

psprt-handler

COLLABORATORS

	<i>TITLE :</i> psprt-handler		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 9, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	psprt-handler	1
1.1	psprt.guide	1
1.2	copyright	2
1.3	introduction	2
1.4	requirements	3
1.5	limitations	3
1.6	installation	3
1.7	printing	4
1.8	configuration	5
1.9	post.library	8
1.10	psfonts	9
1.11	troubleshooting	9
1.12	psprttest	11
1.13	history	12
1.14	credits	13
1.15	author	13

Chapter 1

psprt-handler

1.1 psprt.guide

=====

P S P R T - H A N D L E R

v1.32

=====

Copyright © 1994,1995 by Daniel Weber

TABLE OF CONTENTS

Copyright, Distribution and Disclaimer

Introduction

Requirements

Limitations

Installation

Printing a PostScript File

Configuration

post.library

PostScript Fonts

Troubleshooting

psprttest.ps

History

Credits and Thanks

Contacting the Author

1.2 copyright

PSPRT-Handler and associated utility software is
Copyright © 1994,1995 by Daniel Weber. All rights reserved.

PSPRT-Handler is freely redistributable.

THIS PROGRAM AND ITS DOCUMENTATION MAY BE DISTRIBUTED FOR NON-PROFIT PURPOSE ONLY. IT MAY NOT BE MODIFIED IN ANY WAY WITHOUT THE PRIOR WRITTEN PERMISSION OF THE AUTHOR. USE AT YOUR OWN RISK. NO WARRANTY. NO REFUNDS. NO CARRIER.

IT IS ILLEGAL TO DISTRIBUTE THIS PROGRAM ON DISKS WHICH COST MORE THAN US \$5 PER FLOPPY DISK, OR MORE THAN US \$50 PER CD.

1.3 introduction

Introduction

PSPRT-Handler allows you to print PostScript files on a no-postscript-capable printer using the
post.library
.

My motivation behind this project was the need to print PostScript files on my printer. However I do not have a postscript printer and therefore I searched for another solution - without success. The PSPRT-Handler was born...

Please note that the printing of a PostScript file using a software based PostScript interpreter takes more time than a real PostScript printer does. To speed up the printing, for example, you can decrease the density and set the dithering type to 'ordered' (using the Prefs/PrinterGfx program).

To have a feeling how long a print job can last, you have the possibility to open a progress indicator. The progress indicator is a small window in the top left corner of the default public screen with a bar which shows you how much of a file has been processed so far. See

Configuration

for more details about how to enable and disable the progress ←
indicator.

1.4 requirements

Requirements

- PSPRT-Handler requires an Amiga with at least 2 MBytes of memory.
- Kickstart and Workbench 2.04 or higher are also required.
- As PostScript interpreter the
post.library
is used.
- A printer.

1.5 limitations

Limitations

The current version of the PSPRT-Handler does only support black and white printers. Colour printers will be supported in a later version. At the moment any colour PostScript files will be printed in black and white.

The PSPRT-Handler needs a temporary file to print a PostScript file. Once a day this will be handled more dynamically.

1.6 installation

Installation

The Installation of the PSPRT-Handler is quite easy...

- 1) Put the PSPRT-Handler into L:
 - 2a) If you are using Workbench 2.1 or later,
copy the PSPRT and PSPRT.info files to the
DEVS:DOSDrivers/ directory.
 - 2b) If you do not use Workbench 2.1 or higher you
will have to append the entry in mountlist.psp_{rt}
to your DEVS:mountlist.
Then edit the file S:User-Startup by adding the line
mount PSPRT:
to mount the PSPRT: at boot time.
 - 3) Put the psprt.prefs file into the ENV: assign, and
edit it to suit your needs. For a permanent setup,
copy it into ENVARC: assign too. This file contains
the
configuration
for the
-

PSPRT-Handler.
 For the default settings see
 Configuration
 .

Note that if you do not have the
 post.library
 installed
 correctly, the PSPRT-Handler won't work.

The PSPRT-Handler searches the init.ps file (that comes along with
 the
 post.library
 in the POST: or the L: assign.

The PSPRT-Handler interprets and prints a postscript file with a
 task priority of 0. This can slow down your whole system.
 However, to change the priority you simply have to set the priority
 argument in the dosdriver file respectively the mountlist to your
 desired value (e.g. Priority = -1).

