

HWGPOST

COLLABORATORS

	<i>TITLE :</i> HWGPOST		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	HWGPOST	1
1.1	HWGPOST.guide	1
1.2	HWGPOST.guide/Copyrights & Distribution	1
1.3	HWGPOST.guide/Background	3
1.4	HWGPOST.guide/Overview	4
1.5	HWGPOST.guide/New Features	4
1.6	HWGPOST.guide/Requirements	5
1.7	HWGPOST.guide/Compatibility	5
1.8	HWGPOST.guide/Why?	5
1.9	HWGPOST.guide/Contents	6
1.10	HWGPOST.guide/Installation	7
1.11	HWGPOST.guide/Tutorial	9
1.12	HWGPOST.guide/Support	11
1.13	HWGPOST.guide/FAQ	12
1.14	HWGPOST.guide/Special Notes	14
1.15	HWGPOST.guide/To Do	15

Chapter 1

HWGPOST

1.1 HWGPOST.guide

HWGPOST

This document contains all the information needed to set up and use HWGPOST Beta 11.

Copyrights & Distribution

Background

Overview

Contents

Installation

Tutorial

Support

Common Problems, FAQ

Special Notes

To Do

NOTE: On pre V39 (< Workbench 3.0) AmigaGuide you will see some funny marks on the screen. They are V39 formatting commands which make the text look nicer under >=V39. They do not reduce the amount of information shown in this document.

1.2 HWGPOST.guide/Copyrights & Distribution

Copyrights, Legalese & Sentiments

HWGPOST and its documentation is ©1993-96 by Heinz Wrobel. It is not in the public domain.

Based on previous work by Adrian Aylward, HWGPOST and its documentation have been created by Heinz Wrobel. HWGPOST is dedicated to Joan Thuesen.

I need to explain something. I started out with the post 1.7 sources by Adrian Aylward. He did an amazing job with post up to and including 1.7. I do not claim to have put it all together myself from the the very first byte. But I have thrown out or reworked almost every single line of it and there is not much left that is not replaced or truly unchanged. Adrian Aylward set up many useful concepts. I reworked most of them and threw some out completely for going towards PostScript Level 2.

While it may not look so from the outside, HWGPOST is a major step away from its origins, and so I claim the right to have something new here. Note that I did not take over any code from other sources than post 1.7 except for some fixes to post 1.7 that people sent me (Thanks especially to Tom Rokicki for his fixes and other unrelated things). No GNU code, no IBM code, nor anything else. To be honest, I tried to look at some ghostscript code once for a particular feature, but I found ghostscript so ugly that I threw it out faster than you can say out (I never even found the code for the feature I was looking for). This means, that all that you read in the 'History' file is only related to my work.

Because of all the above, there will be different rules on HWGPOST that start out right now even with this beta version:

- Unlike with post 1.7, I am restricting use somewhat:

For use in other products, commercial or not, or claims of compatibility to HWGPOST to enhance the value of a product, you need to contact me first to work something out.

(Except for Tom Rokicki. He may use HWGPOST whenever he finds it acceptable for AmigaTeX)

- If you are just a user of HWGPOST, consider it Giftware. I like money, too, but it doesn't have to be money. Anything nice will do. Books, Accelerator/Networking boards, Ax000's, HD's, a Sonata Type 1 font, software, non technical things, whatever. If you intend to use HWGPOST and forget about me intentionally, don't expect any support or bug fixes.
 - If you want to use HWGPOST commercially (e.g. to make money by using it in your office), I feel that you should consider this very fact when thinking about a gift. If you think HWGPOST to be good enough to support you making a living, you should value it as such.
-

- This archive may be distributed only free of charge. It may not be modified or ripped apart without my permission.
- ANY USER OF HWGPOST IS REQUIRED TO REGISTER the use of HWGPOST with me via
 - email
 - , or
 - smail
 - .

All important rights reserved, so to speak. This is currently a private undertaking for fun, but I have not yet lost hope that there is maybe some reward in it for me, too.

If you don't like the above, I don't want you to use HWGPOST. In fact I dare to declare use of HWGPOST then outright illegal. You should live with post 1.7 or something else then.

Oh well, if you are one of those using pirated [KS/WB] software, don't bother asking for help. I won't help you and I specifically don't allow you to use HWGPOST either in this case.

