all players.

To use lossless MPEG-cut GIMP-GAP must be configured
with:

```
--enable-videoapi-support  
--enable-libavformat
```

Create a STORYBOARD file, with an editor or with the
Storyboard dialog, containing the desired clips of your videofiles.

Lossless encoding becomes possible if the STORYBOARD file uses only
one videotrack and the input clip(s) are MPEG or MJPEG ecoded videos
in the same size as defined in the VID_MASTER_SIZE record
Further the processed clips must not have any transformations
(such as ZOOM, MOVE operations and MACRO executions).

Another requirement is that both decoder and encoder
supports handling of frames as compressed video chunk data.

Currently the libmpeg3 and libavformat based decoders support this feature (by implementing the procedure GVA_get_video_chunk)
The encoders FFMPEG and AVI1 support handling of compressed video chunk data when the parameter "Dont Recode Flag" is set.

a) lossless MPEG1 / MPEG2 cut

=====

Use the Master Encoder to encode your STORYBOARD file

(Menu: Video->Encode->Master Encoder)

Select the FFMPEG Encoder
and set its Parameters like this:

Select an MPEG video codec that matches your input video(s)
"mpeg2video" if your input clip(s) are .vob or .ifo file from DVD
"mpeg1video" if your input clip(s) are MPEG1 strams from VCD

Dont Recode Flag: ON (checked)
Intra Only Flag: ON (checked)

The Dont Recode Flag in the GIMP-GAP FFMPEG encoder forces copying of frames 1:1 from input video clips where possible, and does call the FFMPEG encoder engine for all the other frames.

This way the user may cut videoclips at any desired frame, without respecting intra frame picture group boundaries.

If there is a cut that does not start with an intra frame the encoder engine will be used until the next intra frame with matching size is read in sequence.

Starting from this intra frame it can copy the already encoded frames 1:1 from the input MPEG video to the output MPEG video as long as the desired frames are fetched in ascending sequence with stepsize 1.

If the quality settings for the FFMEG encoder are different from th quality of the input videoclip(s) this will be visible especially at those points, where the encoder switches from encoding engine to 1:1 copy mode.

b) lossless MJPG cut

=====

Lossless cut for MJPEG encoded videos is supported

by the AVI1 encoder.

Set the "Dont Recode Flag" in the JPEG-Options of the AVI1 encoder and select "MJPG" as Video CODEC. to trigger the lossless cut feature of the AVI1 encoder.

Lossless 1:1 copy of frames is possible if the input clips are JPEG images or MJPEG encoded videos (typically JPEG images packed into an AVI file).

The lossless MJPEG cut was tested with MJEG encoded AVI videoclips produced with the OLYMPUS SP560UZ digital camera.

How to make a GIMP Plug-In an "Animated" one

This README_developers file should be read by Plug-In authors:

HOWTO-write-animated-plug-ins.txt

How to make a GIMP Plug-In an "Animated" one:

 First of all:

```

the plugin must be able to operate on a single Drawable,
-----
further it must be able to run with last_values and in interactive mode.
-----
```

For the Animated Plugin Call we need an Iterator Procedure. This Procedure is named like the Plugin with the extension "_Iterator". The Iterator Procedure has to know the Plugin's internal datastructure and has to modify Its "LastValues" step by step at each call.

The GAP-PDB-Browser <Image>/Filters/Animation/Filter All Layers checks for the existance of Iterator Procedures in the PDB

- 1.) <plugin_name>_Iterator
- 2.) <plugin_name>_iter_ALT

If one of the iterator procedures was found in the PDB, the "Apply Varying" Button is set sensitive to allow animated calls for the selected procedure.

You can generate the "plug_in_XXXX_Iterator" for a Plugin as separated Sourcefile, ready to compile. (The Plugin must be found in the PDB at generation time)

this example uses the Plugin whirlpinch

1.a # for bourne and ksh users:

```

GAP_DEBUG=y
export GAP_DEBUG
```

How to make a GIMP Plug-In an "Animated" one

```
1.b # for csh users
    setenv GAP_DEBUG y

2. # change to the directory, where you like to generate the sourcecode
   cd /usr/local/gimp-0.99.17/plug-ins/whirlpinch

3. # start Gimp
   gimp

4. # open or create any image
   File->New

5. # call the GAP Plugin for Animated Filter apply
   # from within the image
   Filters->Animation->Filter all Layers

   # if you have set GAP_DEBUG as shown above,
   # the Window should contain the Button "Gen Code by name"

   # Type the name of your Plugin into the Search Inputfield
   # (Click "Search by Name" Button, to check if the Plugin
   # is available in the PDB)

   # Then click "Gen Code by Name" Button and then "Cancel" Button.

   # This will generate 4 files in the current directory:
   gen_filter_iter_forward.c
   gen_filter_iter_tab.c
   plug_in_whirlpinch_iter_ALT.inc
   plug_in_whirlpinch_iter.c

   # You can quit gimp and delete the 3 files named "gen_filter_iter_*"

6. # compile and link the generated Source, plug_in_whirlpinch_iter.c
   # and install the linked executeable
   # in global or private plug-in directory

   gimptool --install plug_in_whirlpinch_iter.c

   # (if you dont have gimptool, you can use a copy of the original Plugin's
Makefile
   # and change the Plugins <name> by <name_iter> to do that job.)

7. # start gimp again,
   # and open or create an Image that has at least 3 Layers.
   # Test the "Animated Filter apply"

   Filters->Animation->Filter all Layers

   # Use the "Apply Varying" Button,
   # it should be sensitive now (if all went well so far).

8. # In case of error:
```

How to make a GIMP Plug-In an "Animated" one

```
# If you get an Error Message (in the shell, where you started gimp)
# that looks like:

    ERROR: xxxx_Iterator stored Data mismatch in size N != M

# you have to change the generated code manually.
# (check for calls to "gimp_set_data" or "gimp_get_data" that are using
# the plugins name as key argument within the plugin's sourcecode.
# The passed datastructure has to match exactly in size with the generated one
# for the example above the generated structure is:
# plug_in_whirl_pinch_iter.c:
    typedef struct t_plug_in_whirl_pinch_Vals
    {
        gdouble    whirl;
        gdouble    pinch;
        gdouble    radius;
    } t_plug_in_whirl_pinch_Vals;
```

If you are the Author of a Plugin you may decide to include the `_Iterator` within the original Sources of your Plugin. In that case you have to check the name argument in the run procedure.

