

YP

COLLABORATORS

	TITLE : YP		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	YP	1
1.1	YP - The YAFA animation player	1
1.2	YP/Introduction	1
1.3	YP/Requirements	1
1.4	YP/Usage	2
1.5	YP/Usage/Shell start	2
1.6	YP/Usage/Workbench start	3
1.7	YP/Usage/Runtime control	3
1.8	YP/Usage/Display types	4
1.9	YP/Usage/Batch mode	4
1.10	YP/Technical stuff	5
1.11	YP/Technical stuff/Making it efficient	5
1.12	YP/Technical stuff/Protecting the 'illegal' display against Intuition	5
1.13	YP/Technical stuff/How to play 'deep' anims on 8bit displays	6
1.14	YP/YAFA file format	7
1.15	YP/Wildfire	8
1.16	YP/Bugs, History	9
1.17	YP/Copyright, Distribution	11
1.18	YP/Author	11
1.19	YP/Option -pubscr	12
1.20	YP/Option -modeID	12
1.21	YP/Option -info	12
1.22	YP/Option -wait	13
1.23	YP/Option -verbose	13
1.24	YP/Option -stats	13
1.25	YP/Option -nodeactivate	13
1.26	YP/Option -fastblit	13
1.27	YP/Option -scale	14
1.28	YP/Option -fastXPK	14
1.29	YP/Option -firstframe	14

1.30 YP/Option -lastframe	14
1.31 YP/Option -speed	15
1.32 YP/Option -framespersecond	15
1.33 YP/Option -buffers	15
1.34 YP/Option -asynbuff	16
1.35 YP/Option -caching	16
1.36 YP/Option -loadall	16
1.37 YP/Option -zoom	16
1.38 YP/Option -dither	16
1.39 YP/Option -gray	17
1.40 YP/Option -discretergb	17
1.41 YP/Option -discretergb2	17
1.42 YP/Option -module	17
1.43 YP/Option -sample	18
1.44 YP/Option -executable	18
1.45 YP/Batch file example	18

Chapter 1

YP

1.1 YP - The YAFA animation player

YP - The YAFA animation player
v0.96 (Sun 20-Jul-97)

Introduction

Requirements

Usage

Technical stuff

YAFA file format

Wildfire

Bugs/History

Copyright/Distribution

Author

1.2 YP/Introduction

The program YP is a player for YAFA animation files.

It has some features that let your AMIGA become a real multimedia machine:

-play directly from disk

or

preload the complete animation for maximum playback speed

-can use all AGA screenmodes

-has a special batch file mode (for LARGE anim shows!)

-can play frame-synchronous sound effects (ProTracker modules, 8SVX samples)

1.3 YP/Requirements

Hardware

- MC68020 or higher
- AGA chipset
- recommended: FastRAM, HardDisk

Software

- AmigaOS V39 (3.0) or higher
- xpkmaster.library and compressors (LIBS:compressors/xpk?????.library)
if XPK compressed YAFA animations are to be played

1.4 YP/Usage

- Shell start
- Workbench start
- Batch mode

- Runtime control
- Display types

1.5 YP/Usage/Shell start

When launched from shell YP has two modes of operation:

- 1) YP <YAFA file> [options]
- 2) YP -batch <batch file> [general options]

For details about the batch mode see [here](#).

The options are not case-sensitive, and also the order of options and anim filename on the command line is not important.

If you type 'yp -h' or 'yp ?' then YP prints the list of available options. Another handy thing is the ASL filerequester that opens if there is something wrong with the specified YAFA file (if there was no name given or if it was misspelled or if the file is no YAFA anim).

The shell options can be divided into three sections:

General options

The general options let you choose a type of display and other modes that influence the behavior of YP for the complete time it is running.

