# NAME

README.aix - Perl version 5 on IBM Unix (AIX) systems

# DESCRIPTION

This document describes various features of IBM's Unix operating system (AIX) that will affect how Perl version 5 (hereafter just Perl) is compiled and/or runs.

## Compiling Perl 5 on AIX

For information on compilers on older versions of AIX, see *Compiling Perl 5 on older AIX versions up to 4.3.3.*

When compiling Perl, you must use an ANSI C compiler. AIX does not ship an ANSI compliant C-compiler with AIX by default, but binary builds of gcc for AIX are widely available.

## Supported Compilers

Currently all versions of IBM's "xlc", "xlc_r", "cc", "cc_r" or "vac" ANSI/C compiler will work for building Perl if that compiler works on your system.

If you plan to link Perl to any module that requires thread-support, like DBD::Oracle, it is better to use the _r version of the compiler. This will not build a threaded Perl, but a thread-enabled Perl. See also *Threaded Perl* later on.

As of writing (2009-08) only the IBM XL C for AIX or XL C/C++ for AIX compiler is supported by IBM on AIX 5L/6.1.

The following compiler versions are supported by IBM:

XL C and XL C/C++ V7, V8, V9, V10

The XL C for AIX is integrated in the XL C/C++ for AIX compiler.

If you choose XL C/C++ V9 you need APAR IZ35785 installed otherwise the integrated SDBM_File do not compile correctly due to an optimization bug. You can circumvent this problem by adding -qipa to the optimization flags (-Doptimize='-O -qipa'). The PTF for APAR IZ35785 which solves this problem is available from IBM (April 2009 PTF for XL C/C++ Enterprise Edition for AIX, V9.0).

Perl can be compiled with either IBM's ANSI C compiler or with gcc. The former is recommended, as not only it can compile Perl with no difficulty, but also can take advantage of features listed later that require the use of IBM compiler-specific command-line flags.

If you decide to use gcc, make sure your installation is recent and complete, and be sure to read the Perl INSTALL file for more gcc-specific details. Please report any hoops you had to jump through to the development team.

## Incompatibility with AIX Toolbox lib gdbm

If the AIX Toolbox version of lib gdbm 1.8.x is installed on the system then Perl will not work. This library contains a defect version of the dbm_store() function. The lib gdbm will be automatically removed from the wanted libraries.

## Perl 5.10 was successfully compiled and tested on:

```
AIX Level                 | Compiler Level          | w th | w/o th
--------------------------+-------------------------+------+-------
5.1 TL9 32 bit            | XL C/C++ V7             | OK   | OK
5.1 TL9 32 bit            | gcc 3.2.2               | OK   | OK
5.1 TL9 64 bit            | XL C/C++ V7             | OK   | OK
5.2 TL10 32 bit           | XL C/C++ V8             | OK   | OK
5.2 TL8 64 bit            | VA C/C++ V6             | OK   | OK
5.2 TL10 64 bit           | XL C/C++ V8             | OK   | OK
5.3 TL7 32 bit            | XL C/C++ V9 + IZ35785   | OK   | OK
```

```
5.3 TL7 32 bit         | gcc 4.2.4              | OK   | OK
5.3 TL7 64 bit         | XL C/C++ V9 + IZ35785  | OK   | OK
6.1 TL1 32 bit         | XL C/C++ V10           | OK   | OK
6.1 TL1 64 bit + IZ39077 | XL C/C++ V10         | OK   | OK


w th   = with thread
w/o th = without thread
OK     = tested
```

Successfully tested means that all "make test" runs finish with an result of 100% OK. All tests were conducted with -Duseshrplib set.

## Building Dynamic Extensions on AIX

Starting from Perl 5.7.2 (and consequently 5.8.x / 5.10.x) and AIX 4.3 or newer Perl uses the AIX native dynamic loading interface in the so called runtime linking mode instead of the emulated interface that was used in Perl releases 5.6.1 and earlier or, for AIX releases 4.2 and earlier. This change does break backward compatibility with compiled modules from earlier Perl releases. The change was made to make Perl more compliant with other applications like Apache/mod_perl which are using the AIX native interface. This change also enables the use of C++ code with static constructors and destructors in Perl extensions, which was not possible using the emulated interface.

It is highly recommended to use the new interface.

## Using Large Files with Perl

Should yield no problems.

## Threaded Perl

Should yield no problems with AIX 5.1 / 5.2 / 5.3 and 6.1.

IBM uses the AIX system Perl (V5.6.0 on AIX 5.1 and V5.8.2 on AIX 5.2 / 5.3 and 6.1) for some AIX system scripts. If you switch the links in /usr/bin from the AIX system Perl (/usr/opt/perl5) to the newly build Perl then you get the same features as with the IBM AIX system Perl if the threaded options are used.