1.7 printing

Printing a PostScript File

To print a PostScript file you just have to send it to PSPRT: instead to PRT:.
 This can be done by either

COPY your_postscript_file TO PSPRT:

or

TYPE >PSPRT: your_postscript_file

or by selecting the PSPRT: instead of PRT: as output device in your printer
 utility, texteditor or whatever.

The PSPRT-Handler writes to a temporary file while printing a postscript file.
 By default the 'T:' assign is used as directory for that file. Using the
 optional preference file 'psprt.prefs' this directory can be changed to suit
 your needs (see

configuration
).

If the PostScript interpreter reports an error while printing, a requester will
 be opened to inform you. Please refer to the documentation of the interpreter
 for more detailed information about the error.

While printing a file a progress indicator can be optionally enabled or disabled
 (see

configuration
 for more details).

Usage

PSPRT:<options>, where <options> can be:

`fit` scale interpreted postscript data in a way that a page fits the printable area on the physical page.
This options works together with the following paper sizes: US Letter, US Legal, European A4, Narrow Tractor, and Wide Tractor.

`force` adjust DPI values according to the current width and height of the printable area.
See also
Configuration
.

If no <options> are given, the PSPRT-Handler prints the graphics using the settings for density and page size of the Workbench preferences.

NOTE that you cannot use the 'force' and 'fit' options together.

Examples:

```
a2ps test.c >PSPRT:fit
```

```
copy demo.ps to PSPRT:
```

1.8 configuration

Configuration

The PSPRT-Handler supports now a configuration file, that let you customize the handler to suit your particular wishes. This configuration is automatically loaded by each print job. This allows you to change settings between the print jobs.

The PSPRT-Handler does not complain if the file could not be found or an error occurred while reading and processing the configuration file.

You do not need a configuration file if you are satisfied with the default settings. The default settings are:

```
temporary files are written to T:,  
and the progress indicator is disabled.
```

The handler looks for configuration file PSPRT.PREFS in the ENV: assign.
For a permanent setup, you should copy the file into the ENVARC: assign too.

The current version of the PSPRT-Handler still supports the PSPRT_TMPDIR environment variable. But the path for the temporary files defined in this variable is overridden by the 'TempDir' keyword from the psprt.prefs config file.

The configuration file itself is a standard ascii text file that can be edited using a texteditor.
In the following you find a keyword overview and a small sample config file.

Keyword	Syntax	Description
tempdir	tempdir "directory"	Sets directory for temporary files (Default directory: "t:"). Directory names containing whitespaces must be enclosed in double or single quotes.
progressbar pb	progressbar pb	Opens a progress indicator when printing a file, a bar which shows you how much of a file has been processed so far (turned off by default).
nopprogressbar nopb	nopprogressbar nopb	Disables the progress indicator (default).
fit	fit	Scales interpreted postscript data in a way that a page fits the printable area on the physical page. This option works together with one of the following paper sizes: US Letter, US Legal, European A4, Narrow Tractor, and Wide Tractor.
nofit	nofit	Disables data scaling (default), see 'fit' description above.
windowx	windowx <xpos>	Sets X position for progress indicator window (default: 16). Example: windowx 0
windowy	windowy <ypos>	Sets Y position for progress indicator window (default: 16). Example: windowy 10
priority	priority <number>	Sets the task priority of the handler to <number>. The range of <number> is between -128 and 127. It's suggested that you do not set the priority higher than 10. Example: priority -1
initpsfile	initpsfile <filename>	This option can be used to specify the path and filename of the init.ps file.

		It can be useful if you do not have the <code>init.ps</code> file in either the <code>post:</code> or the <code>l:</code> assign. Example: <code>initpsfile s:myinit.ps</code>
<code>usepreferences</code> <code>useprefs</code>	<code>usepreferences</code> <code>useprefs</code>	This option forces the <code>psprt-handler</code> to use the preferences to determine the printable area. Using this option you have also the possibility to set specify the area by setting the margins, paper length, and print pitch using the printer preferences program. See
	Troubleshooting .	
<code>memvlen</code>	<code>memvlen <size></code>	Sets virtual memory size for <code>post.library</code> to <code><size></code> (default: 50000). (*) Example: <code>memvlen 75000</code>
<code>memflen</code>	<code>memflen <size></code>	Sets font cache size for <code>post.library</code> to <code><size></code> (default: 60000). (*) Example: <code>memflen 100000</code>
<code>memllen</code>	<code>memllen <size></code>	Sets path line size for <code>post.library</code> to <code><size></code> (default: 10000). (*) Example: <code>memllen 15000</code>
<code>memhlen</code>	<code>memhlen <size></code>	Sets memory for halftones for the <code>post.library</code> to <code><size></code> (default: 20000). (*) Example: <code>memhlen 25000</code>
<code>density</code>	<code>density <number></code>	Sets the printer density to <code><number></code> . <code><number></code> is an integer value from 1 to 7. A value of zero (0) can be used to drop back to the default preferences density. If this option is not given, or an illegal value is specified, the default preferences density is used. A density of one (1) is the lowest density. Example: <code>density 7</code>
<code>pagedots</code>	<code>pagedots <x> <y></code>	Sets the dimension of the page in dots. <code><x></code> is the width and <code><y></code> the height of the page. Use the options marked with <code>'(*2)'</code> to manipulate the page. Note that if the dimension, specified with this option, is too small, the resulting output may be cropped. Example: <code>pagedots 2324 3198</code>
<code>aspect</code>	<code>aspect</code>	This option causes that one of the dimensions may be reduced/expanded