Example Code:

```
query()
{
    static GimpParamDef args_plugin[] =
    {
        {PARAM_INT32, "run_mode", "non-interactive"},
        {PARAM_IMAGE, "image", "the image"},
        {PARAM_DRAWABLE, "drawable", "the drawable"},
        {PARAM_INT32, "value", "my_parameter value"},
    };
    static gint nargs_plugin = G_N_ELEMENTS (args_plugin);

    static GimpParamDef args_iter[] =
    {
        {PARAM_INT32, "run_mode", "non-interactive"},
        {PARAM_INT32, "total_steps", "total number of steps (# of layers-1 to apply the
related plug-in)"},
        {PARAM_FLOAT, "current_step", "current (for linear iterations this is the
layerstack position, otherwise some value inbetween)"},
        {PARAM_INT32, "len_struct", "length of stored data structure with id is equal
to the plug_in proc_name"},
    };
    static gint nargs_iter = G_N_ELEMENTS (args_iter);

    static GimpParamDef *return_vals = NULL;
    static gint nreturn_vals = 0;

    gimp_install_procedure("plug_in_XXXX,
                          "plug_in_XXXX can do ....",
                          "",
                          "Authors Name",
```

How to make a GIMP Plug-In an "Animated" one

```
    "Copyright",
    "Date",
    "<Image>/Filters/Effects/XXXX",
    "RGB*, INDEXED*, GRAY*",
    PROC_PLUG_IN,
    nargs_plugin, nreturn_vals,
    args_plugin, return_vals);

gimp_install_procedure("plug_in_XXXX_Iterator,
    "This extension calculates the modified values for one
iterationstep for the call of plug_in_XXXX",
    "",
    "Authors Name",
    "Copyright",
    "Date",
    NULL, /* do not appear in menus */
    NULL,
    GIMP_PLUGIN,
    nargs_iter, nreturn_vals,
    args_iter, return_vals);
}

static void p_delta_long(long *val, long val_from, long val_to, gint32 total_steps,
gdouble current_step)
{
    double    delta;

    if(total_steps < 1) return;

    delta = ((double)(val_to - val_from) / (double)total_steps) *
((double)total_steps - current_step);
    *val = val_from + delta;
}

static void
run (const gchar    *name,
     gint           n_params,
     const GimpParam *param,
     gint           *nreturn_vals,
     GimpParam     **return_vals)
{
    static GimpParam values[1];
    GimpPDBStatusType status = GIMP_PDB_SUCCESS;
    GimpRunMode       run_mode;
    gint32            len_struct;
    gint32            total_steps;
    gdouble           current_step;

    long    pval;
    long    *pval_from, *pval_to; /* plug_in_XXXX has one parameter of type long */
                                     /* values for 1.st and last layer
                                     * when they were processed by
"plug_in_gap_layers_run_animfilter"
                                     */
```

How to make a GIMP Plug-In an "Animated" one

```
*nreturn_vals = 1;
*return_vals = values;
values[0].type = GIMP_PDB_STATUS;

run_mode = param[0].data.d_int32;

if (strcmp (name, "plug_in_XXXX") == 0)
{
    .... /* start the plugin itself */
}
else if (strcmp (name, "plug_in_XXXX_Iterator") == 0)
{
    /* Iterator procedure is usually called from
    * "plug_in_gap_layers_run_animfilter"
    * (always run noninteractive)
    */
    if ((run_mode == GIMP_RUN_NONINTERACTIVE) && (n_params == 4))
    {
        total_steps = param[1].data.d_int32;
        current_step = param[2].data.d_float;
        len_struct = param[3].data.d_int32;

        if(len_struct == sizeof(pval))
        {
            /* get _FROM and _TO data,
            * This data was stored by plug_in_gap_layers_run_animfilter
            */

            gimp_get_data("plug_in_XXXX_ITER_FROM", pval_from);
            gimp_get_data("plug_in_XXXX_ITER_TO", pval_to);

            p_delta_long(&pval, *pval_from, *pval_to, total_steps, current_step);

            gimp_set_data("plug_in_XXXX", &pval, sizeof(pval));
        }
        else status = GIMP_PDB_CALLING_ERROR;
    }
    else status = GIMP_PDB_CALLING_ERROR;
}

values[0].type = GIMP_PDB_STATUS;
values[0].data.d_status = status;
}
```

Important for Plugin_Authors:

I have made Iterator Procedures for more than 50 existing Procedures.
(see `gap_filter_iterators.c` and subdirectories `iter_ALT/*/*.inc`)
If your Plugin is found in `gap_filter_iterators.c`, and You make updates
to your Plugin's interface, you should write (or generate) your own `_Iterator`
Procedure,
to keep its Animated capabilities intact.
(You don't need to change gap sources to do that, because the Iterator
named "plug_in_XXXX_Iterator" is used rather than "plug_in_XXXX_Iterator_ALT")

Video Configuration (gimprc)

```
-----  
Video Configuration (gimprc)  
-----
```

There are optional configuration values for the GIMP-GAP video plug-ins. Those values are stored in your personal gimprc file (located in \$HOME/.gimp-2.6/gimprc) or in the system-wide gimprc file (usually located at /usr/local/etc/gimp/2.6/gimprc)