The threaded Perl build works also on AIX 5.1 but the IBM Perl build (Perl v5.6.0) is not threaded on AIX 5.1.

## 64-bit Perl

If your AIX system is installed with 64-bit support, you can expect 64-bit configurations to work. If you want to use 64-bit Perl on AIX 6.1 you need a APAR for a libc.a bug which affects (n)dbm_XXX functions. The APAR number for this problem is IZ39077.

If you need more memory (larger data segment) for your Perl programs you can set:

```
/etc/security/limits
default:                    (or your user)
    data = -1               (default is 262144 * 512 byte)
```

With the default setting the size is limited to 128MB. The -1 removes this limit. If the "make test" fails please change your /etc/security/limits as stated above.

## Recommended Options AIX 5.1/5.2/5.3 and 6.1 (threaded/32-bit)

With the following options you get a threaded Perl version which passes all make tests in threaded 32-bit mode, which is the default configuration for the Perl builds that AIX ships with.

```
rm config.sh
./Configure \
```

```
-d \
-Dcc=cc_r \
-Duseshrplib \
-Dusethreads \
-Dprefix=/usr/opt/perl5_32
```

The -Dprefix option will install Perl in a directory parallel to the IBM AIX system Perl installation.

## Recommended Options AIX 5.1/5.2/5.3 and 6.1 (32-bit)

With the following options you get a Perl version which passes all make tests in 32-bit mode.

```
rm config.sh
./Configure \
-d \
-Dcc=cc_r \
-Duseshrplib \
-Dprefix=/usr/opt/perl5_32
```

The -Dprefix option will install Perl in a directory parallel to the IBM AIX system Perl installation.

## Recommended Options AIX 5.1/5.2/5.3 and 6.1 (threaded/64-bit)

With the following options you get a threaded Perl version which passes all make tests in 64-bit mode.

```
export OBJECT_MODE=64 / setenv OBJECT_MODE 64 (depending on your shell)


rm config.sh
./Configure \
-d \
-Dcc=cc_r \
-Duseshrplib \
-Dusethreads \
-Duse64bitall \
-Dprefix=/usr/opt/perl5_64
```

## Recommended Options AIX 5.1/5.2/5.3 and 6.1(64-bit)

With the following options you get a Perl version which passes all make tests in 64-bit mode.

```
export OBJECT_MODE=64 / setenv OBJECT_MODE 64 (depending on your shell)


rm config.sh
./Configure \
-d \
-Dcc=cc_r \
-Duseshrplib \
-Duse64bitall \
-Dprefix=/usr/opt/perl5_64
```

The -Dprefix option will install Perl in a directory parallel to the IBM AIX system Perl installation.

If you choose gcc to compile 64-bit Perl then you need to add the following option:

```
-Dcc='gcc -maix64'
```

## Compiling Perl 5 on older AIX versions up to 4.3.3

Due to the fact that AIX 4.3.3 reached end-of-service in December 31, 2003 this information is provided as is. The Perl versions prior to Perl 5.8.9 could be compiled on AIX up to 4.3.3 with the following settings (your mileage may vary):

When compiling Perl, you must use an ANSI C compiler. AIX does not ship an ANSI compliant C-compiler with AIX by default, but binary builds of gcc for AIX are widely available.

At the moment of writing, AIX supports two different native C compilers, for which you have to pay: **xlC** and **vac**. If you decide to use either of these two (which is quite a lot easier than using gcc), be sure to upgrade to the latest available patch level. Currently:

```
xlC.C      3.1.4.10 or 3.6.6.0 or 4.0.2.2 or 5.0.2.9 or 6.0.0.3
vac.C      4.4.0.3  or 5.0.2.6 or 6.0.0.1
```

note that xlC has the OS version in the name as of version 4.0.2.0, so you will find xlC.C for AIX-5.0 as package

```
xlC.aix50.rte   5.0.2.0 or 6.0.0.3
```

subversions are not the same "latest" on all OS versions. For example, the latest xlC-5 on aix41 is 5.0.2.9, while on aix43, it is 5.0.2.7.

Perl can be compiled with either IBM's ANSI C compiler or with gcc. The former is recommended, as not only can it compile Perl with no difficulty, but also can take advantage of features listed later that require the use of IBM compiler-specific command-line flags.

The IBM's compiler patch levels 5.0.0.0 and 5.0.1.0 have compiler optimization bugs that affect compiling perl.c and regcomp.c, respectively. If Perl's configuration detects those compiler patch levels, optimization is turned off for the said source code files. Upgrading to at least 5.0.2.0 is recommended.

