FPL

COLLABORATORS

	TITLE : FPL	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FPL

1.1 FPL General Documentation

This documents the FPL language as it runs from version 11. If you happen to use any lower version, upgrade!

1.2 Installation

AMIGA
=====

Copy the file "fpl.library" into your LIBS: directory. If you have a 68020 (or later) processor, copy the file found in the subdirectory "020/" to your LIBS: instead of the other file.

Make an FPL: assign in your user-startup to the root directory of the FPL tree.

If you ever intend to do anyting yourself with the fpl.library, you should consider copying the 'include/' directory tree to INCLUDE:.

```
    OTHER
    =====
  FPL have not been used by me "for real" in other systems than Amiga, which
makes installation procedure and other stuff to remain undocumented. Please
contribute with your experiences!
```

1.3 No warranty

```
  BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE
LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE
STATED IN WRITING THE COPYRIGHT HOLDERS PROVIDE THE LIBRARY "AS IS" WITHOUT
WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS
WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
NECESSARY SERVICING, REPAIR OR CORRECTION.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL
ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE LIBRARY AS
PERMITTED, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL,
INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE
THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE
LIBRARY TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER
PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
```

1.4 Distribution

```
  ##########################################################################
  #                                                                        #
  # This program is free software; you may redistribute for non            #
  # commercial purposes only. Commercial programs must have a written      #
  # permission from the author to use FPL. FPL is *NOT* public domain!     #
  # Any provided source code is only for reference and for assurance       #
  # that users should be able to compile FPL on any operating system       #
  # he/she wants to use it in!                                             #
  #                                                                        #
  # You may not change, resource, patch files or in any way reverse        #
  # engineer anything in the FPL package.                                  #
  #                                                                        #
  # This program is distributed in the hope that it will be useful,        #
  # but WITHOUT ANY WARRANTY; without even the implied warranty of         #
  # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.                   #
  #                                                                        #
  # Daniel Stenberg                                                        #
  # Ankdammsgatan 36, 4tr                                                  #
  # S-171 43 Solna                                                         #
  # Sweden                                                                 #
  #                                                                        #
  # FidoNet 2:201/328    email:dast@sth.frontec.se                         #
  #                                                                        #
  ##########################################################################
```

1.5 Files in the package

```
Files included in the FPL package:

[root] (dir):   Main files of the FPL package.

  demo.FPL              FPL program featuring almost everything the language
                        includes. Run this by entering "src/SFPL demo.FPL".
  HISTORY               Changes to FPL during history. Important information
                        if you are updating from an older version.
  FUTURE                Changes that will be done to FPL in the near and far
                        future.
  README                Short text about what FPL is.

docs (dir):

  FPLdoc.guide          General FPL distribution details. Bug reports,
                        distribution rights and more.

  FPLuser.guide         Programming language documentation. Including
                        programming  concepts  and  examples .

  FPLlib.guide          Documentation covering fpl.library. How to implement
                        it in your own code and all that has to be done to
                        create a comforable FPL environment.

  fpl.doc               autodoc formatted fpl.library function reference
file.

  FPL.README            Short text to include in software packages where FPL
                        is used!

FPL (dir):      FPL program examples. Check out these programs to learn about
                FPL programming.

  char2ASCII.FPL        Asks for a character and outputs it's ASCII code.
  ASCII2char.FPL        Ascs for an ASCII code and output the character.

util (dir):     fpl.library implementation examples.

  FrexxCalc.c           Source code to FrexxCalc. Execute to compile.
                        FrexxCalc - the utility - is an advanced command line
                        calculator which is especially made for people who
                        wants a calculator able to evaluate the result of
                        C  expressions .

  FrexxCalc.doc         Short documentation file to the uper util

  LibList               New Amiga library management utility.

  LibList.doc           Short documentation file to the uper util

src (dir):      FPL source code

  COMPILING             Read this to get information about how to compile.
  Makefile              (UNIX) Uncomment the lines affecting the machine
```

```
                        you're compiling on!
  smakefile             (Amiga) SAS/C version of the makefile.
  [the rest]            FPL shared library Source code.

--- Amiga only: ---

  fpl.library           FPL library! Must be placed in your LIBS: directory.

debug (dir):    FPL debugger source directory

funclib (dir):  Funclib development directory

include (dir):  Include file tree to fpl.library. Copy them all to INCLUDE:
```

1.6 Requirements

FPL is a platform independent script language. There will be different requirements in different environments.

AMIGA:
fpl.library should work on all Amigas using AmigaDos version 1.2 (dos.library V33) or higher.