		to preserve the aspect ratio of the print. (*2)
center	center	This option tells the printer driver to put the image between the left and right edge of the paper. (*2)
force	force	This option forces the handler to adjust the DPI values according to the current width and height. This is useful when the postscript page is larger than the printable area of your printer. Give it a try when the printable area is too small for the postscript page. See
	Troubleshooting	.

(*): Please refer to the
 post.library
 documentation for
 more details about these memory sizes.

(*2): These options may only be useful if the page size is set using the
 'pagedots' option.

All keywords are case-insensitive (for example: TEMPDIR, tempdir, TeMpDiR). For any not given keyword, the PSPRT-Handler uses the default setting for that option.

Any comments in the configuration file must start with a semicolon (';').

Example:

```

;
; PSPRT-Handler configuration file
;

tempdir    "t:"
progressbar
priority    -1
usepreferences

```

1.9 post.library

post.library

The `post.library` is a software based PostScript interpreter, that is used by the PSPRT-Handler. The `post.library` must be installed correctly to ensure a smooth printing.

Version 1.7 of the `post.library` can be found on Fish #828 or Aminet.

A newer version, well actually a complete new version, that is a major step towards PostScript Level 2 and that solves some problems with the older `post.library` v1.7, is Heinz Wrobel's implementation. It can be found on Aminet as `HWGPOSTbeta6` (`text/print/HWGPOSTbeta6.lha`).

1.10 psfonts

PostScript Fonts

I get often asked where one can find PostScript fonts.

To answer all these questions, I made a small summary of archives and locations that contain PostScript fonts (as far as I know):

- The `Post186bin.lha` archive (1.8MB) contains usable PostScript fonts. You will find there the most often used fonts such as Times-Roman, Courier, Helvetica, a.m.m.. This archive can be found on Aminet as `text/print/Post186bin.lha`
- The archive `gs2_3_fonts.lha` that can also be found on Aminet as `text/print/gs2_3_fonts.lha` contains a set of GhostScript fonts (f.e.: `bchb.gsf`). To use these fonts as PostScript fonts you just must rename them to their 'original' names (f.e.: `phvr.gsf` -> Helvetica, these 'original' names can be found in the header of each GhostScript font).
- Various free PostScript fonts can be found in the `text/font/` directory on the Aminet (f.e.: `text/font/PSFonts.lha`).
- Well there exist also various PostScript font collections on cdrom...

1.11 troubleshooting

Troubleshooting

The main things to expect when dealing with printers are troubles and frustrations. If all else fails, just be glad it's not MS-DOS.

from UNIX System Administration Handbook, 2nd edition.

The printable area is too big or too small.

To determine the correct and best possible print area the psprt-handler uses the MaxXDots and MaxYDots from the PrinterExtendedData structure. However, some printer drivers do not fill in the correct values or set these fields at all. To solve this problem the handler determines the page dimension (in pixels) using the Dots/Inch values and the paper sizes in centimeters. This works in the most cases.

But unfortunately the area can still have not the correct dimension.