Some legal mumbo jumbo I should not forget: PostScript is copyrighted by Adobe Systems Incorporated.

If I forgot to point to any other important trademarks or copyrights, it is not out of malice, but just an oversight. Please point it out to me and I will correct it for the next public release. All trademarks/copyrights belong to their owner and should be respected.

1.3 HWGPOST.guide/Background

Background

What you get here is a PostScript interpreter called HWGPOST. Very soon I may be able to call it PostScript Level 2 compliant now. It is still rather raw. My intention is to abuse you as beta tester.

HWGPOST is project that I started because I had quite some problems with post.library 1.7 (and 1.86 where post.library stayed the same) not handling some PostScript code correctly. So I built on the post 1.7 source. Don't let this fool you. If you look at the separate 'History' file, you will see that this isn't just a beefed up 1.7. It is much more.

HWGPOST is a major step towards PostScript Level 2 already and as time and motivation permits it will hopefully become a R&W book compliant PostScript Level 2 interpreter library with maybe some Display PostScript extensions.

Currently this archive is probably mostly useful to users of 'post.library' up to 1.7 or users of applications like AmigaTeX which

rely on `'post.library'`.

1.4 HWGPOST.guide/Overview

You want to know what you get here, right? Check out what this [←](#) is.

New Features

Requirements

Compatibility

Why?

1.5 HWGPOST.guide/New Features

Major new features for HWGPOST Beta 11

- HWGPOST requires a bug fix to `'mathieeesingbas.library'` not only on 68040 machines with OS 3.1. There is a fix included now, called `'MathSingFix'`. You must run this fix after `'SetPatch'` and before any user software uses `'mathieeesingbas.library'`. Just put a call to the program `'MathSingFix'` as first line into your `'S:User-Startup'`. This should do the job. There are no side effects or resident memory allocations. There is no need to `'run'` this program in the background or redirect any output. Only messed up values in the library base are fixed. This bug fix automagically cleans up these known problems with the library:
 - On ≥ 68020 machines, the multiplication and division calls crashed.
 - On 68040 machines, fpu instructions were never used, even with an active `68040.library`.
- The bug with font encodings not being processed correctly should be fixed.
- PostScript strings passed to the library should be processed more compatibly now and with less need of memory.
- Error handling should be less likely to fail now in obscure circumstances.

Sorry, `setsystemparams` and `setpagedevice` still have to wait.

1.6 HWGPOST.guide/Requirements

Requirements

OS

You need at least version 37, which is equivalent to OS 2.04. Some features require OS 3.1 as minimum version. I recommend OS 3.1.

Memory

As much as possible. HWGPOST tends to need noticeable more memory than `post.library 1.7` for complex PostScript files. You should have at least 2MB to have fun with it.

Harddisk

You don't want to live without it.

1.7 HWGPOST.guide/Compatibility

Compatibility

HWGPOST should be compatible to all applications using `post.library` version 1.7 with the exception of the `'callexfunc'` and some other special non standard operators, e.g. I have found no problems with `'PostView'`. If you have any compatibility problems, please report them to me.

Some applications relying on non language compliant internals of `'post.library'` may need to be slightly adapted.

AmigaTeX

The AmigaTeX PostScript files `'installPSfont.pro'`, and `'pstopk.pro'` in the directory `'TEX:ps'` contain redefinitions of the `'findfont'` operator. If you comment out these definitions and replace the `'init.ps'`, AmigaTeX works well with HWGPOST.

1.8 HWGPOST.guide/Why?

Why this beta?

A lot of stuff is in place within HWGPOST already even though it is probably still far from completion. Actually it is so much that I hope to find bugs with your help now. I do not have test cases and PostScript files anymore to test all the PostScript Level 2 features in

depth. You will read some 'untested' comments in the 'History' file. I have checked for basic functionality with all my changes, but nobody knows where the bugs hide yet. So I am asking you for bug reports to make HWGPOST get better.

I am interested in:

- Enforcer hits, mungwall problems, reproducible crashes.
- deviation of behaviour compared to Adobe Level 2 products.
- Short and precise PostScript example files that break with HWGPOST.

I need your complete computer and OS configuration with any bug report including any non C= WB things that you run like MagicMenu/ReqTools/CpuBlit.