- pubscr (or -ps) <name>
- modeID (or -mid) <hex number or 'req'>
- info (or -i)
- wait (or -w)
- verbose (or -v)
- stats (or -st)
- nodeactivate (or -nd)
- fastblit (or -fb)
- scale (or -sc)

`-fastXPK` (or `-fx`)

Animation options

The animation options are used to set buffer sizes and other things that are specific for each animation.

```
-firstframe (or -ff) <number>
-lastframe (or -lf) <number>
-speed (or -s) <number>
-framespersecond (or -fps) <number>
-buffers (or -b) <number>
-asyncbuff (or -ab) <number>
-caching (or -c)
-loadall (or -l)
-zoom (or -z)
-dither (or -d)
-gray (or -g)
-discretergb (or -drgb)
-discretergb2 (or -drgb2)
```

Sound options

The sound options are used to make some noise while the anims are played.

```
-module (or -mod) <ProTracker module name> <startframe> <stopframe>
-sample (or -smp) <IFF sample name> <startframe> <stopframe> <channel>
                        <volume> <repeat>
-executable (or -exe) <command> <startframe>
```

1.6 YP/Usage/Workbench start

YP can be launched from Workbench either directly via its own icon or via the "Default Tool"-field of an anim's icon.

All shell options are usable as ToolTypes. The ToolTypes-parsing is not case-sensitive, the leading "-" must be omitted and the long versions of the options keywords must be used. (eq. "fastxpk" instead of "-fx" in the shell)

It is possible to pass several anims to YP at once using the Multiselect-feature of Workbench (hold SHIFT key, single-click anim icons, double-click YP icon). The anims are played one after each other like in batch mode.

1.7 YP/Usage/Runtime control

While YP is running it can be controlled via some keystrokes:

```
F1..F10    set playback speed
Space      pause/continue
s          sleep
n          next anim (in batch mode)
Return     single frame steps (continue with Space)
Help       frame numbers on/off
Esc/LMB    exit
```

1.8 YP/Usage/Display types

In the current version YP offers three display types:

The 'hardware-display'

This display is used by default. It is in fact a low resolution screen of the default monitor (PAL or NTSC depending on the Amiga hardware). The full screen dimensions are 352x290 (PAL) and 176x145 (PAL 2x2 zoom). Smaller anims are displayed in the center of the screen. This display is programmed on hardware level and is currently the most stable one (concerning screen double-buffering and anim batch mode). If the width of the screen is a multiple of 64 (64, 128, 192, 256 or 320) then YP switches to a special AGA mode that uses 64-bit Bitplane-DMA-reads thus making CPU accesses to Chip-RAM much faster. This speedup is most notable with screens deeper than 6 bits (more than 64 colors).

The 'gfxlib-display'

If an anim is in HAM mode or is wider than 352 pixels then YP automatically switches to this display, which can also be done by using the -modeID option. The main advantage of this display is the ability to use any of the AGA screen modes. Anims are centered on the display, the AGA BorderBlank feature is activated. Screen double-buffering is synchronized with the display using a VBlank interrupt, so the anim playback speed depends on the vertical frequency of the respective monitor driver. (see also -fps)

The 'pubscreen-window'

This display type is activated by the -pubscr option. It is programmed in a 'high-level' manner using graphics.library and is not meant to be really useful (yet) as it still has some serious problems like the very slow blitting-routines and the lack of display double-buffering. It could be improved to make use of graphics boards but:

- I don't have any experience in programming for CyberGFX etc.
- I don't have access to such hardware (for program testing).

So if you want to help implementing graphics board support in YP don't hesitate to contact me!!

1.9 YP/Usage/Batch mode

The batch mode is a unique feature that allows a sequence of anims to be played as if they were one large anim. All specified anim files are opened and initialized at program start so that they can be played without a visible gap between each other.

Batch files have the same structure like the command line in single anim mode (but without the program name 'YP' of course!), each line contains a YAFA file name and corresponding anim options, sound options may also be specified here (appended to any of the lines). batch file example

Infect! & WK-Artworks have produced some big anim demos that use YP's outstanding performance:

```
DATAWORLD    -released at The Party 1995
WILD SUMMER  -released at The Summer Party 1996
DATAWORLD 2  -released at The Party 1996
```


These demos are available on Aminet and on several CD-ROM publications.

1.10 YP/Technical stuff

Making it efficient

Protecting the 'illegal' display against Intuition

Built-in XPK routines

How to play 'deep' anims on 8bit displays

1.11 YP/Technical stuff/Making it efficient

Making it efficient

YP is programmed in C (gcc 2.7.0). The most CPU-intensive routines (such as C2P, planar image zooming, Delta-decompression) are coded and carefully optimized in assembler (ASM-One 1.25).