1.7 Bug reports

Sometimes you will encounter a bug in FPL. Although I cannot promise I can or will fix the bug (I might not even agree it's a bug!), I want to hear about any bugs you encounter, just in case I do want to fix them.

To make it possible for me to fix a bug, you have to report it. In order to do so you must know how to recognize one and how to report it .

1.8 What is a bug?

FPL is executed as a part of a software not necessary coded by FrexxWare . Errors that occur are _very_ hard for the user to trace back to the library and you should as a FPL programmer complain to the software programmer who, if it really turns out to be an FPL problem, reports it to me.

If FPL executes an illegal instruction, or dies with an operating system error message that indicates a problem in the program (as opposed to something like "disk full"), it's most certainly a bug.

Taking forever to complete a command can be a bug, but you must make certain that it really was FPL's fault. Some commands simply take a long time. If the input was such that you KNOW it should have been processed quickly, report it as a bug. If you don't know whether the command should take a long time, find out by looking in the manual or by asking for assistance.

If a command you are familiar with causes an FPL error message in a case

where its usual definition ought to be reasonable, it is probably a bug.

If a command does the wrong thing, that is a bug. But be sure you know for certain what it ought to have done. If you aren't familiar with the command, or don't know for certain how the command is supposed to work, then it might actually be working right. Rather than jumping to conclusions, show the problem to someone who knows for certain.

Finally, a command's intended definition may not be best for coding with. This is a very important sort of problem, but it is also a matter of judgment. Also, it is easy to come to such a conclusion out of ignorance of some of the existing features. It is probably best not to complain about such a problem until you have checked the documentation in the usual ways, feel confident that you understand it, and know for certain that what you want is not available. If you are not sure what the command is supposed to do after a careful reading of the manual, check the index and glossary for any terms that may be unclear. If you still do not understand, this indicates a bug in the manual. The manual's job is to make everything clear. It is just as important to report documentation bugs as to report program bugs.

Since the AmigaDos and some other OSes lacks the beauty called memory protection, the memory allocated by FPL might get changed by another process than FPL itself. Therefore you must be sure of that there is none of the other programs running together with FPL that might be the cause of the bug.

FPL puts a lot of responsibility on the coder using the library. Returning wrong values, freeing wrong variables, missing assigns etc, etc may just be the crashing/failing/bugging reason. Before complaining to me, first complain to the coder of the software you're controlling with FPL, and make absolutely sure he has followed the coding guidelines.

The very useful Amiga debugging tools Mungwall and Enforcer are excellent tools to track illegal memory accesses and other violations. Do use the versions that can work together with the 'SegTracker' to be able to more clearly distiguish whether it's fpl.library that bugs or the host software.

1.9 How to report a bug

When you decide that there is a bug, it is important to report it and to report it in a useful way. A useful bug report is an exact description or why not a copy of the FPL program executed and a detailed description of the effects of the suspected bug.

The most important principle in reporting a bug is to report FACTS, not hypotheses or categorizations. It is always easier to report the facts, but people seem to prefer to strain to posit explanations and report them instead. If the explanations are based on guesses about how FPL is implemented, they will be useless; I will have to try to figure out what the facts must have been to lead to such speculations. Sometimes this is impossible. In any case, that is unnecessary and time-consuming work for us.

Amigas and AmigaDos exist in extremely many different configurations. When reporting a suspicious behavior, include information such as Amiga model, OS version, amount of memory (both chip and fast) and all other useful data such as processor, MMU and co-processors (I advise you to simply include the output

made with Sysinfo). Also make sure you always include the version number of the library you are using (on Amiga: the cli command "version FULL fpl.library" will give answer to the uncertain) and which program you used FPL in when you discovered the bug.

Report the bug through our BBS, paper mail, email or voice phone. See how to reach us for information on how.

Once again, I do not promise to fix the bug; if the bug is serious, or ugly, or easy to fix, I probably will.

1.10 About the project

What is FPL?
Who made FPL?
Why make FPL?
Why use FPL?

FPL has been developed almost entirely under UNIX using memory protection.

The Amiga version is edited, tested and run error free with Mungwall and Enforcer as background processes.

SAS/C Development System's symbolic debugger 'cpr' is *NOT* fun to use when debugging FPL. The amount of linked lists is large and 'cpr' handles them badly. (In my opinion, the debugger is the weakest part of that, in all other ways superb, package.) Although the ability to debug shared libraries is extremely valuable!

FPL is currently being used in the major FrexxWare project called FrexxEd. A highly customizable, programmable, zero limitation text editor (running only at Amiga beta testers at the moment). If you decide to use FPL in your application, please let me know!

