Amiga PERL Port:
Installation and User Guide

PERL VERSION – 4.0.1.7 PATCHLEVEL 35
AMIGA PERL PORT VERSION – 0.1.0

Maintainer: Kent Dalton – Kent.Dalton@FtCollinsCO.NCR.com

Contents

1 Introduction 1
1.1 System Requirements 2
2 Installation 2
2.1 Caveats and Suggestions 3
2.1.1 Caveats 3
2.1.2 Suggestions 3
3 Limitations 3
4 Known Bugs 4
4.1 Bugs Not Expected To Be Fixable 4
4.2 Bugs Expected To Be Fixed in Future Releases 4
5 Reporting (and Fixing) Bugs 4
6 Hacking Perl 5
7 References 5

1. Introduction

Welcome to the Amiga port of Perl 4.035. For those of you who may not be familiar with perl, it is a programming language that has become very popular among user of the Unix operating system. The language is very useful for the manipulation of text, files, and processes and the syntax is very C-like. Perl is interpreted, which can be very convenient when developing small utilities and one-shot scripts.

I'm calling the current version of Amiga Perl 'Beta' since it still has a couple of known, minor, bugs and fails a few of the regression tests. It is very usable,

however, and I have not had any major problems (i.e. no gurus!) with it now that I have gotten it stabilized. Obviously, the program has no warranty, since it is free software. For more warranty and redistribution information, see the 'README', 'Copying', and 'Artistic' files that should have been included with the program.

In the spirit of perl, I freely encourage users to modify the source code and fix bugs to their heart's content. I would like to remain the maintainer of Amiga perl, so if you have any bug-fixes or useful patches, let me know! (See the section on 'Reporting Bugs' below for more information.)

1.1. System Requirements

- Amiga OS 2.04 or higher is **absolutely required**! Do not send e-mail asking about 1.3 and below support. Any such requests will be silently ignored.
- An absolute minimum of 1Mb of memory is required to use perl. At least 3Mb is recommended for executing non-trivial scripts along with background processes and spawned processes. In general, the more RAM you have, the better.
- A hard disk is strongly recommended. Perl will probably be pretty worthless without one.
- A complete GCC 2.2.2 installation is recommended. It comes with a lot of the support stuff needed by perl (making perl easier to install) (ixpipe, ixemul, etc) and you can use it to hack the source (see 'Hacking' section below)!

2. Installation

1. Install libs/ixemul.library in LIBS:
2. Install ixpipe handler, the ixemul Unix-like pipe handler. (See the documentation in the ixpipe directory for more information)
3. Make the following assigns if they do not already exist:
    1. USR: – similar to UNIX /usr directory.
    2. BIN: – similar to UNIX /bin directory.
    3. DEV: – similar to UNIX /dev directory.
    4. TMP: – similar to UNIX /tmp directory.

4. Copy or link the perl binary to a directory somewhere in your path.

5. Make a directory under usr:, usr:lib/perl

6. Copy the perl lib/ files to usr:lib/perl/. Perl will be able to automatically locate them there. Otherwise you'll have to set you environment variables up to find the lib files (see man page).

2.1. Caveats and Suggestions

2.1.1. Caveats

The version of ixemul.library that comes with perl expects real Unix–style pathnames. Devices are denoted by a leading '/', previous directories are denoted by '..'. For example:

```
"/c/delete" in perl is the same as "C:delete" in AmigaDOS.
"../../doc" in perl is the same as "//doc" in AmigaDOS.
```

If you really hate this behavior (IMHO, it's a feature), you can reconfigure ixemul.library with the ixconfig stuff from the GCC distribution.

Another caveat is that many Unix perl scripts will expect certain standard Unix tools to exist. I suggest hunting down any Unix work-alike tools you can find and putting them in BIN:. If anyone comes up with a good set of freely distributable ones, let me know and maybe we can include them in future binary distributions.

2.1.2. Suggestions

I suggest adding the BIN: assign to your search path (preferably in your "user-startup" file. I also suggest making the BIN: to be a directory called "bin" in the USR: assign. This way, ixemul.library will translate references to both: "/usr/bin" and "/bin" to the BIN: assign.

3. Limitations

Most of the limitations of this release are due to the fact that AmigaDOS is not Unix, which means that a number of the perl features do not make sense on a single-user, non-Unix, OS.

- The fork function is not supported.

- The functions related to passwords, user, and group id's are not supported.
- The chroot function is not supported.
- The dbm stuff is not supported.
- Some scripts which expect "/dev/null" to exist will have to be modified since it does not exist on the amiga.

4. Known Bugs

4.1. Bugs Not Expected To Be Fixable

- The "-e" option typically does not work as expected due to the way most Amiga shells handle quoting. I know csh 5.19, for one, fails very badly at this. This bug is basically beyond my control.
- Scripts which expect the "#!" shell execution mechanism to work will fail. I've included a program called "cscript" to help alleviate this problem (more info in the cscript directory of this distribution.)

4.2. Bugs Expected To Be Fixed in Future Releases

- Fails a number of tests (majority located in "op/*.t") due to missing Unix commands (cat, touch, etc).
- Fails a few of the tests outright.

5. Reporting (and Fixing) Bugs

Bugs that are general perl bugs go to Larry Wall (lwall@netlabs.com). Bugs specific to the Amiga should be reported directly to me (Kent.Dalton@FtCollinsCO.NCR.com). Don't expect an incredibly rapid response unless your problem is extremely serious... (You do have the source after all!)

Speaking of the source, I fully welcome others to fix bugs or just plain hack the code and encourage those who do to send me fixes and patches via e-mail so I can include them in the main release. *Please send patches as context diffs* (if you don't have a program that can generate context diffs or don't know what context diffs are, let me knoe via e-mail). I will try to include a complete contributors list in future versions to give credit where it is due.

6. Hacking Perl

Rebuilding this perl port requires GCC 2.2.2 with ixemul.library (Removing the dependence on GCC/ixemul would be pretty non-trivial.), God only knows how much RAM (probably a bare minimum of 4Mb without loading WB), a Hard Disk, and an accelerated CPU. Well, actually the accelerater is IMHO, but if you actually do try to rebuild perl on a 7MHz 68000 using GCC, I bet you'll end up agreeing with me! The Makefile was created for use with Ben Eng's BMake, any other reasonably full-featured make should be able to digest it, though.

As configured with the Makefile, you should be able to build perl using gcc with no compilation errors (or even warnings). I've tried to preface most of the amiga-specific changes with #define AMIGA's. There really aren't tons of code changes. (The really hard part was getting it configured, etc and then isolating the little changes that were necessary (in such a big program)).

7. References

There are basically two primary perl references for those of you who wish to learn perl. First, is the mega-man page that is included with this distribution. And second, is the book "Programming Perl" published by O'Reilly and Associates, ISBN: 0-937175-64-1. Both are highly recommended reading. The Reference Card that comes with the Perl book is also available in postscript and LaTeXformats via anonymous FTP from any site that archives the newsgroup comp.sources.misc