The 'realtime system' in YP has been developed during the last few months. It is based on a VBlank interrupt for counting the speed values and switching the screen buffers, a SoftInterrupt for transferring the frame from the load buffer to the screen buffer (doing C2P, Delta-decompression etc.) and the OS process itself for loading the frames using asynchronous I/O (sends DOS packets to the filesystem).

I spent quite a lot of time on optimizing the player, and I think I nearly reached the limit now. The major bottleneck is the sloooooow Chip-RAM which makes it impossible to play large anims (eg. 640x480x8) at high speed. But this doesn't hurt too much as I think it is simply unreasonable to deal with animations of that size on today's AMIGAs (I'm talking about full screen action and not static presentations or titling).

A friend of mine has a 68060 with lots of Fast-RAM and a fast HD, and he can play 'Dataworld' at 78fps on his A1200! So don't lament that YP runs too slow on your machine...

1.12 YP/Technical stuff/Protecting the 'illegal' display against Intuition

Protecting the 'illegal' display against Intuition

The 'hardware-display' is a method normally used in demos: switch off multitasking and bang in your own copperlists. The 'gfxlib-display' of YP is also not very system conform, it is made with MakeVPort() and MrgCop(), and then comes LoadView() and makes it visible.

These methods are quite efficient because they are on a very 'low level' in the system software. The problem is that YP can't disable the multitasking OS since it has to make use of it for loading and playing. But when the OS is running other tasks can open screens, or the user could switch the screens using <LCommand M>. This would definitely kill YP's display!

I experimented a while and found a good way to prevent other tasks from destroying the 'illegal' display: YP installs it's own input handler at a high priority that 'eats' all inputs, and LockIBase() makes it impossible for Intuition to do anything. This works fine as far as I have tested it, no screen blanker has a chance when YP is running :-)
Of course YP can't use Intuition itself when it has locked IBase, so one has to be very careful when using the -executable option.

1.13 YP/Technical stuff/How to play 'deep' anims on 8bit displays

How to play 'deep' anims on 8bit displays

The task of 'color reduction' or 'rendering' is everything but trivial. Image processing programs usually offer a number of sophisticated algorithms which incorporate histograms and several kinds of dithering. Everybody who has ever rendered images this way knows that it can take quite a while and is surely not suitable for realtime rendering in animation players. There are several ways to render in realtime, take a closer look at your favourite MPEG or QT/AVI player to find some more or less useful ones. I decided not to use dithering or HAM8 in YP because these methods are relatively slow (at least on my 68030!) and in my opinion the results are not worth it. So I tried to do something better and implemented the 'Discrete RGB' renderer which is fast and looks at least as good as some other player's HAM8 display - honestly!

Currently YP offers three ways of rendering:

332 Quantization

This quick'n'simple way to reduce the color resolution is used by default. The RGB components are cropped to 3 (R and G) or 2 bits (B) and are then put together to yield 8 bit pixels.

The result looks rather poor, but it's really fast and gives you an idea of how the anim should look like.

Gray

This one is selected by the option -gray. The RGB components are added using 'luminance weights'. All major color television standards use these weights (30% Red, 59% Green, 11% Blue) which result from the different sensitivity of the human eye for light of these three colors. YP's conversion routines use slightly different weights which has technical reasons and is by the way hardly visible. (3/8 Red, 4/8 Green, 1/8 Blue, note how easily they can be calculated with a few add and shift operations)

The result looks really good, the routines are as fast as the '332' ones, but there are no colors at all! :-)

Discrete RGB

This one is selected by the option -drgb. The RGB components are displayed as separate pixels on screen. The 256 pens are divided into three sections according to the 'luminance weights': 3/8=96 pens Red, 4/8=128 pens Green, 1/8=32 pens Blue. Note the increased color resolution, there could be $96 \times 128 \times 32 = 393216$ colors on screen simultaneously!

The main disadvantage of this method is the reduced lateral resolution. Since two Truecolor pixels appear as six pixels on screen the routines have to skip every third pixel horizontally and every second vertically.

The result looks quite good, the routines are about as fast as the above

mentioned ones, but there are less details visible.

Discrete RGB 2

This one is selected by the option `-drgb2`. It is basically the same as the 'Discrete RGB' above. The difference here is that all Truecolor pixels are actually displayed which makes the screen four times as large as the anim frames (double width and height). The larger screen dimensions can be compensated by simply selecting a higher resolution (using `-modeid`).