If you edit gimprc files by hand, you must do this before you startup GIMP.

```
# configure where to find the executable mplayer program.  
# use the full name including the absolute path.  
# This is required if mplayer is NOT found in one of the directories  
# specified in PATH but is installed somewhere else.  
# (relevant only on UNIX systems)  
(mplayer_prog "/usr/local/bin/mplayer")
```

```
# configure where to find the executable wavplay program.  
# use the full name including the absolute path.  
# This is required if wavplay is NOT found in one of the directories  
# specified in PATH but is installed somewhere else.  
# (relevant only on UNIX systems)  
(wavplay_prog "/usr/X11R6/bin/wavplay")
```

```
# there are 2 Values to define direcorey and basenane  
# for Video Copy/Cut/Paste operations  
(video-paste-dir "/home/hof/gap_video_paste_dir")  
(video-paste-basename "gap_video_paste_")
```

```
# the video-confirm-frame-delete can have values "yes" or "no"  
# where "no" does not show the confirm dialog windows when the user  
# tries to delete video frames of the video.  
# (default is "yes")  
(video-confirm-frame-delete "yes")
```

```
# the gap video API keeps an internal cache for  
# the specified number of frames at read access from  
# videofiles.  
(video-max-frames-keep-cached 50)
```

```
# the gap player supports caching of frames  
# (both image sequences or frames fetched from videofiles)  
# to achieve fast playback speed when playback  
# is repeated in loops or pingpong mode.  
# a cache size of 0 turns caching OFF.  
# the cache size can be set in kilobytes (K) or megaytes (M)
```

```

(video_playback_cache "100M")

# the gap player supports caching of gimp tiles
# note that frame playback does NOT use gimp_tiles (see video_playback_cache)
# but caching of gimp tiles is relevant for other tile based processing features
# (especially detail tracking that accesses the same region of gimp tiles in many
loop iterations)
# the cache size is specified in number of tiles.
(video_player_cache_ntiles 200)

# the gap player has several widgets to set the
# position (e.g. the currently displayed framenumber)
#
# - the "Go Buttons" are an array of buttons that allow
#   positioning by simply moving the mouse over the buttons.
#   in some situations this may be annoying to some users
#   therefore this optional way of positioning can be turned
#   on/off
#   via the gimprc option video_player_show_go_buttons "yes" or "no"
#   (default is "yes")
# - the position scale widget can be turned on/off
#   via the gimprc option video_player_show_position_scale "yes" or "no"
#   (default is "yes")
(video_player_show_go_buttons "yes")
(video_player_show_position_scale "yes")

# videoindexes are files that store seekoffsets for keyframes
# and enable fast and frame exact random access to frames in a videofile.
# if gimp-gap should create such files automatically when needed
# you should configure the value "yes".
#
# if you want gimp-gap to ask each time before creating videoindex file
# then configure value "ask".
# if you want GIMP_GAP to ask even in case where creating a videoindex is not
required
# because native timecodebased seek works, then configure "ask-always".
# Note: the "ask-always" setting is rather annoying for productive usage, but
# useful for testing and development purpose.
# (ask is the default when nothing is configured)
# if gimp-gap should not create such videoindex files at all
# configure value "no"
(video-index-creation "ask")

# configure where to store and check for video index files
# if video-index-dir is not configured
# the GAP video API uses a directory named gvaindexes
# in your gimpdirectory by default
(video-index-dir "/path/to/writeable/directory")

# If the gimp-gap videoapi uses libmpeg3 there is a built-in
# workaround for a libmpeg decoder specific bug that sometimes
# causes crashes when MPEG1 videos are closed.
# The workaround is implemented for UNIX OS only and
# forks a child process with its own signal handler.

```

Video Configuration (gimprc)

```
# the child process tries open/close. Crashes are then caught
# in the signal handler of the child process to tell the parent
# process (via retcode) that this videofile cant be handled safe by libmpeg3.
#
# This workaround makes open videofiles safer but slow.
# (mainly caused by additional open/close rather than by forking).
# setting the gimprc parameter
# (video-workaround-for-libmpeg3-close-bug "no")
# can skip the workaround, and makes opening of viedofiles faster
# with some risk of crashing when libmpeg3 is used.
#
# per default (without this gimprc parameter) the workaround is enabled.
#
(video-workaround-for-libmpeg3-close-bug "no")

# optional external audioconverter program
# must be able to handle the same parameters as the
# standard converter script audioconvert_to_wav.sh
# There is NO need for the audioconvert_program gimprc configuration
# if you use the standard converter script.
(audioconvert_program "/usr/local/bin/my_private_audioconverter")

# Configure GIMP-GAP for explicite use of the UFRaw 3rd party plug-in
# for non-interactive loading of frame files from raw filetypes.
# configuration (gap-load-ufraw-%s "yes")
# Use this when the generical gimp-file-load fails to load Canon .cr2
# or other raw files
# (because it picks the file-tiff-load plug that is not capable to load those
filetypes)
(gap-load-ufraw-dng "yes")
(gap-load-ufraw-nef "yes")
(gap-load-ufraw-cr2 "yes")
(gap-load-ufraw-cr "yes")

# NON-XCF frame processing/export overwrite behaviour:
# -----
# if you want to use GIMP-GAP with other fileformats than XCF
# for your frames, you will get a warning dialog at the first
# attempt (per GIMP-session) to save one of the frames with GIMP-GAP plug-ins.
# If you dont want this warning dialog, you can use the gimprc
# file to configure how GIMP-GAP should save frames depending on the
# extension.
# for all configured fileformat extensions GIMP-GAP uses the configuration from the
# gimprc file and does not show the warning dialog.
# configuration key is video-save-mode-for-%s where %s stands for an extension
# such as jpg, tif, cr2,....
# possible configuration values are:
# "overwrite"
# automatically overwrite the discfile image that corresponds
# to the current frame (which is opened in the gimp session)
# on frame exchange opertions (such as go to next/previous frame)
# where the file save plug-in is called with the background layer as active
layer.
#
```

Video Configuration (gimprc)

```
# "overwrite_flattened"
#   automatically overwrite the discfile image that corresponds
#   to the current frame (which is opened in the gimp session)
#   on frame exchange operations (such as go to next/previous frame)
#   where the current frame is flattened before calling the file save plug-in.
#
# "readonly"
#   frame exchange will fail in case the current frame image has
#   unsaved changes (e.g. the dirty flag is set).
#   GIMP-GAP does not overwrite frame files on disc in this mode.
#   It is intended for fileformats where export/overwrite
#   is not available or not wanted.
#
# "readonly_discard"
#   GIMP-GAP does not overwrite frame files on disc in this mode,
#   but discards unsaved changes without any notice on frame exchange.
#   USE THIS ONLY WHEN YOU KNOW WHAT YOU ARE DOING ON YOUR OWN RISK !
#
# Here are recommended gimprc configuration settings
# for some common used fileformats:
(video-save-mode-for-jpg "overwrite_flattened")
(video-save-mode-for-jpeg "overwrite_flattened")
(video-save-mode-for-bmp "overwrite_flattened")
(video-save-mode-for-ppg "overwrite_flattened")
(video-save-mode-for-gif "overwrite")
(video-save-mode-for-dng "readonly")
(video-save-mode-for-cr2 "readonly")
(video-save-mode-for-nef "readonly")

# old way (supported for backwards compatibility)
#   where "yes" is the same as video-save-mode-for-%s "overwrite_flattened"
#   and "no" is the same as video-save-mode-for-%s "overwrite"
(video-save-flattened-jpg "yes")
(video-save-flattened-jpeg "yes")
(video-save-flattened-bmp "yes")
(video-save-flattened-png "yes")
(video-save-flattened-ppm "yes")
(video-save-flattened-gif "no")

# video storyboard layout options
# -----
# layout options for the storyboard dialog including
# number of rows/column and thumbnail sizes for both
# cliplist (upper) and storyboard (lower) thumbnail area.
#
# thumbnail_size supported values are:
# "large"
# "medium" (default)
# "small"
#
# row and column settings:
(video-storyboard-thumbnail_size "medium")
(video-storyboard-columns "5")
(video-storyboard-rows "6")
```