To solve this the psprt-handler offers various solutions:

- 1) You may use the 'usepreferences' keyword in the configuration file. Using this keyword allows you to set the print area with the preferences printer program. There you can change the left and right margins, the paper length, and print pitch to specify the print area.
The handler uses the following calculations to get the width and height (as suggested by the printer autodocs):

```
WIDTH = (RIGHT_MARGIN - LEFT_MARGIN + 1) / CHARACTERS_PER_INCH  
HEIGHT = LENGTH / LINES_PER_INCH
```

- 2) Use the 'force' option to tell the handler to adjust the DPI values according to the current width and height.
- 3) Another way defining the printable area is by using the 'pixdots' keyword in the configuration file with the number of dots in x and y direction as arguments.
To evaluate the correct number of dots you have to know the page size (inch) and multiply it with the DPI values.

```
WIDTH = (PAGE_WIDTH_INCH - MARGINS) * X_DPI  
HEIGHT = (PAGE_HEIGHT_INCH - MARGINS) * Y_DPI
```

The MARGINS have to be guessed. Good starting values are 1/2 inch for the width and on inch for the height.

PostScript Errors.

I received various mails about PostScript errors. PostScript errors are reported by the interpreter, the post.library in this case. Refer to the post.library documentation or a PostScript book for further details.

However here come the descriptions of some errors you might encounter:

- configurationerror
setpagedevice request cannot be satisfied.

- dictfull
Dictionary is full.
- dictstackoverflow
Too many begins.
- dictstackundeflow
Too many ends.
- invalidaccess
Access attribute violated (e.g. attempted to write a read-only object).
- invalidfont
Bad font name or dictionary.
- ioerror
Some kind of error during input or output.
- limitcheck
Some implementation-dependent size restriction has been exceeded.
- nocurrentpoint
The current point is not defined, yet.
- rangecheck
Operand is too big or too small.
- stackoverflow
The stack was full before the last push.
- stackundeflow
You tried to pop from an empty stack.
- syntaxerror
PostScript's syntax has been violated.
- typecheck
Operand is of the wrong type.
- undefined
Name is not defined in any dictionary on the stack.
- VMerror
Virtual memory full.

1.12 psprttest

psprttest.ps

Included to the psprt-handler archive you find the psprttest.ps PostScript file. This file may help you test the handler or to configurate it correctly.

The PostScript file draws two triangles, one in the top left edge and

the other in the bottom right edge of the page:

```

.....
. | / .
. | / .
.   .
.   .
.   .
.   .
.   .
.   .
.   /|.
.   /_|.
.....

```

I made the image that simple so that you do not have to waste ink, toner, or whatever your printer uses.

1.13 history

History

This history table may be rather incomplete. It contains only bigger changes made to the software. No history entry prior to 37.294.

37.530 (04.June.95)

- Bug fixed, the handler used always 'letter' as paper size.

37.460 (28.June.95)

- Added the 'usepreferences' and 'useprefs' keywords to the configuration file.
- Added the 'pagedots' keyword to the configuration file.
- Added the 'priority' keyword to the configuration file to set the task priority of the psprt-handler while interpreting and printing a page.
- Added the 'memvlen', 'memflen', 'memllen', and 'memhlen' keyword to the configuration file to select post.library memory sizes. Refer to the


```

post.library
documentation for more
details.
```
- The 'fit' option works now also with the Narrow and Wide Tractor paper sizes.
- The handler searches the Init.ps file "quietly" without bringing up a requester.
- Fixed problem with some printer drivers (again).

37.422 (31.Mar.95)

- Progress indicator added, a bar which shows you how much of a file has been processed so far (refer to Configuration for more details about enabling/disabling the progress indicator).

37.320 (23.Mar.95)

- Added config file support (see Configuration for more details).

37.312 (23.Feb.95)

- Matrix dot printers better supported.

37.310 (11.Jan.95)

- Environment variable 'PSPRT_TMPDIR' added.

37.304 (08.Jan.95)

- 'fit' option added (see Configuration).

37.302 (17.Nov.94)

- first release.

37.294 (01.Oct.94)

- (...)

1.14 credits

Thanks go to...

Adrian Aylward - without his post.library this wouldn't be possible.
Stefan Walter - for the SIM debugger environment.

Heinz Wrobel - for HWGPOST that solves some problems with post.library v1.7.

Joseph M. Hinkle - for all the discussions about handlers.

Bryan Ford, Rene Eberhard, Christian Schneider, Tony Leneis, Gary Raposo, Swen Stullich, Macro Krause, Dirk Böhmer, Kai Bolay, and many more for their support and suggestions.

1.15 author

Author

If you have bugreports, questions, ideas, flames or complaints (constructive criticism is always welcome), or if you just want to contact me, write or send a letter to:

Daniel Weber

Internet: dweber@amiga.icu.net.ch

Mail: Daniel Weber
Hoeflistrasse 32
CH-8135 Langnau
Switzerland.