The results are amazing, higher screen resolutions give much better colors because of smaller RGB component pixels, and there is no loss of details!

The downside is that the larger screen requires a lot more CPU time (mostly for Chunky2Planar) resulting in a noticeable slowdown.

1.14 YP/YAFA file format

!!! NEW !!!

The 'YAFA-Extensions-2' have been determined recently. They include TrueColor anims (15 and 24 bit) and better sound support. Detailed documentation is to be released soon.

The problem of an effective and fast compression method for TrueColor anims (like Radius Cinepak for AVI/QT) is still unresolved. If anybody has some ideas on this please contact me.

The animation file format YAFA is a try to develop a new standard that is able to supersede IFF-ANIM. The strength of YAFA lies in the great variety of compression options which makes it easy to adapt anim files to certain requirements. Massive compression using XPK may be the goal of the user of a fast machine, while the user of a slower computer prefers faster playback at the cost of less compression.

Delta-compression like it is used in IFF-ANIM and YAFA is useful for 'static camera and some flying objects'-like animations, but it gains nothing at 'crazy camera flight through a tunnel'-like animations. IFF-ANIM offers no alternative here, but with YAFA you can choose among the lots of XPK packers that are available. When it comes to maximizing XPK compression ratios the option to store the frames as chunky images is another switch to try.

YAFA features:

- bit depth 1...8 (2...256 colors, HAM6, HAM8)
- planar or chunky frames for all depths
- optional timecode-per-frame
- optional palette-per-frame
- optional Delta-compression
- optional XPK-compression
- annotations (similar to ANNO in IFF-8SVX)

Software developers!

The YAFA anim file format can be the replacement of IFF-ANIM. Please consider implementing YAFA loaders/savers in your animation software in order to make YAFA more popular and easier to handle. Player programs for graphics cards are needed too.

There is a documentation on the YAFA file format available on Aminet:

doc/misc/YAFA-doc.lha
For further help and support you are welcome to contact me.

1.15 YP/Wildfire

Wildfire - where dreams become virtuality

A Special Effect Program and many more

Wildfire is not only another brandnew image-processing-program -
it's a professional animation-processor with some really new concepts.

It allows you for example the creation and processing of animations in
several formats, the generation of time-dependant special-effects
and the design of large projects containing several animations and
frame-synchronized sound-effects.

Because the common animation-formats on the Amiga have too many
lacks or restrictions we have tried to create a better one: YAFA.
This animation-format for example features many compression-methods
and realtime-zooming.

One of the outstanding features of Wildfire is the fact that it
makes for you no difference if you apply changes to a whole animation
or if you process only one image. A single image is only a
very special case of an animation.

In contrary to common image-processors you can use envelopes instead
of plain numbers as effect-parameters. This makes the easy creation
of time-dependant special-effects possible.
So you haven't to fight against the "do"-loops in an ARexx-script to
get an animation outside your image-processor anymore.

All of the numerous effects can be arranged in any number and
combination. Because the output of any of these effects can be
one of the 8 temporary-buffers Wildfire allows you to
PROGRAM IMAGES in an comfortable way !

Main Features:

- fast and comfortable Converter:
 - input: YAFA, ANIM5/7/8, mpeg, image-sequences
 - output: YAFA, ANIM5/7/8, image-sequences
 - Preview for almost any operation
 - Project-Editor which allows the creation of large and complicated
projects easily using the mouse
 - Effect-Generator featuring time-dependant parameters via envelopes
-

- a stunning library of effect-PlugIns
- some really new effects, real 3D-effects
- (un)loading of PlugIns at any time
- Envelope-Editor
- Timecodes-Editor
- Transition-Maker for easy creation of transitions between two animations
- easy-to-use Player-interface
- Filmstrips to simplify the perfect timing of huge projects
- AnimInfo-function displaying size-profiles etc.
- fontsensitive GUI without MUI
- internal gadget-toolbox which handles mathematical expressions instead of only plain numbers
- calculator which allows for example the export of user-defined symbols to the gadgets in the whole-program
- Drag&Go
(using drag.gadget 0.2 Copyright © 1996 Jörg Kollmann, see bonus drawer!)
- ARexx-port with about 400 commands
- macros, custom-windows, docks
- import of images from other programs
- Developer-Kit (with working example) which allows you to create your own effect-PlugIns
- and many more