Video Configuration (gimprc)

```
(video-cliplist-thumbnail_size      "medium")
(video-cliplist-columns             "12")
(video-cliplist-rows                "2")

# force storyboard clip playbacks to the aspect defined
# in the storyboard file properties.
# if turned off playback clips according to their original pixelsize.
(video-storyboard-force-aspect-playback "yes")

# maximal number of open videofiles for storyboard render
# processor. The storyboard processor typically keeps more than
# one videofile open when processing multiple input video clip
# references within a storyboard. This is done for performance reasons.
# But if there are too many open videofiles this may lead to
# run out of memory and other resources. Therefore the number
# of simultaneous open videofiles is limited to a default
# of 12.
# The integer parameter video-storyboard-fcache-size-per-videofile
# sets the number of frames to be cached per open video handle
# for backwards playing video clips. (it is not relevant for
# the standard case where video clips are read in ascending frame sequence)
#
# If you have less than 1GB of memory, you may use a smaller
# value (minimum is 2) when you are running out of memory
# while storyboard processing.
# You still are able to process storyboards with many different
# videoclips but this will be slower because there will be more
# open, seek and close operations required with a smaller limit.
(video-storyboard-max-open-videofiles 12)
(video-storyboard-fcache-size-per-videofile 36)

# The boolean parameter video-storyboard-preview-render-full-size
# sets the size for storyboard rendering for preview purpose
# (in the player widget that is embedded in the storyboard dialog)
# the value "no" picks the minimum of current preview size and storyboard size
# for rendering.
# Note that preview rendering is forced to full size (regardless
# to the setting of this parameter) in case the storyboard
# includes at least one clip with fixed width or height.
# (e.g. the storyboard records of the following types:
#  VID_FIT_SIZE mode=none
#  VID_FIT_SIZE mode=width proportions=change_proportions
#  VID_FIT_SIZE mode=height proportions=change_proportions
# )
# The value "yes" uses full original storyboard size for rendering of the preview.
# This may be desired in case the user wants to extract a composite frame
# from the playback widget in the storyboard dialog at full size.
(video-storyboard-preview-render-full-size "no")

# the integer parameter video-storyboard-resource-log-interval
# enables logging of cached resources during storyboard processing.
# value n (positive integer) triggers resource logging every n-th frame.
#     where logging includes information about cached images
#     and currently open video handles.
```

Video Configuration (gimprc)

```
# This is intended for debug purpose.
# value 0 (is the default) turns off this type of logging.
(video-storyboard-resource-log-interval 0)

# the boolean parameter video-storyboard-multiprocessor-enable
# enables multiprocessor support for the storyboard processor
# The current implementation uses a parallel running prefetch thread
# that reads frames from referenced videofiles into a frame cache
# in advance.
# in case num-processors is configured with value 1 the default is "no" (otherwise
"yes")
(video-storyboard-multiprocessor-enable "no")

# the boolean parameter video-enoder-ffmpeg-multiprocessor-enable
# enables multiprocessor support for the ffmpeg based video encoder
# The current implementation uses a parallel running encoder thread
# that uses a ringbuffer queue that is feed by the storyboard processor.
# in case num-processors is configured with value 1 the default is "no" (otherwise
"yes")
(video-enoder-ffmpeg-multiprocessor-enable "no")

# the boolean parameter video-enoder-ffmpeg-show-expert-settings
# defines the initial mode of the FFMPEG based videoencoder Parameter dialog
window.
# where value "no" hides all notebook tabs with details video encoder
# parameters. In this mode the user can set the parameters
# by selecting one of the preset names via a combo box.
(video-enoder-ffmpeg-show-expert-settings "no")

# the video-libavformat-seek-gopsize parameter
# enables native (fast) seek operations in the ffmpeg (libavformat)
# video read access api.
# values <= 0 disables native seek operations.
# values > 4 define how many frames shall be sequentially read
# after seek operation.
# For some videofiles libavformat positioning
# to keyframe seems not to work properly. This will result
# in some trash frames when positioning to B or P frame, because
# the related keyframe required for proper decoding may be skiped
# by the seek operation.
# If you get trashed frames after seek operations you may
# increase the video-libavformat-seek-gopsize parameter
# (values upto 128 can make sense). Note that a high value
# increases the chance to pre-read the releated keyframe and fixes
# the problem with trashed frames, but makes seek significantly slower.
#
# libavformat native seek operates timecode based while gimp-gap
# operates frame number based. therefore positioning via native seek
# may deliver wrong frame positions for videos with individual timing
# per frame. For frame exact operation on such (rare) videofiles
# you should disable native seek by setting the video-libavformat-seek-gopsize
# parameter to 0 and let gimp-gap create a videoindex.
# (videoindex creation requires an initial full scann of the video,
# but offers exact positioning.
```

Video Configuration (gimprc)

```
# positioning via videoindex is slower than native seek)
#
# an internal default value of 8 is used if no other value is configured.
# (this should be OK for most videofiles, if you work with mpeg2 dvd stuff
# .vob files, a value of 4 shall be enough and give optimal positioning performace)
#
(video-libavformat-seek-gopsize "8")

# The gimprc parameter video-libavformat-timecodeolog
# triggers additional timecode logging at video index creation
# it creates a logfile with same name as the videoindex + ".timelog"
#
# The gimprc parameter video-libavformat-timecodeolog
# defaults to "0" where no .timelog file is written.
# *.timelog files are useful for developers
# to analyze and verify native timcode based
# seek operations. *.timelog files contain timecodes and stepsize differences
# for each frame of the analyzed video and CRITICAL warnings where
# variable or non-plausible timecodes were detected.
#
(video-libavformat-timecodeolog "1")

# (video-gva-libavformat-video-analyse-persistent "yes")
# the ffmpeg video access api performs
# analyses on videofiles to detect parameters that
# are required for fast native timecodebased seek
# access and to detect if seek support generally works for a videofile.
# This analyse is normally triggered at the 1st seek attempt
# and takes some time (usually less than 10 secs) because probereads
# and seek tests are performed.
# If the parameter
# video-gva-libavformat-video-analyse-persistent
# is set to value "yes" the results of the analyse is saved to
# a file (located in gimprc subdirectory gvaindexes)
# If the same video is opened another time, the analyse can be skipped
# if the required parameters can be successfully fetched from the file.
# (if the video file was modified since creation of the analyse result
# file the fetched parameters are not used and analyse is NOT skipped)
#
#
# Setting this parameter to value "no" ignores
# anaylse results files (even if they are still valid)
#
(video-gva-libavformat-video-analyse-persistent "yes")

# the api for ffmpeg video access can be configured how to handle
# read errors when decoding video files with the gimprc parameter video-gva-
# libavformat-continue-after-read_errors
# the default option ("yes") is to skip the current package (e.g. the frame that
# cant be read/decoded)
# and try to continue reading the next frame.
# the other option ("no") is to stop when the first read error occurs
# (available since 2010.10.02)
(video-gva-libavformat-continue-after-read_errors "yes")
```