I am also interested in a short note if you have files using specific Level 2 features available with HWGPOST that don't break.

Thank you for your help!

1.9 HWGPOST.guide/Contents

What is included?

'HWGPOST.guide'
This file.

'post.library'
Interpreter using OS IEEE libraries for all Amigas.

'post.library.20'
Compiled for >=68020 using inline coprocessor math.

'MathSingFix'
A required fix for 'mathieeesingbas.library'.

'History'
Informal but detailed overview on the changes since post 1.7.

'init.ps'
A new 'init.ps'. You must use it.

'HWGPOSTResources'
Example declaration of available external resources, e.g. fonts.
Referenced by 'init.ps'.

'PATH_FONT'
Example file to demonstrate automatic font lookup by the resource mechanism. This should be copied into a directory 'ENV:HWGPOST'.

``postlib.h'`
The include file with some infos on how to use new features of HWGPOST ``post.library'`.

``hwgpostlib.doc'`
Docs on how to use the programmers interface of HWGPOST for your own programs.

``post'`
A hacked up post 1.7 frontend to try out HWGPOST. Nothing fancy.

``devs/datatypes/PostScript'`
``devs/datatypes/PostScript.info'`
``devs/datatypes/PostScriptBIN'`
``devs/datatypes/PostScriptBIN.info'`
The datatype descriptor files for \geq OS 3.0. To be placed into ``DEVS:datatypes'`.

``classes/datatypes/HWGPOST.datatype'`
The datatype for \geq OS 3.0. To be placed into ``SYS:classes/datatypes'`.

``devs/dosdrivers/PRTPS'`
``devs/dosdrivers/PRTPS.info'`
Driver files for \geq OS 2.1 to mount the ``PRTPS:'` handler. To be placed into ``DEVS:DosDrivers'`.

``l/prtpost-handler'`
The ``PRTPS:'` handler itself. To be placed into ``L:'`.

Don't ask me for my sources. Currently I have no plans whatsoever to release them in full or part to anyone. After reading through the History file, you might understand why.

1.10 HWGPOST.guide/Installation

Installation

1. Get an overview of what is in the package as described below.
 1. You sure have looked at the chapter
 Contents
 , haven't you?
Before you start to install the files, you should know what they do and where they usually go. That chapter answers some of this.
 2. Check out the ``History'` file. Even if you are not a PostScript wizard, there is probably useful information for you in there. Don't expect to remember and understand everything. Just get a feeling of the kind of information there. Browse through it a little.

2. If you already use a version of 'post.library', save your existing files. This beta of HWGPOST might break your setup and I don't want you to be out in the cold.
 3. I am sure you already have the 'POST:' and 'PSFONTS:' AmigaDOS assigns that are needed for old post 1.7. If not, create two directories for them in a place of your choice and set up the mentioned assigns. While neither assign is truly needed by 'post.library' itself, they are a somewhat useful standard by now and supporting tools and frontends tend to rely on them. The datatype likes to find 'init.ps' in 'POST:' and the default font lookup configurations are set up for 'PSFONTS:'.
 4. Put all your PostScript fonts into 'PSFONTS:'. This includes IBM-style Type 1 PFB files or Type 3 fonts. Do not place AFM or PFM files there. HWGPOST doesn't need them. If you want to keep AFM and PFM files, use subdirectories within 'PSFONTS:' for them.
 5. Set up the complete configuration for post.library as described below.
 1. Put 'init.ps', 'HWGPOSTResources', and (if you want to) the 'post' frontend into 'POST:'. 'init.ps' is a setup file that should always be run before any other PostScript file you want to render or process. Usually, you don't have to change anything here. 'HWGPOSTResources' is used via 'init.ps' and tells the library where to find resources like e.g. fonts.
 2. Edit 'HWGPOSTResources' for your font setup in 'PSFONTS:'. You'll probably need to throw out and add some lines. If you have quite a few fonts in 'PSFONTS:' or want to have this done semi-automatically, you might want to postpone this step and leave the resource file as is for now. Below we will create the needed lines while testing HWGPOST.
 3. If you don't have an environment directory for HWGPOST already, create the directories 'ENV:HWGPOST' and 'ENVARC:HWGPOST'. Put 'PATH_FONT' into 'ENV:HWGPOST'. It sets up the standard font lookup procedure. If necessary, edit it to suit your needs. It shouldn't be necessary for standard setups, though. For a permanent environment setup, don't forget to copy the files into 'ENVARC:HWGPOST', too!
 6. Copy your choice of 'post.library[.20]' into 'LIBS:'. The 20 version is compiled for a 6888x mathematical coprocessor and for >=68020 CPUs. Don't try to use it if you don't have an FPU. It will refuse to open in that case. The standard library should run on any Amiga.
 7. Shut down any running applications that might be using 'post.library' and do an 'avail flush'.
 8. If you run >= OS 3.0, you can install the datatype files. You might want to reboot after doing so to be on the safe side. The 'devs/datatypes...' files belong into 'DEVS:datatypes' and the 'classes/datatypes...' files belong into 'SYS:Classes/datatypes'.
-