Wildfire is SHAREWARE and can be downloaded from Aminet. (gfx/edit)

1.16 YP/Bugs, History

Known bugs

The gfxlib-display still has some problem:

- palette-per-frame can cause problems with double-buffering
(LoadRGB32() needs more time than the VBlank; currently I got no idea how to fix this, please contact me if you know how to switch colors and bitmaps simultaneously in a clean way using graphics.library)

History

v0.96 (Sun 20-Jul-97)

- 15bit 'Discrete RGB' and 'Discrete RGB 2' renderers optimized
- fixed a little bug that could cause (harmless) Enforcer hits when playing anims that were created by some older Wildfire versions
- more statistics output if only one anim was played ("TOO SLOW/FAST")

v0.95 (Wed 09-Jul-97)

- fixed 'Discrete RGB' renderers for odd frame heights

v0.94 (Sun 29-Jun-97)

- YAFA-Extensions-2 determined
- support of TrueColor anims, three different ways of rendering, new options -gray, -drgb and -drgb2
- gfxlib-display now uses Video Overscan dimensions which are not affected by Overscan Preferences

v0.93 (Wed 28-May-97)

- added option -fps
(playback speed (nearly) independent of video frame rate)
- more "-verbose" information
(video frame rate, anim nominal FPS)

v0.92 (Mon 26-May-97)

- gfxlib-display improved:
 - option "-wait" works
 - no more trash between two anims in batch mode
- fixed options parsing behind "-batch" option

v0.91 (Mon 17-Mar-97)

- frame numbers on display (HELP key)
- fixed sample player bug that was introduced in v0.89

v0.90 (Thu 06-Mar-97)

- major speedup for the hardware-display
(thanx to Bernd Rösch for the tip!)

v0.89 (Sat 01-Mar-97)

- code cleanup and optimization
- SeekAsync() fixed (finally?)
- when launched by Wildfire then the requesters open on the Wildfire screen
- single frame step added (RETURN key)

v0.88 (Thu 26-Dec-96)

- added "-fastXPK" unpacking routine for xpkRAKE
(the original routine in the library modifies the packed data, this leads to checksum errors when using YP's "-caching" mode!)

v0.87 (Mon 11-Nov-96)

- Workbench-startup added (featuring multiselect and tooltypes)
- fixed "SeekAsync"-bug
- implemented support for extended Delta-Compression format
(one or two start-frames as delta-to-empty-frame or plain image)

v0.86 (Tue 29-Oct-96)

- '-info' caused returncode 20
-

--nodeactivate' was always on
-fixed internal XPK routines using the original source codes
--fastXPK' is now an official feature

v0.85 (Thu 26-Sep-96)

-Aminet release

(no consistent history available)
many versions with lots of fixes and new features.

first version December 95, used for 'Dataworld' release at The Party 95

1.17 YP/Copyright, Distribution

Copyright

YP is Copyright (c) 1996, 1997 by Michael Henke. All rights reserved.
Commercial use is prohibited, if you want to use it for commercial purposes
you have to get an explicit permission from the author first.

YP is released as FREEWARE. It is allowed to charge a fee to recover
distribution costs, but no profit may be made by selling or otherwise
distributing the program. It is not allowed to distribute YP without this
documentation. Modified versions of the program may not be distributed.

It is not allowed to disassemble or otherwise reverse engineer the program.
If you are interested in the source code please contact the author.

Disclaimer

No guarantee is given nor implied that this program is fit for any use.
The program and the information within this text are provided on "AS-IS"
basis. The entire risk as to its quality and performance is with the user.
In no event will the author be liable for direct, indirect, incidental or
consequential damages resulting from any defect in the program. The author
reserves the right to make changes to the program or the documentation
without notice.

