

Programming Project Interface Docs

COLLABORATORS

	<i>TITLE :</i>		
	Programming Project Interface Docs		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Programming Project Interface Docs	1
1.1	Programming Project Interface Docs	1
1.2	Introduction	1
1.3	Usage	2
1.4	Settings	2
1.5	The '%s' designator	3
1.6	The Kill Extension Checkbox	3
1.7	Defaults File	4
1.8	Output	4
1.9	Legalities	4

Chapter 1

Programming Project Interface Docs

1.1 Programming Project Interface Docs

Welcome to the Programming Project Interface Amigaguide file.

If you have questions or problems not covered in this file, then please email me at:

platt@mhdl.moorhead.msus.edu.

I currently do not give out my real address over the Internet, so if you email me at that address and that account no longer exists (which shouldn't happen until about June of 1995) then mail root at that machine and ask to have your mail forwarded. There is no guarantee that it will get to me, but the chances are a lot better than not bothering at all.

Any comments and suggestions are welcome.

BTW - AmigaDOS 2+ is required.

Introduction
Usage
Legalities

1.2 Introduction

For some time I have been looking for a sort of control panel that I could use for any programming language. SAS has its own editor and it interfaces to the compiler. E has EDS, ESEE, and E-Gui. However, I wanted to use the same interface across the board; for any language and any editor (which ruled out EDS and ESEE). At first E-Gui seemed like a good option, but its author made it shareware. Thus was born PPI (which is freeware).

With PPI, you can do several things. You can start a compiler (any compiler), a text editor, and two other actions. The

actions can be defined as you wish. Personally, I use them to back up my source and use EFormat.

For more usage details, see Usage .

I wrote PPI in E (a VERY good language, thanks Wouter!) and the source is also freeware. However I do retain the rights to the source. Also, no modified version of PPI may be released. The source is only for other E programmers to learn from. Anyone who decides they would like to give E a shot should mail me and I'll help start you out. You hardcore C programmers ought to look at it too. E does things C can't even dream of (e.g. expression quoting). On top of all this E compiles extremely fast (about 2 seconds for this program) and produces extremely (I like that word! :) tight code.

BTW - Yeah I DO like E! Thanks for asking.

Please read Legalities for any questions about redistribution and source code/binary usage.

1.3 Usage

To start PPI you can just click on the icon or invoke it from the cli. Nothing too fancy there! :+)

Once PPI is up and running you will see its interface which consists of several buttons and some string gadgets.

At this point, we'll break this down to some main areas:

(From this point forward, I would have PPI running so this makes sense.)

Settings
Output

1.4 Settings

The '%s' designator
The Kill Extension Checkbox
Defaults File

1.5 The '%s' designator

At the bottom of the window you will see some text like this:

```
Workfile = device:directory/filename
```

This will be empty and only say 'Workfile = ' if this is the first time you have run PPI and have not saved its preferences before.

This workfile can be changed by pressing the Pick button with the mouse or by pressing 'p' while the window is active. This brings up the asl requester.

The workfile is very important to using PPI. In the string gadgets you can specify the workfile name by using '%s' in place of its filename.

Example:

In the Compiler string gadget you could have the following text:

```
sc link %s
```

If your workfile is sys:programming/project/whatever.c the command line that would be executed is:

```
sc link sys:programming/project/whatever.c
```

The '%s' designator can be used in any of the string gadgets. Another example:

In the Action 1 string gadget you could have the following text:

```
copy %s t:backup-of-my-source
```

The '%s' designator always represents the full path and filename of your workfile.

NOTE: You can make an action which looks like this:

```
run >NIL: sys:apps/edit/turbotext %s
```

In other words, you can detach a process completely from PPI (otherwise it waits for you to quit) and you can redirect output from the action. This may seem obvious, but I just want to make sure it's documented.

1.6 The Kill Extension Checkbox

Above the workfile text you will see a checkbox which says 'Kill Extension For Compiling'. This is a switch which will

remove the workfile's filename extension before passing it to the compiler. Some compilers require this (e.g. E and some Modula-2 compilers).

Example:

With '%s' = sys:dots.e and the checkbox on/checked and the compiler string = 'ec %s' hitting the compile button would pass the following command to ADos:

```
ec dots
```

If the checkbox were off/unchecked we would get:

```
ec dots.e
```

1.7 Defaults File

If you press the 'Save Settings' button, PPI will save ALL of your current strings and your checkbox state to s:PPI.prefs. PPI automatically tries to load this file in at startup. This is a plain text file and may be manipulated with a text editor if you wish. Just remember that the last line in the file is for the checkbox and must be either 'T' rue or 'F'alse.

Currently, PPI does not let you load alternative settings files. The basic assumption here is that most people will only use one compiler at a time.

1.8 Output

For the most part, there is none!!! However, you should note that if your compiler (or any other command you invoke from PPI) produces any output, the output will appear in a window which has a close gadget for when you are through with it.

1.9 Legalities

First off, you use this software at your own risk. By running this software, compiling any form of the source code, or reading this text file you agree to the following terms:

- The author of PPI (Vincent Platt) is not responsible for ANY negative effects you or your system suffer.
 - The author of PPI is released from all liabilities resulting from the use of PPI.
-

- The source code may not be modified and any modified version and may not be released into any market (free or commercial).
 - Any form of PPI may not be used for any commercial profit. (Though a commercial developer may use PPI for his own purposes.)
 - The source code is to be used for learning purposes only.

 - PPI may only be distributed in its original form through public domain (read NON-PROFIT) channels which includes the following files:

```
PPI.info
PPI/
  PPI
  PPI.info
  PPI.guide
  PPI.guide.info
  PPI.e
```
 - PPI may be distributed by Fred Fish.
 - PPI may NOT be distributed through Genie, Compuserve, Tymnet, or any other X.25 service. This is because these are commercial (read PROFIT) organizations.
-