9. If you want to, you can install the 'PRTPS:' handler now, too. The handler files go into 'L:' and the devs/dosdriver... files into 'DEVS:DOSDrivers'. For OS 2.04, you'll need to hack up a mount file from the included driver file. Let me suggest that you rather upgrade to OS 3.1. It is well worth it.
10. HWGPOST requires a bug fix to 'mathieeesingbas.library' not only on 68040 machines with OS 3.1. There is a fix included now, called 'MathSingFix'. You must run this fix after 'SetPatch' and before any user software uses 'mathieeesingbas.library'. Just put a call to the program 'MathSingFix' as first line into your 'S:User-Startup'. This should do the job. There are no side effects or resident memory allocations. There is no need to 'run' this program in the background or redirect any output. Only messed up values in the library base are fixed. This bug fix automagically cleans up these known problems with the library:
 - On >=68020 machines, the multiplication and division calls crashed.
 - On 68040 machines, fpu instructions were never used, even with an active 68040.library.

1.11 HWGPOST.guide/Tutorial

Tutorial

1. You have completed the Installation as described in this manual, right? See
Installation
, for more information.

2. Let's see if the frontend works by starting it in a shell:

```
post ?
```

You should get a terse list of options. Not very informative, is it? Type another '?' and hit ENTER. The list should now be a lot more descriptive.

3. We will do a semi-automatic font setup now via the 'enumfonts.ps' PostScript program that is included. This step assumes that you have at least one Type 1 font file in 'PSFONTS:'. If you don't, you simply won't see any output. In any case, you'll see if the library works ok by running this PostScript program. Try this in a shell:

```
post init.ps enumfonts.ps
```

For every Type 1 font that HWGPOST can find and use, you'll get a suitable line to be put into 'HWGPOSTResources' as output. If you redirect this output into a temporary file, you can replace the demo contents of 'HWGPOSTResources' with this file after removing any duplicate font entries.

If you get any error messages, you need to check your configuration steps again. If files cannot be found for some reason, you might want to try the 'SnoopDos' tool from Aminet to see what is going wrong. Send

email
, if you really can't make it

work.

4. Ok, in the previous step we saw that the library works. It is time to display something.

```
post init.ps screen uncpattern.ps
```

A configuration display of the hacked frontend will show up on a separate screen. You can chose printer or IFF output, densities, page size, and color depth. Right now, simply click the 'OK' button. The frontend should have picked up your configured printer page size as default.

The library will now be running the startup files. After processing uncpattern.ps, it will go into pause mode. You should have a simple uncolored image made of a star pattern in the bottom left area of the page display.

Use the menu 'Control/Continue after pause' menu item or RAMIGA-C to continue, and then RAMIGA-Q to quit the program.

5. In general it is not a great idea to run the file you want to render as init file for interactive use. The typical startup line for the frontend is more like:

```
post init.ps screen
```

You would use the 'File' menu after selecting your processing options to process files. To change these options again, you would select the 'Project/Restart' menu item.

6. For noninteractive use, i.e. if you want to process a file and be done with it, you can of course put it into the command line. To simply print a PostScript file without any special options, you could use this command line:

```
post init.ps file.ps
```

This will try to pick up your PrinterGfx preferences settings, process file.ps, and send the output to the printer. Fairly simple, isn't it?

7. A detailed description of the hacked up post frontend is not available in this archive, i.e. the docs are lousy. You might want to experiment a little. If you don't like the frontend, let me

suggest to get 'PostView' from Aminet. It is a fairly nice frontend for HWGPOST.

