AIXPlatform

Compiling on AIX

creating an installp package

The latest version 2.2 and 2.4 distributions include a directory ./build/aix with scripts that will create an installp package. This approach expects you to also download and package the APR and APR-UTIL packages in advance. Unfortunately, the build/aix scripts are not yet included in the APR downloads. They are available at http://dl.aixtools.net/httpd/apr-buildaix-1.5.x.tar and http://dl.aixtools.net/httpd/apr-util-buildaix-1.5.x.tar

Also needed : patches to the APR and APR-UTIL config.layout: http://dl.aixtools.net/httpd/apr-config.layout.patch and http://dl.aixtools.net/httpd/apr-util-config.layout.patch

Preparations

Preparations: see *prerequisites* below as well. One goal of these scripts is to prevent the need for mixing RPM's from other sources (e.g., perlz and/or AIX Toolbox). Mixing libraries will sometimes break things when updates are applied. In short, as much as possible - "generic" BOS AIX is used. To prevent filename collisions /opt/freeware and /usr/local are not used. Likewise, /usr/lib is not intended as a target directory. (Note: the build/aix scripts check for the existence of zlib include files and setup links in when they are missing in the places where ./configure expects them (e.g., /usr/include) - so that you do not need to install a so-called development RPM. These files are available - just not where configure looks).

The build/aix (inspired by build/pkg) scripts also refer to a new layout in ./config.layout – AIX. There is a discussion about this layout below.

NEEDED: * coreutils (for the install to DESTDIR - /var/tmp/root/httpd-version). I can wish otherwise, but apr expects a behavior that only GNU/coreutils install has - both /usr/bin/install and /usr/ucb/install fail during the install to a DESTDIR. (This means if you only want to use "make install" rather than mkinstallp.ksh you can skip installing coreutils). * bos.adt.insttools : this is the fileset that includes the AIX script mkinstallp which is needed by build/aix /mkinstallp.ksh (FYI: mkinstallp.ksh automates creating the template mkinstallp needs)

RECOMMENDED: add /opt/bin to your PATH as first argument. This is the area where the buildaix (available via sourceforge or at aixtools) and build/aix/* scripts put binaries and scripts.

Step by Step

For both versions 2.2 and 2.4 APR and APR-UTIL are needed. So the first step is to download both, and the APR build/aix scripts listed above (until they (I hope) appear in the distributions).

```
# build/aix/buildaix.ksh
# installp -d build/aix all
```

This will create the ASF.apr-vac.rte and ASF.apu-vac.rte and install them in /opt/bin (When using gcc these will be named ASF.apr-gcc.rte and ASF-apu-gcc.rte. APR and APR-UTIL are dependent on the compiler used. The different fileset names helps to identify which compiler is supported. The files installed have the same name and position.

You are almost ready to build HTTPD. For version 2.4 you must have pcre somewhere. You can build this yourself, but easier may be to use a prepackaged one, e.g. from aixtools (http://dl.aixtools.net/httpd/aixtools.pcre.powerpc.8.30.0.I). If all is ready then all you should need is:

```
# build/aix/buildaix.ksh
# installp -d build/aix all
# mkgroup -a daemon
# vi /etc/httpd/httpd.conf # to set your serverName
# /opt/httpd/sbin/apachectl -l # check application without loading any modules
# /opt/httpd/sbin/apachectl -L # check for modules that do not load (e.g., mod_ldap)
```

At this point you can edit /etc/httpd/httpd.conf to comment out LoadModule statements that are preventing /opt/httpd/sbin/httpd from loading.

configure returns an error

My favorite command when that happens is:

```
# grep configure: config.log | tail
```

Note also, build/aix/buildaix accepts additional arguments for configure, for example to disable FEATURE use

```
# build/aix/buildaix.ksh  --disable-FEATURE
```