Video Configuration (gimprc)

```
# gimp_gap frame fetcher configuration
# -----
#
# the gap frame fetcher cache is currently relevant for storyboard processing
# and filtermacro processing. it is used to keep the configured
# number of images, videohandles and videoframes in cache for potential
# reuse in further processing steps to avoid multiple load or open/close
# operations.
#
# images are cached as gimp images that are marked with a temporary image parasite
# "GAP-IMAGE-CACHE-PARASITE"
# those parasite identifies an image as member of the gimp_gap frame fetcher cache.
# (gap does not open a display on such images)
# setting the gap_ffetch_max_img_cache_elements too high can result in
# slow processing because the gimp core starts to swap when too much
# or too large images are loaded.
#
# using values up to 100 or more may speed up storyboard processing
# on machines with large memory (4G or more).
# (note: expect speed up only in case the storyboard has multiple references
# to images, such as applying the same mask image to more frames
# or using the same image to render multiple frames)
#
# default value is 18.
#
(gap_ffetch_max_img_cache_elements "18")

# number of video handles to keep cached.
# default value is 6
(gap_ffetch_max_gvc_cache_elements "6")

# number of video frames per video handle to keep cached.
# note that video frames are not cached as gimp image.
# video frames are cached by the gimp gap video api that allocates
# the memory (and will fail in case not enough memory is available to hold the
# configured
# number of video frames)
# default value is 36
(gap_ffetch_gva_frames_to_keep_cached "36")

# gimp-gap has implemented 2 algorithms to locate small detail
# of a layer in a corresponding layer.
# the older implementation is slower but may may be activated
# with this parameter for test purpose
# default is no
(gap-locate-details-use-old-algorithm "no")

# Move Path debug feature to trigger logging current parameters
# while rendering a frame to stdout.
# default is no
(video-move-path-log-render-params "no")

# Move Path rotation threshold is a float number between 0.0 and 1.0
```

```
# This threshold vaule allows Move path render engine to skip the time consuming
# rotate transformations on very small angles.
# default is 0.0125 degree
(video-move-path-rotate-threshold "0.0125")

# enables the debug menu in the storyboard dialog
# and preview image extraction by click on the preview image
# in the storyboard attributes dialog window.
# default is no
(video-enable-storyboard-debug-features "no")
```

About GIMP-GAP

MAINTAINERS

Currently active maintainers

Wolfgang Hofer
E-mail: wolfgangh@svn.gnome.org
Userid: wolfgangh

NEWS

Here is a short overview what's new in GIMP-GAP-2.7.0:

(compared to gimp-gap release 2.6.0)

- New Plug-Ins added:
 - plug-ins to render animated Water and Fire effects.
 - New colormask plug-in added
The colormask filter can apply transparency for pixels matching the colormask image.
intended for processing frames where moving objects can be isolated from the non-moving constant background by applying the constant background as colormask on all frames.
the colormask feature is also available as new mask anchor mode when processing stroyboard clips.
(anchor modes are: ClipColormask, Clip, Master)
 - Foreground Extraction based on the alpha matting algorithm.
This plug-in uses a tri-map provided by the user where pixels are marked as FORGROUND, BACKGROUND or UNDEFINED and renders transparency and color for all UNDEFINED pixels.
 - support to run gimp_color_balance tool as animated filter (added wrapper plug-in).
 - new plug-in to apply the MovePath functionality (transitions and move object along path) in a "process one frame per call" style, intended to be called as filter

with the modify frames feature.

- new plug-in to fill selected (small) areas by blending surrounding colors to cover pixel errors. Intended to fix images and videos shot with a camera that has some defect sensor pixels or dirty lens.
- new plug-in for tracking a significant detail in frames. This plug-in is integrated in the Playback and can record movements of the detail as XML parameters for the MovePath feature. The recorded movements are useful to compensate unwanted shaking of videos recorder without a stativ.
- GIMP-GAP now supports speed control of movements and other transitions via Acceleration characteristic presets. Those presets are available
 - in the MovePath tool
 - in Storyboard transitions and Storyboard filtermacro pair calls when applied with varying values.
 - in animated filter calls with varying values. relevant in Filter All Layers and Modify Frames feature when applying a filter

The GAP dbbrowser now supports acceleration characteristic graph and spinbutton (that replaces the "Apply Varying" button of older GIMP_GAP releases)

- The Storyboard now supports rotation of the processed clips by any angle and more complex transistions similar to the move path feature (based on settings saved with the Movepath dialog as xml parameter file)
- A new storyboard processing feature allows adding external transparency for clip type movie. This is done via format string that refers to the external frames (typical gray image frames) the format string has placeholder %s (replaced by basename of the video) and %06d (replaced by the currently processed frame number)

The format string can be specified in the storyboard master properties dialog

- Added features to the Morph plug-in to support more convenient creation of workpoints and lead to better results (especially for morping tweens of a video)

new plug-in "Video/Morph/Morph Workpoint Generator"

Generates workpointfiles for each frame in the specified frame range.

new feature for the morph plug-in "Video/Morph/Create Tween Frames"

that creates a directory with (renumbered) copies of the processed frames and additionally generated tween frames that are rendered via fade or morphing. individual workpointfiles per processed frame are located via name convention (extension .morphpoints) and used for morphing when they exist. (such files can be created with the help of the plug-in "Morph Workpoint Generator").

- updated gimp-gap video API and ffmpeg-based video encoder to support the libraries provided with the ffmpeg-0.7.11 release This includes various bugfixes related to video de/encode

(but breaks backwards compatibility when seeking positions by frame number in videofiles that do not start with a keyframe).

.. see ChangeLog for details,

- added preset for HD video encoding with H264 codec.
(GIMP-GAP must be built with libx264 to use that preset)
- better performance for ffmpeg based video encoder on single and multiprocessor machines.

