sh: ar: not found

This is a message from your shell telling you that the command 'ar' was not found. You need to check your PATH environment variable to make sure that it includes the directory with the 'ar' command. This is a common problem on Solaris, where 'ar' is in the /usr/ccs/bin/ directory.

122.6 MAKE TEST

122.6.1 op/stat.t test 4 in Solaris

op/stat.t test 4 may fail if you are on a tmpfs of some sort. Building in /tmp sometimes shows this behavior. The test suite detects if you are building in /tmp, but it may not be able to catch all tmpfs situations.

122.6.2 nss_delete core dump from op/pwent or op/grent

See nss_delete core dump from op/pwent or op/grent in *perlhpux*.

122.7 PREBUILT BINARIES OF PERL FOR SOLARIS.

You can pick up prebuilt binaries for Solaris from http://www.sunfreeware.com/, http://www.blastwave.org, ActiveState http://www.activestate.com/, and http://www.perl.com/ under the Binaries list at the top of the page. There are probably other sources as well. Please note that these sites are under the control of their respective owners, not the perl developers.

122.8 RUNTIME ISSUES FOR PERL ON SOLARIS.

122.8.1 Limits on Numbers of Open Files on Solaris.

The stdio(3C) manpage notes that for LP32 applications, only 255 files may be opened using fopen(), and only file descriptors 0 through 255 can be used in a stream. Since perl calls open() and then fdopen(3C) with the resulting file descriptor, perl is limited to 255 simultaneous open files, even if sysopen() is used. If this proves to be an insurmountable problem, you can compile perl as a LP64 application, see Building an LP64 perl for details. Note also that the default resource limit for open file descriptors on Solaris is 255, so you will have to modify your ulimit or rctl (Solaris 9 onwards) appropriately.

122.9 SOLARIS-SPECIFIC MODULES.

See the modules under the Solaris:: and Sun::Solaris namespaces on CPAN, see http://www.cpan.org/modules/by-module/Solaris/ and http://www.cpan.org/modules/by-module/Sun/.

122.10 SOLARIS-SPECIFIC PROBLEMS WITH MODULES.

122.10.1 Proc::ProcessTable on Solaris

Proc::ProcessTable does not compile on Solaris with perl5.6.0 and higher if you have LARGEFILES defined. Since largefile support is the default in 5.6.0 and later, you have to take special steps to use this module.

The problem is that various structures visible via procfs use off_t, and if you compile with largefile support these change from 32 bits to 64 bits. Thus what you get back from procfs doesn't match up with the structures in perl, resulting in garbage. See proc(4) for further discussion.

A fix for Proc::ProcessTable is to edit Makefile to explicitly remove the largefile flags from the ones MakeMaker picks up from Config.pm. This will result in Proc::ProcessTable being built under the correct environment. Everything should then be OK as long as Proc::ProcessTable doesn't try to share off_t's with the rest of perl, or if it does they should be explicitly specified as off64_t.