prerequisites

- xlc and gcc both are reported to work
- It's strongly suggested to install the coreutils RPM and be sure /opt/freeware/bin/install is high in your PATH and found by APR configure for BSD-compatible install.
    - covener believes this might solve failures to install DSO's.
    - This is probably fixed in trunk (by using CP instead of install)
- You should have the "installp" openssl, but you may want to compile your own openssl into /opt/opensslXYZ and pass it to configure with --with-ssl
    - Recover from https://www.ibm.com/support/pages/64-bit-applications-fail-link-against-libssla-or-libcryptoa if it affects your system
    - If you build your openssl, add to LDFLAGS "-R/openssl-prefix" so the CLI's can find their own openssl
- It is very useful to install prerequisites via the yum available with the AIX toolbox for Linux alpha applications. Once you have yum, you can install e.g.
    - autoconf m4 libtool zlib expat expat-devel pcre pcre-devel libnghttp2 libnghttp2-devel
        - There are a lot of transitive dependencies if you try not to use yum or try to build them all yourself
- For something long forgotten, adding that prefix to your LIBPATH helps some stage of the build
    - export PATH=$HOME/gnu/bin:/opt/freeware/bin:$PATH
    - export LIBPATH=$LIBPATH:$HOME/gnu/
- Setting CONFIG_SHELL=/usr/bin/bash can speed up ./configure significantly.
- Old tarballs with an old config.guess/config.sub require AIX 6.1 and later to impersonate AIX 5.3
    - export ac_cv_build="powerpc-ibm-aix5.3.0.0"
    - export ac_cv_build_alias="powerpc-ibm-aix5.3.0.0"
- To build from trunk, you'll need a working python. You can install gdbm and python2.6 from the "AIX Toolbox for Linux" RPMs, or from http://www.perzl.org/.
- If you didn't use yum for expat or pcre::
    - Grab a srclib/expat, or expat and expat-devel from http://www.perzl.org
    - Grab a srclib/pcre, or pcre and pcre-devel from http://www.perzl.org
- To find these prerequisites at runtime:
    - Append /opt/freeware/lib64 to LIBPATH in $bindir/envvars
    - If running the Apache Test Framework, pass -httpd $bindir/bin/apachectl

2.4 specific info

- If you compile mod_unixd as static, you also need this in server/Makefile +EXPORT_DIRS = $(top_srcdir)/include $(top_srcdir)/os/$(OS_DIR) $(top_srcdir)/modules/arch/unix/

64-bit

- Re-configure after changing any of these.
- Set both OBJECT_MODE=64 and CC="xlc_r -q64"
- Sometimes you need to also set:
    - export AR="ar -X64"
    - export lt_cv_path_NM="/usr/bin/nm -B -X64"
- If you're building with SSL:
    - install pkg-config from perzl or toolbox
    - install openssl and openssl-devel from perzl or toolbox
    - If using third-party openssl packages, export LD_LIBRARY_PATH=/opt/freeware/lib64 before running configure

Link errors with bundled expat and when using DESTDIR

- If you use a bundled expat, you might find the link of apr-util fails. The following change might help:

    ```
    perl -p -i.orig -e \"s#[^ ]+libexpat.la#-L$curdir/srclib/apr-util/xml/expat/lib/ -lexpat#; s#[^ ]+libapr-1.la##;\" $PWD/srclib/apr-util/Makefile
    ```

AIX layout discussion

With the latest distributions there is a layout for AIX in config.layout to assist with standardization of packaging on AIX.

The goals:

- man pages in /usr/share/man/...
- read only, system independent files in /opt/httpd/...
- system dependent files in /var/httpd/...
- no files in /usr - except for man pages

so:

- no files in /usr/local
- httpd is used as directory name rather than apache2
- /opt/httpd for binaries and static files (to be a USR part of an installp package)
- /var/httpd as base for files that are system dependent and may be modified
- /var/httpd/etc as base for config files
- /usr/share/man for the man pages so that man httpd-foo works directly
- /etc/httpd/* is used for config files
- this layout will directly support multiple installations in system WPARs
- Optional: If all server related files are to be in a separate filesystem (e.g., /var/httpd you can consider moving /etc/httpd to /var/httpd/etc and creating a symbolic link from /etc/httpd to /var/httpd/etc

Testing on AIX

To test httpd functionality you need (per 2014-08-06) at least perl v5.10 in order to add the needed modules and modules. So, using standard AIX provided perl you must test on AIX 7.1 as AIX 5.3 has perl v5.8.2 and AIX 6.1 has perl v5.8.8.

basic process

- Have apache httpd installed and configured normally (e.g. ServerName is defined, or you will get a warned about this for EVERY test)

```
svn checkout  http://svn.apache.org/repos/asf/httpd/test/framework/trunk ./test
```

This should also download ApacheTest, if not use something like:

```
svn checkout  http://svn.apache.org/repos/asf/perl/Apache-Test/trunk ./test/Apache-Test
```

Now make sure your perl is current and load the bundle ApacheTest. This may take some trials - and time.

```
perl -MCPAN -e 'install Bundle::ApacheTest'
```

With the basics up and ready.

```
# su testuser
% find /opt/httpd -name apxs
% perl Makefile.PL -apxs ''full-path from find above''
% t/TEST
```

Using OpenSSL 1.1

OpemSSL 1.1.0/1.1.1 prior to mid-2018 would produce a SIGILL on shutdown. If you use LoadFile of libcrypto as a workaround, it is safe to remove.