If you decide to use gcc, make sure your installation is recent and complete, and be sure to read the Perl INSTALL file for more gcc-specific details. Please report any hoops you had to jump through to the development team.

### OS level

Before installing the patches to the IBM C-compiler you need to know the level of patching for the Operating System. IBM's command 'oslevel' will show the base, but is not always complete (in this example oslevel shows 4.3.NULL, whereas the system might run most of 4.3.THREE):

```
# oslevel
4.3.0.0
# lslpp -l | grep 'bos.rte '
bos.rte            4.3.3.75  COMMITTED  Base Operating System Runtime
bos.rte             4.3.2.0  COMMITTED  Base Operating System Runtime
#
```

The same might happen to AIX 5.1 or other OS levels. As a side note, Perl cannot be built without bos.adt.syscalls and bos.adt.libm installed

```
# lslpp -l | egrep "syscalls|libm"
bos.adt.libm      5.1.0.25  COMMITTED  Base Application Development
bos.adt.syscalls  5.1.0.36  COMMITTED  System Calls Application
#
```

## Building Dynamic Extensions on AIX

AIX supports dynamically loadable objects as well as shared libraries. Shared libraries by convention end with the suffix .a, which is a bit misleading, as an archive can contain static as well as dynamic members. For Perl dynamically loaded objects we use the .so suffix also used on many other platforms.

Note that starting from Perl 5.7.2 (and consequently 5.8.0) and AIX 4.3 or newer Perl uses the AIX native dynamic loading interface in the so called runtime linking mode instead of the emulated interface that was used in Perl releases 5.6.1 and earlier or, for AIX releases 4.2 and earlier. This change does break backward compatibility with compiled modules from earlier Perl releases. The change was made to make Perl more compliant with other applications like Apache/mod_perl which are using the AIX native interface. This change also enables the use of C++ code with static constructors and destructors in Perl extensions, which was not possible using the emulated interface.

## The IBM ANSI C Compiler

All defaults for Configure can be used.

If you've chosen to use vac 4, be sure to run 4.4.0.3. Older versions will turn up nasty later on. For vac 5 be sure to run at least 5.0.1.0, but vac 5.0.2.6 or up is highly recommended. Note that since IBM has removed vac 5.0.2.1 through 5.0.2.5 from the software depot, these versions should be considered obsolete.

Here's a brief lead of how to upgrade the compiler to the latest level. Of course this is subject to changes. You can only upgrade versions from ftp-available updates if the first three digit groups are the same (in where you can skip intermediate unlike the patches in the developer snapshots of Perl), or to one version up where the "base" is available. In other words, the AIX compiler patches are cumulative.

```
vac.C.4.4.0.1 => vac.C.4.4.0.3  is OK     (vac.C.4.4.0.2 not needed)
xlC.C.3.1.3.3 => xlC.C.3.1.4.10 is NOT OK (xlC.C.3.1.4.0 is not available)


# ftp ftp.software.ibm.com
Connected to service.boulder.ibm.com.
: welcome message ...
Name (ftp.software.ibm.com:merijn): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
... accepted login stuff
ftp> cd /aix/fixes/v4/
ftp> dir other other.ll
output to local-file: other.ll? y
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.
ftp> dir xlc xlc.ll
output to local-file: xlc.ll? y
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.
ftp> bye
... goodbye messages
# ls -l *.ll
-rw-rw-rw-   1 merijn   system    1169432 Nov  2 17:29 other.ll
-rw-rw-rw-   1 merijn   system      29170 Nov  2 17:29 xlc.ll
```

On AIX 4.2 using xlC, we continue:

---

```
# lslpp -l | fgrep 'xlC.C '
  xlC.C                       3.1.4.9  COMMITTED  C for AIX Compiler
  xlC.C                       3.1.4.0  COMMITTED  C for AIX Compiler
# grep 'xlC.C.3.1.4.*.bff' xlc.ll
-rw-r--r--  1 45776101 1            6286336 Jul 22 1996  xlC.C.3.1.4.1.bff
-rw-rw-r--  1 45776101 1            6173696 Aug 24 1998  xlC.C.3.1.4.10.bff
-rw-r--r--  1 45776101 1            6319104 Aug 14 1996  xlC.C.3.1.4.2.bff
-rw-r--r--  1 45776101 1            6316032 Oct 21 1996  xlC.C.3.1.4.3.bff
-rw-r--r--  1 45776101 1            6315008 Dec 20 1996  xlC.C.3.1.4.4.bff
-rw-rw-r--  1 45776101 1            6178816 Mar 28 1997  xlC.C.3.1.4.5.bff
-rw-rw-r--  1 45776101 1            6188032 May 22 1997  xlC.C.3.1.4.6.bff
-rw-rw-r--  1 45776101 1            6191104 Sep  5 1997  xlC.C.3.1.4.7.bff
-rw-rw-r--  1 45776101 1            6185984 Jan 13 1998  xlC.C.3.1.4.8.bff
-rw-rw-r--  1 45776101 1            6169600 May 27 1998  xlC.C.3.1.4.9.bff
# wget ftp://ftp.software.ibm.com/aix/fixes/v4/xlc/xlC.C.3.1.4.10.bff
#
```