8. Try HWGPOST and send me detailed bug reports with reproducible examples!

1.12 HWGPOST.guide/Support

Support

What should a bug report look like?

=====

I prefer bug reports created with the C= 'report' tool. You can find this tool on Fish Disk 920. Any bug report must contain step by step instructions to reproduce the bug on a standard WB setup. Telling me "You need to install this pack of special OS hacks to make the bug appear or those non standard libraries from somewhere" usually leads to no result. If it is a bug related to a PostScript program, I'd need a short example program to reproduce it. If the bug is dependent on the exact layout of the program (e.g. with eexec problems), you should think about using lha and uuencode to preserve any spaces, tabs, newlines, or other special characters in the file that might otherwise be lost in email.

AN ABSOLUTE MUST is the output of 'C:VERSION FULL' (or a version comment in the file or printed in a window if 'C:VERSION' does not yield useful results) and 'C:LIST' for any program involved. This includes but is not limited to:

- The OS (Use 'ShowConfig')
- The frontend ('post', 'postband', 'postview', etc)
- The library 'post.library'
- The support files 'init.ps', 'HWGPOSTResources'
- For printing, the printer driver

If you don't give me this information with each bug report, I might decline to look at it.

Oh, and please don't swamp me with hundreds of kilobytes of email without asking first. Thank you.

My Address

=====

Heinz Wrobel
Karlstr. 16
82131 Gauting

Germany
 FAX +49 89 850 51 25 (I don't like phone calls)
 <heinz@hwg.muc.de>

1.13 HWGPOST.guide/FAQ

Common problems, Frequently Asked Questions

This will evolve into a list of problems people run into that are definitely not bugs in HWGPOST.

PS files fail because the name 'SymbolEncoding' is not known.

=====

Throw out the 'Symbol' font you have, buy the Adobe Type Manager and/or install a working Symbol font. You are using a Symbol font that violates Adobe PostScript specs by relying on an external definition of SymbolEncoding. This name does not exist in the standard. Your current Symbol font is broken.

A font does not work right or causes errors.

=====

Have you preregistered it correctly in 'HWGPOSTResources'? If you depend on the 'PATH_FONT' feature, is the file name truly correct including case? Have you double checked?

NOTE: If there is a bug with a font, I will need the font to fix it. If it is a commercial font, you have to use

smail

, to send

me one legal copy for my own use if I don't have it already. I can't and won't buy fonts just to get your problems fixed.

A PS file causes 'undefinedresource' errors.

=====

The most likely cause is that a font could not be found.

Depending on your setup do some of this:

- Check what fonts are needed (by reading the PS file or checking on the font search with a tool like 'SnoopDos')
- If you don't have the font, you need to get it.
- Check if the entry in 'HWGPOSTResources' is correct if applicable.
- Check if 'ENV:HWGPOST/PATH_FONT' is set up correctly if applicable.
- Set '@DefaultFontName' (as in 'init.ps') to a default font name of your choice. Of course this font must exist!

Always remember that filenames can be changed, PostScript fontnames must match.

The error 'page size not set in preferences' pops up.
=====

You need to run the 'PrinterGfx' prefs editor and set the limits to something other than 'Ignore or Multiply'.

Is there an easy way to install new type 1 fonts?
=====

Yes. Copy your new fonts into 'PSFONTS:'. Then do the steps outlined below.

1. Run the frontend and the C:sort command

```
post init.ps enumfonts.ps >t:allfonts
sort t:allfonts t:sortedfonts
```

2. Now check 't:sortedfonts' and 'HWGPOSTResources' for duplicate fonts or fonts that you don't want to have installed.
3. Insert whatever is left of 't:sortedfonts' into your HWGPOSTResource file.

That's it. If it doesn't work then, it's most likely the font or the document that is broken.

I only get dots printed instead of letters.
=====

Quite likely you don't have the font needed installed. At least it isn't installed correctly. So HWGPOST reverts to its default setting of a dummy dot font. This default setting is done in init.ps. it can be changed of course. Check which fonts are needed with e.g. the 'SnoopDos' tool and install them as described above. This should get you going.

My Amiga crashes with a Guru 8000000B!
=====

The 'mathieeesingbas.library' in the OS ROM is buggy. You need to run a patch to fix this library. The patch, 'MathSingFix' is included. See

Installation
, for more information.