1.18 YP/Author

Main code and YAFA format development

Michael Henke (Smack/Infect!)
Praetoriusstr. 1/205
06124 Halle/Saale
Germany

EMail: epgbd@cluster.urz.uni-halle.de

Additional code and author of "WildFire - The animation processor",
creator of "DataWorld" - the animation demo released at "The Party 1995"

Andreas Maschke (WK-Artworks)
Zenkerstraße 5
06108 Halle/Saale
Germany

EMail: epgbc@cluster.urz.uni-halle.de

1.19 YP/Option -pubscr

`-pubscr (-ps) <name>`

- activates the pubscreen-window mode
- <name> is the name of the public screen, case-sensitive
- this display mode is rather experimental, it does not support double buffering which leads to a more or less annoying 'flickering'

1.20 YP/Option -modeID

`-modeID (-mid) <hex number or 'req'>`

- activates the gfxlib-display
 - the modeID is a hexadecimal number, don't use any prefixes (like \$ or 0x)
 - if 'req' (without the quotes) or an unavailable modeID is specified an ASL screenmode requester opens
 - to get the modeID of a certain screen mode use 'req' and the option `-verbose`
 - examples:
- | | |
|------------------------|-------|
| LoRes | 0 |
| HiRes-Laced | 8004 |
| Multiscan Productivity | 39024 |

1.21 YP/Option -info

`-info (-i)`

- print some information about the anim(s) (like `-verbose`) but don't play
-

1.22 YP/Option -wait

`-wait (-w)`

- open a black screen before the first and after the last anim, continue with <Space>
- useful for video recording because you know exactly when the anim starts
- 'hardware-display' and 'gfxlib-display' only

1.23 YP/Option -verbose

`-verbose (-v)`

- print some information about the anim(s) and also some more warnings

1.24 YP/Option -stats

`-stats (-st)`

- print statistics (played frames, play time, frames-per-second) on exit

1.25 YP/Option -nodeactivate

`-nodeactivate (-nd)`

- do not pause and restore system colors when window gets deactivated
- 'pubscreen-window' only (option `-pubscr`)

1.26 YP/Option -fastblit

`-fastblit (-fb)`

- use faster blitting routine for copying image data into the window, this usually leads to even more 'flickering'
 - 'pubscreen-window' only (option `-pubscr`)
-

1.27 YP/Option -scale

`-scale (-sc)`

- make the window resizeable, the anims are scaled to the current window size which is quite slow (thats why it's optional)
- 'pubscreen-window' only (option `-pubscr`)

1.28 YP/Option -fastXPK

`-fastXPK (-fx)`

- use YP's internal unpacking routines (currently for FAST, NUKE and RAKE) instead of `xpkmaster.library` and the corresponding compressor libraries
- a significant speed increase can be achieved by this because using `xpkmaster.library` results in some overhead:
 - open/close the compressor library for each `XpkUnpack()` (=every frame!)
 - checksum the packed data before unpacking itthe latter function can of course be considered as useful because it avoids crashing the unpacking routine on corrupted data, but if you are sure that a YAFA anim is not corrupted (after playing it without '`-fx`') the risk of crashing YP's unpacking routines should be nearly 0 (ZERO :-)
- the unpacking routines used in YP are based on these compressors:
 - `xpkFAST.library` V1.06 (27-Jul-94) by Christian von Roques
 - `xpkNUKE.library` V1.1 (02-Nov-92) by Urban Dominik Müller
 - `xpkRAKE.library` V1.7 (06-Sep-95) by Karsten Dagefördethey were testet with many different files and seem to work well.
if you encounter problems with them (an anim plays nice without '`-fx`' but crashes with it) then please contact me and also specify which versions of `xpkmaster.library` and the compressor were used for packing.

1.29 YP/Option -firstframe

`-firstframe (-ff) <number>`

- the first frame of the anim to be played, frames before this are skipped

1.30 YP/Option -lastframe

`-lastframe (-lf) <number>`

- the last frame of the anim to be played
- can be greater than the number of frames in the anim, playing will restart if necessary

1.31 YP/Option -speed

`-speed (-s) <number>`

- set the playing speed for the anim, overrides the value contained in the file
- the speed value is the number of video frames that each anim frame is shown, the smaller the value the faster it plays, minimum is 1 (see also `-fps`)

1.32 YP/Option -framespersecond

`-framespersecond (-fps) <number>`

- set the playing speed for the anim, overrides the speed value contained in the file
- due to the strict synchronization of the player with the video frames (double buffering is done by a VBlank IRQ) it is only possible to play at certain discrete FPS values depending on the frame rate of the monitor driver.
here some examples: (FPS = VIDEO_HZ / SPEED)

	speed	1		2		3		4		5		6		7		8	
50Hz (eg. PAL)	FPS		50		25		16.6		12.5		10		8.3		7.1		6.2
60Hz (eg. NTSC)	FPS		60		30		20		15		12		10		8.5		7.5
72Hz (eg. EURO72)	FPS		72		36		24		18		14.4		12		10.2		9