On AIX 4.3 using vac, we continue:

```
# lslpp -l | grep 'vac.C '
 vac.C                        5.0.2.2  COMMITTED  C for AIX Compiler
 vac.C                        5.0.2.0  COMMITTED  C for AIX Compiler
# grep 'vac.C.5.0.2.*.bff' other.ll
-rw-rw-r--  1 45776101 1           13592576 Apr 16 2001  vac.C.5.0.2.0.bff
-rw-rw-r--  1 45776101 1           14133248 Apr  9 2002  vac.C.5.0.2.3.bff
-rw-rw-r--  1 45776101 1           14173184 May 20 2002  vac.C.5.0.2.4.bff
-rw-rw-r--  1 45776101 1           14192640 Nov 22 2002  vac.C.5.0.2.6.bff
# wget ftp://ftp.software.ibm.com/aix/fixes/v4/other/vac.C.5.0.2.6.bff
#
```

Likewise on all other OS levels. Then execute the following command, and fill in its choices

```
# smit install_update
 -> Install and Update from LATEST Available Software
 * INPUT device / directory for software [ vac.C.5.0.2.6.bff    ]
 [ OK ]
 [ OK ]
```

Follow the messages ... and you're done.

If you like a more web-like approach, a good start point can be
http://www14.software.ibm.com/webapp/download/downloadaz.jsp and click "C for AIX", and follow
the instructions.

## The usenm option

If linking miniperl

```
cc -o miniperl ... miniperlmain.o opmini.o perl.o ... -lm -lc ...
```

causes error like this

```
ld: 0711-317 ERROR: Undefined symbol: .aintl
ld: 0711-317 ERROR: Undefined symbol: .copysignl
ld: 0711-317 ERROR: Undefined symbol: .syscall
ld: 0711-317 ERROR: Undefined symbol: .eaccess
ld: 0711-317 ERROR: Undefined symbol: .setresuid
```

```
ld: 0711-317 ERROR: Undefined symbol: .setresgid
ld: 0711-317 ERROR: Undefined symbol: .setproctitle
ld: 0711-345 Use the -bloadmap or -bnoquiet option to obtain more
information.
```

you could retry with

```
make realclean
rm config.sh
./Configure -Dusenm ...
```

which makes Configure to use the `nm` tool when scanning for library symbols, which usually is not done in AIX.

Related to this, you probably should not use the `-r` option of Configure in AIX, because that affects of how the `nm` tool is used.

## Using GNU's gcc for building Perl

Using gcc-3.x (tested with 3.0.4, 3.1, and 3.2) now works out of the box, as do recent gcc-2.9 builds available directly from IBM as part of their Linux compatibility packages, available here:

```
http://www.ibm.com/servers/aix/products/aixos/linux/
```

## Using Large Files with Perl

Should yield no problems.

## Threaded Perl

Threads seem to work OK, though at the moment not all tests pass when threads are used in combination with 64-bit configurations.

You may get a warning when doing a threaded build:

```
  "pp_sys.c", line 4640.39: 1506-280 (W) Function argument assignment
between types "unsigned char*" and "const void*" is not allowed.
```

The exact line number may vary, but if the warning (W) comes from a line line this

```
  hent = PerlSock_gethostbyaddr(addr, (Netdb_hlen_t) addrlen, addrtype);
```

in the "pp_ghostent" function, you may ignore it safely. The warning is caused by the reentrant variant of gethostbyaddr() having a slightly different prototype than its non-reentrant variant, but the difference is not really significant here.

## 64-bit Perl

If your AIX is installed with 64-bit support, you can expect 64-bit configurations to work. In combination with threads some tests might still fail.

## AIX 4.2 and extensions using C++ with statics

In AIX 4.2 Perl extensions that use C++ functions that use statics may have problems in that the statics are not getting initialized. In newer AIX releases this has been solved by linking Perl with the libC_r library, but unfortunately in AIX 4.2 the said library has an obscure bug where the various functions related to time (such as time() and gettimeofday()) return broken values, and therefore in AIX 4.2 Perl is not linked against the libC_r.

## AUTHORS

H.Merijn Brand <h.m.brand@xs4all.nl> Rainer Tammer <tammer@tammer.net>

## DATE

Version 0.0.10: 07 Aug 2009