- Major Bugfixes

- fixed a bug in the GAP ffmpeg video encoder that resulted in lower quality of the encoded video.
(disabled buggy pre conversion to yuv420P colormodel for the ffmpeg encoder. now feeding RGB to the codec or convert with functions of the encoding engine)
- fixed video encoder crash on 64bit Systems.
- bugfixes in the storyboard processing now use less memory resources and produces smoother movent.
(the old code did run out of memory when processing multiple HD videoclips)

(See ChangeLog for details on further bugfixes)

Here is a short overview whats new in GIMP-GAP-2.6.0:

(compared to gimp-gap release 2.4.0)

- Storyboard undo support
- Storyboard section support. complex scenes can be described within subsections where each subsection can arrange multiple clips in multiple individual video tracks.
- Storyboard clips support new deinterlace modes Odd First, Even First
- Player supports extracting audio track when playing clips that refer to a videofile that has one or more audiotrack(s)
(the extracted audio can be used for audioplayback if audiosupport is enabled and wavplay is installed)
- Filtermacro processing now supports mapping to persistent drawable id references.
This allows applying recorded filtermacros in another gimp session in case the recorded last values buffer data contains references to drawable ids.
such as the drawable for a bump map in the plug_in_bump_map,

or all the layers that were used to map on a quader in `plug_in_map_object`.

The persistent references can refer to anim frames (series of numbered images), multilayer images, or videofiles.

Filtermacros can be applied with fixed or varying values in Storyboard clips. For varying values 2 filtermacros are required to provide from and to values. Iteration of persistent drawable ids is supported in case both refer to the same image, anim frame series or the same videofile.

- video extract supports generating transparency via bluebox effect.
- the master video encoder dialog now stays open while the encoder process is running and shows encoding progress (with Cancel option)
- lossless frame extract and videocut for MJPEG encoded videoclips is supported by the storyboard processor and in the AVI1, FFMPEG and RAWFRAMES encoders (via dont recode flag)
- frontends for old external video encoders and the old xanim based frame extract plug-in are not built per default.
- Modify Frames now supports creation of grayscale layer from alpha channel, or layermask or mix of both
- added wrapper for rotate any angle.
(provides support for animated apply in "filter all layers" or in fltermacros for rotation effects in storyboard clips)
- added wrapper for the resynthesizer
(provides support for animated apply of the 3rd party plug-in resynthesizer)
- external libs for video read and write access were updated to ffmpeg-0.5 and libmpeg3-1.8
- bugfixes and updates for the use with GIMP-2.6.x releases.(see ChangeLog for details)

Here is a short overview whats new in GIMP-GAP-2.3.0:

(compared to gimp-gap release 2.2.1)

- Player support to playback storyboard composite video.
The rendering respects all tracks and effects of storyboard processing and gives an exact preview of the final results.
This feature is triggered via Storyboard Play button pressed with CTRL-Key button held down.
- Player cache with configurable size (via gimprc and Player dialog)
- Player layout

- using icons for the buttons
(that are now smaller and allow more space for the video frame)
- go buttons and position scale are configurable
(via gimprc and Player dialog)
- Storyboard supports overlapping frames within one single track.
this makes it easy to create any transitions effects between 2 scenes
without the need of having multiple video tracks.
- Storyboard supports layer mask (and mask sequences)
that can be applied to any type of videoclip to control
transparency for the masked parts in all the frames of the videoclip.
Mask sequences can be used to make custom transitions effects
- Storyboard dialog supports editing multiple tracks,
where track 0 is used to define layermasks.
The layermask definitions can be applied to any
clip (by drag & drop onto a clip properties dialog window)
- Storyboard support for frame aspect (4:3 or 16:9)
Thumbnails in the Storyboard dialog are now displayed respecting
the aspect to reflect the correct video frame proportion
that may differ from the pixelsize.
- Storyboard clip properties now support built in transitions
to flip horizontally, vertically and rotate by 180 degree.
- Storyboard Layout
 - Global Storyboard Layout settings are now saved (in gimprc)
and restored in further sessions.
- Videofile read API update for use with recent ffmpeg (SVN snapshot 2007.04.04)
The API now supports native timcode based seek and allows fast and exact
positioning without the need of creating a video index.
(**Note:** for some videofiles with variable frametiming and corrupted timecode
information the video index is still required)
- New Plug-in that allows explicite creation of video indexes
for a list of videofiles.
- Some bugfixes (see ChangeLog for details)

Here is a short overview whats new in GIMP-GAP-2.2.1:

(compared to gimp-gap release 2.2.0)

- Storyboard dialog support to edit transition attributes.
The new popup dialog allows creation and changes of the storyboard file
attributes
 - FIT_SIZE
 - MOVE X/Y

ZOOM X/Y and
OPACITY

creation is invoked via menu "create transition" in the story board edit dialog, changes are made via right mouse button click on the thumbnail of the transition attribute.

- drag&drop support in the storyboard dialog.
accept drop of images, frameranges and videoclips.
drag&drop copy and move is now possible to arrange existing clips in the storyboard and/or cliplist.
- Modifiy Frames support of additional functions:
 - set selection from alphachannel
 - invert existing layermask
 - apply filter on layermask
 - set layermodes
(normal, dissolve, multiply, divide, screen, overlay, difference, addition, subtract, darken only, lighten only, dodge, burn, hardlight, softlight, color erase, grain extract, grain merge, hue, saturation, color, value)
- new plug-in to reverse frame sequence.
- wheel mouse scroll support for VCR naviagtor and playback dialogs.

Here is a short overview whats new in GIMP-GAP-2.2.0:

(compared to gimp-gap release 2.0.2)

- support upto 8 framenummer digits for frame images (mplayer style)
(older support was limited to 6 digits)
- Videoapi (GVA) for read access in videofiles (.mpg, .avi)
(frames and audio)
- Videoextract Plug-In
extract frames and/or audio samples
from a videofile, based on the GAP Video API (GVA)
- Videoextract Plug-In based on external program MPlayer
- Storyboard dialog to create and edit
Videocliplist (Storyboard text files)
usable for Encoding Videofiles
with the new Master Videoencoder plug-in.
- Player can run in as widget
in the storyboard dialog and the videoextract plug-in.
- The Player can trace the displayed frames
in a multilayer image.
- Storyboard and Player

can optional use the GAP Video API (GVA)
to access frames in videofiles (.mpg, .avi)