I started coding on this project in the beginning of the summer 1992 and since the 2nd of November 1992, it has been released, used and improved.

Any suggestions regarding FPL or the library implementation part, is warmly welcomed. I aim on making FPL superb!

1.11 What is FPL?

FPL (pronounced 'eff-pee-ell') is an advanced flexible real-time ASCII text interpreting multi-file shared library programming language very much inspired by the C programming language.

We say that FPL is "advanced" because there are several ways to solve each programming task, because there are many keywords, functions and expression operators and they enable advanced coding.

We call FPL "flexible" since there is much left to the coder of the software FPL is used in, to create functions and variables that will exist in the FPL

programs for that software.

"real-time" means that FPL does all interpreting and execution at invoke time, with nothing compiled, tokenized or preprocessed. (Even though such possibilities will be implemented in the future versions.)

"ASCII text" means that FPL executes simple texts created in any text editor (I recommend FrexxEd, which is Freeware, highly programmable through FPL and Copyright © by FrexxWare . Yet only available for amiga beta testers...).

"Interpreting" means that the instructions are read one by one and are executed as soon as possible.

"multi-file" means that functions and variables can be accessed and controlled in any number of files at the same time. Using one or several files are the same to FPL internals.

"shared library" means that the entire language is constructed as a selfgovering part of the programs that are using it. When several programs are using FPL at the same time, only one code is loaded into memory. Upgrading the language is not harder than copying new versions of the language to your LIBS: direcrory! (on Amiga).

The fact that FPL is inspired by C is very much noticeable. The code looks and feels almost exactly as C code does.

FPL is extremely powerful and practical when you want your software to include an interpreting language with possibilities for the user himself to create his own functions, text formatters, word interpreters, advanced macros, entire programs, etc, etc... The only restriction is the limit of your mind.

FPL is Freeware, not public domain. Coded for free, given away for free, to be used for free.

(The word 'Frexx' comes from an internal joke among our friends as a word for cool/good/groovy/nice or something like that!)

1.12 Who made FPL?

FPL is programmed mainly by me, Daniel Stenberg, but I got plenty additional help and ideas from Kjell Ericson (who coded the stack allocating and expanding routines), Linus Nielsen (who helped me develop my thoughts around the memory allocation/dealloction re-cycling/caching scheme and compiled it under HPUX) and Björn Stenberg (who gave me ideas and made the first ports to SunOS and Dell Unix!).

We produce our software together under the FrexxWare label.

FrexxWare has released a bunch of small and hopefully useful freely distributed utilities, mainly for use in a development environment such as our own! Get in touch for more information !

Lots of ideas also come from Jonas Engstrom (who's language nowadays seems very poor compared to FPL... :-), Tord Persson (who really have inspired me to continue towards a full C interpreter), Niclas Emdelius (who due to his

troubles implementing FPL in his program has thought of some great ideas to make reality), Stefan Boberg and Lasse Mickos (who in combination with Björn finally made me insert single function invoke support).

I've spent several hundreds of hours on this and yet I've done most of the coding under UNIX (or more correctly AIX on a beautiful RS/6000) enjoying real compiling speed instead of my slow machine at home... (Which until the end of September 1992 was a A500 but since then is a A3000.)

1.13 Why make FPL?

We needed an interpreting script language for our text editor, and I made it entirely for that purpose. Later on we figured out about a dozen of other products that really could have use for a language like this (not the least a few interesting projects we have been thinking of), and we noticed that creating a library of the code wouldn't be that hard.

Software developers in need for a script language will no longer have to code their own. FPL is the solution. FPL is for everyone!

1.14 Why use FPL?

FPL has everything a programming language should have.

FPL programs are possible to write very short and compact.

FPL interprets common ASCII-files, needing no compiling at all. (Although compiling will become a future feature!)

FPL is very similar to C. If you know C, you know FPL.

FPL is very easy to include in your own code and an excellent tool to batch/macro control your software.

FPL is a language on shared library which makes updating the language extremely fast, easy and flexible. (At the moment Amiga, OS/2 and SVR4 UNIX only)

FPL is free.

FPL is a fully resource tracking system. All memory used by FPL is removed whenever any FPL program exits.

FPL is a platform independent language available in several operating systems. And if it isn't already available, porting it to yet another is very simple! When porting your application(s) to other OS'es, the FPL programs will work *exactly* the same!!!

FPL source code is available. You are guaranteed to always have nothing less than the latest FPL version to start from in case of the software support is dropped.