- the specified FPS value will be adjusted to the nearest discrete value if necessary (see `-verbose` output)
specifying `"-fps 15"` for example yields
PAL: 16.6 fps NTSC: 15 fps EURO72: 14.4 fps

1.33 YP/Option -buffers

`-buffers (-b) <number>`

- set the number of frame buffers for the anim, default is 6

- the buffers are needed for smooth playing while loading from disk, the best number depends on several things (average loading speed, file system fragmentation, CPU and DMA speed etc.)

1.34 YP/Option -asynbuff

-asynbuff (-ab) <number>

- set the size of each of the two asynchronous load buffers for the anim, the number is in KBytes, default is 64
- large buffers can make loading faster (depends on drive, filesystem etc.), small buffers reduce memory usage but usually decrease loading performance

1.35 YP/Option -caching

-caching (-c)

- make YP load the complete anim to memory before starting to play

1.36 YP/Option -loadall

-loadall (-l)

- make YP load and XPK-decompress the complete anim before starting to play

1.37 YP/Option -zoom

-zoom (-z)

- perform realtime 2x2 zooming while playing
- 'hardware-display' only

1.38 YP/Option -dither

-dither (-d)

- perform realtime 2x2 zooming with simple dithering while playing
-

·'hardware-display' only

1.39 YP/Option -gray

-gray (-g)

·render a "deep" anim (15 or 24 bits) to 8 bits gray
(see also tech stuff)

1.40 YP/Option -discretergb

-discretergb (-drgb)

·render a "deep" anim (15 or 24 bits) to 8 bits "Discrete RGB"
(see also tech stuff)

1.41 YP/Option -discretergb2

-discretergb2 (-drgb2)

·render a "deep" anim (15 or 24 bits) to 8 bits "Discrete RGB 2"
(see also tech stuff)

·the anim appears four times as large on screen (double width and height)
which is definitely not the same effect like -zoom.
this one affects all display types so you may want to select a high
screen resolution for better results.

1.42 YP/Option -module

-module (-mod) <file name> <start> <stop>

·the specified ProTracker module is loaded to Chip-RAM before anim playing
starts

·module playing starts/stops when the specified anim frames are reached
(NOTE for batch files)

·modules are always loaded using XPK (so YP recognizes uncompressed and
XPK- or PowerPacker-compressed files)

1.43 YP/Option -sample

`-sample (-smp) <file name> <start> <stop> <channel> <volume> <repeat>`

- the specified IFF-8SVX sample is loaded to ChipRAM before anim playing starts
- sample playing starts/stops when the specified anim frames are reached (NOTE for batch files)
- channel is a number 0...3, volume is a number 0...64, repeat is a number: 0=loop forever, other=number of loops
- stereo samples are supported, channel has no meaning then because YP takes two channels for stereo playing automatically
- samples larger than 128 KBytes are not played correctly, this is due to some limitations in the AMIGA audio hardware

1.44 YP/Option -executable

`-executable (-exe) <command> <start>`

- the specified AmigaDOS command is executed when the anim frame <start> is reached (NOTE for batch files)
- NOTE:
 use this with care when YP runs on 'hardware-display' or 'gfxlib-display'!
 the command may not do any output to a shell window,
 it may not do anything with windows and screens.
 (actually the complete Intuition system may not be used!
 see also tech stuff)
- if <start> is 0 (zero) the command is executed before the player's display is opened (it is allowed to use Intuition)

1.45 YP/Batch file example

This is an example of a batch file used by YP:

```
---cut here-----
;
; comment lines have a ';' as first character

; empty lines are allowed too
; every (real) line contains one animation (and maybe anim options)
; sound options can be appended to every animation line
;     NOTE: in batch files the start/stop frames for sounds
;           are frame numbers in the whole anim sequence
;           rather than frame numbers of individual anims
```

```
ram:example1.yafa -c -lf 200 -mod ram:mod1 1 500
ram:example2.yafa -z -ff 20 -lf 400
---cut here-----
```