- Master Videoencoder plug-in
This is a Common GUI to
encode GAP framesequences or STORYBOARD files
to one videofile, including audio.
The user can select all registered
Video Encoder plug-ins.
 - Videoencoder Plug-ins
 - AVI Encoder (based on avilib, codecs: RAW, JPG, XVID)
 - FFMPEG Encoder (based on ffmpeg, handles many MPEG based
videoformats and codecs,
see ffmpeg docs for detailed information)
 - SINGLEFRAME Encoder (generate GIMP-GAP typical frame sequences
as output. This may be useful
to convert STORYBAORD Files to frame sequences)
 - Morph Plug-in
generate tween layers with metamorphose transformations
from source layer to destination layer
The user can set controlpoints for the warp part of the
transformation to track details.
 - Extended Onionskin Configuration
allows now bidirectional references.
(refer to frames +n and -n in the same frame)
 - Added new functions to modify frames plug-in
"Copy layermask from layer above"
"Copy layermask from layer below"
 - Active layer tracking by name or stackposition.
This feature tries to set the active layer at framechanges
in the new loaded frame by matching layername (or stackposition)
to the active layer of the previously displayed frame.
 - Move-Path Plug-in:
Additional method for converting gimp-bezier paths to GIMP-GAP
controlpoints in the MovePath tool.
(SHIFT Grab Path allows smoother match to the bezier curve
depending on the number of handled frames)
- Multiple functions of the "Reset All Points" button.
SHIFT: copy settings of the 1.st controlpoint to all other controlpoints
CTRL: mix settings of 1.st and last controlpoint into the
other controlpoints inbetween.

old news

Here is a short overview whats new in GIMP-GAP-2.0.2:

(compared to the gap version that once was part of gimp-1.2.x release)

- new bluebox plug-in makes selected color transparent.
- new features for the "MovePath" plug-in
 - animated perspective transformations
 - optional respect selected areas of the moving source object (with animated adjustable feather_radius)
 - integrated bluebox filter can be applied to the moving source object
 - tracelayer creation
 - collects all scenes from start to current frame in one tracelayer.
 - tween processing for fast moving objects.
 - collects scenes between 2 frames in the tweenlayer.
 - grab controlpoints from current GIMP-path
 - dynamic resize of the preview to follow current window size.
 - optional instant apply for the preview
- playback for video frames to give a motionpreview in thumbnail quality.
On linux the playback has audiosupport to play a wavefile synchron with the videoframes
- the onionskin plug-ins are now integrated to gimp-gap.
onionskin layer(s) usually do show previous (or next) frame(s) of the video in the current frame.
- new features in "frames to multilayer image" plug-in
optional respect selection(s) in the sourceframes
- new features in "frames modify" plug-in
 - new functions to
 - set/delete/invert selection in all handled frames
- renumber plug-in for renaming frames (on disk)
you can set a new starting number for the first frame and define how much digits (leading zeroes) to use for the framenummer part.
- change framedensity
useful for changing the framerate without changing the playback speed. duplicates frames to get higher density or deletes frames to reduce density.
- render filename (or just the number part) to layer
- support of the open thumbnail standard
(thumbnailfiles in png fileformat in \$HOME/.thumbnails that allows better quality than the old .xvpics thumbnail standard)
- filtermacro scriptfiles
assemble filtermacro scriptfiles by picking filtercalls from the current session and execute them later all together at one drawable.

- gimp-gap now uses the preferences value save always/only on changes (gimprc keyword "trust-dirty-flag") for unconditional/conditional implicate save operations when stepping from frame to frame.
- vcr-navigator internal changes (now handles unlimited number of frames)
- some minor layout changes.
- internal reorganisations/cleanup of the sourcecode including porting to new GIMP-2.0 and GTK+2.2 api.

README

=====

This is an unstable developer's version 2.7.0 of the GIMP Video menu.
<https://github.com/GNOME/gimp-gap>

The GIMP-GAP (GIMP Animation Package) is a collection of Plug-Ins to extend GIMP with capabilities to edit and create Animations as sequences of single frames.

Requires:

=====

- gimp 2.6.0 or higher.
 This release was tested with gimp-2.6.6
- Note:** GIMP-GAP provides some features that are able to call many of the GIMP standard filters. (plug-in programs that are shipped with gimp-2.6.6) Those features depend on the tested PDB interface versions and may fail if newer version are used.

- glib 2.8 or higher.
- For full video encoding and decoding support check also the requirements for ffmpeg and libmpeg3.

ffmpeg and libmpeg3 are included as sourcecode tarball in this GIMP-GAP distribution.

- nasm is required only in case when compiling of the (optional) libmpeg3 is desired.
 (nasm for the .asm sources)

see file extern_libs/README_extern_libs for further information about those libs.

Optional libs for decode/encode videofiles:

- libxvid 1.0.2

The free XVID Codec is used for MPEG4 encoding with the gimp-gap AVI fileformat encoding plug-in. (the FFMPEG libs have built in MPEG4 support and do not depend on XVID Codec)

You can get the xvid codec at:
<http://www.xvid.org/downloads.html>

Optional libs for decode/encode videofiles based on ffmpeg

- libbz2
- libfaac
- libfaad
- libmp3lame
- libx264

For general information about GIMP-GAP concepts please read:

docs/[reference/txt/INTRODUCTION.txt](#)

Feature specific details are documented the files
 docs/reference/txt/*.txt

Additional Installation Notes

=====

This GIMP-GAP release includes sourcecode tarballs of external libraries :

- ffmpeg
- libmpeg3

Those libs are built automatically in case their requirements are available. Configuration options for those libs can be set by editing the files:

extern_libs/configure_options_ffmpeg.txt
 extern_libs/configure_options_libmpeg3.txt

Those configure_option files are read by the master .configure script

Installation from SVN source tree:

=====

```
./autogen.sh # includes generation of the configure script ant the ./configure
call
make
make install
```

A correct installation adds the follwing menu trees to GIMP:

```
<Image>/Video
<Filters>/Split Video to Frames
```

An overview of all menus can be found at
docs/reference/txt/[INTRODUCTION.txt](#)

For the generic installation instructions, see the file INSTALL.

Additional notes:

This version of GIMP-GAP is a separated Package since GIMP-1.3.x
and does not compile/run with GIMP-1.2.x or older GIMP versions than 2.4.

Notes:

=====

- GIMP-GAP provides frontend dialogs for:
 - mplayer 1.0

OLD frontends:

- xanim 2.80.1 exporting edition (with the extensions from loki entertainment)
- mpeg_encode (V1.5R2)
- mpeg2encode (V1.2)

The frontends and the external programs need UNIX environment to run.
The programs mpeg2encode, mpeg_encode, and xanim
are old and you may not need them at all since GIMP-GAP-2.2
provides improved alternatives for all of them.
Note that the old frontends are not bulit by default.