The characters in my fonts are always filled completely.
=====

Check the font files with 'C:TYPE HEX'. There are some public fonts out there that were incorrectly generated with TypeSmith. They violate chapter 3.5 on page 27 of the Adobe Type 1 Font Format description 1.1, the Adobe Black Book.

Somehow I can't find the right solution to my problem here.

```
=====
I admit that the documentation is not that great. You get what you
pay for. Search through the 'History' file as well. Among all the
technical stuff you'll find quite a few hints on how to do things
right. If even this doesn't help, just send some
    email
    .
```

1.14 HWGPOST.guide/Special Notes

Special notes

Special note on the 'post' frontend

```
=====
The included 'post' frontend is a major (ugly) hackup of the original
(ugly) source to suit my testing needs. It may or may not do what you
expect and it might or might not be compatible to the original 'post'.
Anything can happen. If you don't get the output you expect, please
send me a detailed bug report and fall back on 'PostView' or the old
'post' frontend for now.
```

To get more information on the supported options try 'post ?' and
type another '?' at the command line template. For gray scale displays,
play with the 'MAXDEPTH' option. On an A2024, 'MAXDEPTH=2' and
'DISPLAYID 0x41000' give rather nice results compared to the standard
B&W setup.

You won't be able to use your display hardware to the fullest as the
hacked up frontend does not check the display database at all.

Special note on the 'HWGPOST.datatype'

```
=====
To make the datatype work, you need to have 'post.library'
available, and an AmigaDOS assign 'POST:' to where 'init.ps' is located
for best results.
```

The datatype will render all PS files per default with 75 dpi in B&W.
If no 'BoundingBox' comment can be found in the file header the data
type will currently default to an A4 page size (this might change
though). You should read the History file for now to find out what kind
of PostScript files the datatype will accept. Reading the History file
might not be a bad idea in any case.

You can override this behaviour by setting an environment variable.
These are the docs taken straight from the History file:

- The datatype now optionally handles color displays,
densities and other default sizes. How? Via a new

environment variable: "ENV:HWGPOST/DATATYPECONFIG"

The first line of this environment variable is evaluated in DOS ReadArgs() style with this template:

```

COLORS/K/N,BPC/K/N,XDEN/K/N,YDEN/K/N,DENSITY/K/N,
DEFWIDTH/K/N,DEFHEIGHT/K/N,MODEID/K

COLORS:      Either 1 or 3. Only B&W or RGB supported!
BPC:         Bits per color. Anything from 1 to 8. Note
             that BPC*COLORS must be <= 8 or strange things
             may happen due to the OS limit of 8 bitplanes.
XDEN,YDEN:   Densities (dpi). Default is 75 dpi.
DENSITY:     Convenience operator to set both densities.
DEFWIDTH,
DEFHEIGHT:   If no bounding box is specified in the file,
             these values are used. You need to specify
             them in units of 1/100 inch though, not in
             1/72 inch like PostScript likes it. The
             default values for these parameters will give
             an A4 like bitmap size.
MODEID:      A hex screen mode id (e.g. 41000 for an A2024)
             to override the OS settings or the default
             monitor.

```

This environment variable should do the job for a while.

Note that the datatype will always display only the first page of any file. This is intentional as there is no reliable, easy, or fast way to render an arbitrary page.

I tested the datatype only with OS 3.1.

Special note on 'PSPRT:'

=====

I haven't much to say about this one. If it works for you, fine. I hacked it up as better test bed for my replacement 'queue-handler' quite a while ago even though the date doesn't show it. It might not even work for you. Currently I see it as a major hack. There are no config options whatsoever and is B&W only. Colour would eat some amount of memory and it doesn't do strip printing. I included a simple mount file for OS >=2.1 to be placed into 'DEVS:DOSDrivers'.

1.15 HWGPOST.guide/To Do

My TODO list

Well, there is still lots of stuff to do for me as time permits:

- Finish up Binary Object Sequences
- User path operators

- Missing CCITT and DCT filters
 - Device parameters
 - Insideness operators
 - Missing CIE color spaces
 - Whatever else is needed for a languagelevel of 2
 - Bitmap font support and maybe some more DPS extensions
 - TrueType font support
 - Some basic PDF support
-