If you want to use that old stuff, you should install
xanim 2.80.1 (loki)
mpeg_encode and mpeg_play
mpeg2encode and mpeg2decode

on your system and configure gimp-gap with the option --enable-unix-frontends

It is recommanded to install the program mplayer.
MPlayer does support very much videofileformats.

- The GIMP-GAP playback module provides audiosupport for audiofiles
in RIFF WAV format. The audiosupport is based on wavplay (tested
with version 1.4).

If you want to have audiosupport you must install wavplay
and configure and compile gimp-gap without the
configure option --disable-audiosupport

As far as i know, wavplay is available only for LINUX, but i may
be wrong..

The GIMP-GAP playback module further provides a button for converting

other audiofile formats to RIFF WAV.

If you want to use that feature the external following programs must be installed:

- sox
 - lame
- There is no need to install all those external programs to compile GIMP-GAP.
- GIMP 1.1.4 upto GIMP-1.2.x have included older versions of GIMP-GAP as standard plug-in (named gap) with no need of extra installation.

Bugs

====

Please use bugzilla productname gimp-gap for bugreports.

http://bugzilla.gnome.org/enter_bug.cgi?product=gimp-gap

Before reporting a problem, you may want to see if someone else did already - check <http://bugzilla.gnome.org/buglist.cgi?product=gimp-gap>

Other known Bugs

=====

- some mpeg videos do crash at read attempt.
libmpeg3 1.5.4 has problems in its CLOSE procedure on those videos.
The GVA API has a built in workaround for UNIX operating systems and automatically use the libavformat based decoder in such a case (with a little run-time penalty at opening)

for more details see docs/reference/txt/[gap_gimprc_params.txt](#)
(video-workaround-for-libmpeg3-close-bug "no")

you can produce such problem videos

with the ffmpeg encoder,

and with "MPEG1 high quality" preset in the GIMP-GAP Master Encoder dialog

- random frame access is not possible for all supported video files

with the libavformat based video decoder.

(while testing for some mpeg1 encoded videos frame reads always failed after positioning

the current API detects such "problematic" videofiles and emulates seek by very slow sequential frame read operations)

Have fun,

Wolfgang Hofer

Sven Neumann

XchangeLog**2014-02-27 Wolfgang Hofer <hof@gimp.org>**

- fixed bug in the GAP video API that caused endless loop in ffmpeg based video index creation.
 (p_wrapper_ffmpeg_get_next_frame retcode was not returned to the caller !)
 - video index creator and player dialog:
 video index creation now shows the current frame count in the progress bar.
 This is done because the total number of frames to process is just a guess
 and the percentage is kept at constant level if there are (much) more frames to
 process
 (than assumed in the initial guess.)
 * libgapvidapi/gap_vid_api_ffmpeg.c
 * gap/gap_player_dialog.c
 * gap/gap_video_index_creator.c

2014-02-22 Wolfgang Hofer <hof@gimp.org>

- applied fix for the foreground matting feature as suggested at bug #723818
 The warning "Problem!: final_confidence: %f\n" now shall be printed only in
 critical case.
 TODO: Note that GAP still uses a copy of the old foreground matting implementation
 The already available GEGL implementation will be used in future GAP versions
 targeted for the use with GIMP 2.10.
 * gap/gap_fg_matting.c
 * gap/gap_fg_matting_main.c

2014-01-12 Wolfgang Hofer <hof@gimp.org>

- support for RAW files as "readonly" frame sequences for processing
 as high bit depth video source in MovePath operations or Storyboard Clips.
 Note that the final video encoding (ffmpeg engine) still operates at 8 bit depth
 (This will not change in the near future) but rendering in GIMP
 can be done at higher bit depth.
 This feature was implemented to support rendering videos
 based on RAW frame sequences
 (.cr2 files of a timelapse sequence shot on Canon EOS DSLR camera)
 To use RAW files you need a file load plug in that is capable to load
 RAW files without truncating them to 8 bit depth
 in non-interactive way (e.g. without opening a dialog
 for each processed frame)
 For my tests i used the 3rd party plug-in UFRaw (version 0.19.2)
 but it did not work properly out of the box.
 workaround: I patched the source ufraw-gimp.c by commenting out the line:
 // if (run_mode == GIMP_RUN_NONINTERACTIVE) uf->conf->shrink = 8;
 // because this causes loading at reduced size 1/8
 // (i have no idea what is it good for ??)
 new gimprc parameters

 o) (gap-load-ufraw-%s "yes")
 force explicite call of the UFRaw file load plugin
 whenever GIMP-GAP loads a frame image non-interactive mode.

```

o) (video-save-mode-for-%s "readonly")
# NON-XCF frame processing/export overwrite behaviour:
possible configuration values are:
"overwrite"
automatically overwrite the discfile image that corresponds
to the current frame (which is opened in the gimp session)
on frame exchange operations (such as go to next/previous frame)
where the file save plug-in is called with the background layer as active layer.
"overwrite_flattened"
automatically overwrite the discfile image that corresponds
to the current frame (which is opened in the gimp session)
on frame exchange operations (such as go to next/previous frame)
where the current frame is flattened before calling the file save plug-in.
"readonly"
frame exchange will fail in case the current frame image has
unsaved changes (e.g. the dirty flag is set).
GIMP-GAP does not overwrite frame files on disc in this mode.
It is intended for fileformats where export/overwrite
is not available or not wanted.
"readonly_discard"
GIMP-GAP does not overwrite frame files on disc in this mode,
but discards unsaved changes without any notice on frame exchange.
USE THIS ONLY WHEN YOU KNOW WHAT YOU'RE DOING ON YOUR OWN RISK !
* gap/gap_lib.c
* gap/gap_pdb_calls.c [.h]
* docs/reference/txt/gap\_gimprc\_params.txt

```

2014-01-12 Wolfgang Hofer <hof@gimp.org>

```

- audio resample via external tool (sox) did not work in WINDOWS environment.
The old code was restricted to Unix only.
Fixed the parameter substitution and use g_spawn_sync for the external audio
converter call
that now shall work on other operating systems than unix.
tests on windows with sox as external converter were successful,
but only in case when the external converter was configured with
the full absolute path.
sox is available on Linux, Windows, and MacOSX
see http://sox.sourceforge.net/
(note that this fix was already done 2013-01-16 in my local
environment and already mentioned in the commit message on that date
.. but the file gap/gap_story_sox.c was forgotten to be added in the commit
.. now i recognized that mistake and can provide the fix)
* gap/gap_story_sox.c

```

Complete XChangeLog & GIMP-GAP Documentation at [GNOME/gimp-gap /](#)