Building Enduro/X On IBM AIX Platform

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
1.1	2018-11	Enduro/X 6.0 release	MV

Contents

1 About manual **1**

2 Overview **2**

3 Operating System Configuration **3**

4 Installation process **4**

4.1 Packages to be installed 4

4.1.1 Install IBM Open Source RPMs 4

4.1.2 Installing missing packages 5

4.2 Getting the source code 5

4.3 Enduro/X basic Environment configuration for HOME directory 6

4.4 Configuring PostgreSQL 6

4.5 Building the code with xlC 8

4.6 Building the code with GCC 8

5 Unit Testing **9**

5.1 UBF/FML Unit testing 9

5.2 XATMI Unit testing 9

6 Trouble shooting **11**

6.1 Problems with libxml2 11

6.2 Rebuilding with other compiler 11

6.3 Thread local storage issues 12

7 Conclusions **13**

8 Additional documentation **14**

8.1 Resources 14

Chapter 1

About manual

This manual describes how to build *Enduro/X* on IBM AIX Platform. Document is based on AIX 7.1 SP2, TL5. Compilers supported are GNU GCC and IBM xlC.

Chapter 2

Overview

This manual includes basic installation of Enduro/X which does not include building of documentation, does not include platform script (as there is issues with c++ headers), and does not use GPG-ME encryption for bridges.

Chapter 3

Operating System Configuration

For short installation process see ex_adminman(guides)(Enduro/X Administration Manual).

Chapter 4

Installation process

The installation process will install required several open source packages. For getting Enduro/X to work basically we need following packages:

1. git
2. cmake
3. flex
4. bison
5. libxml2
6. gcc (if try GNU Compiler Chain)

4.1 Packages to be installed

The approach how packages can be installed on system may differ. This document list just one the ways how to get system working.

4.1.1 Install IBM Open Source RPMs

Firstly for installation process there will be used official IBM Free software RPM repository and only those package missing there will be installed from www.oss4aix.org. Assuming that you have freshly installed Operating System, we need to install **yum** tool, so that all dependencies can be stratified automatically.

Download and install the RPM packages from the yum_bundle.tar file from https://public.dhe.ibm.com/aix/freeSoftware/aixtoolbox/-ezinstall/ppc/ . Ensure that install process does finish with out any errors.

```
# tar -xvf yum_bundle.tar
# cd yum_bundle
# rpm -ivh *.rpm
```

Once yum is up and running, install following packages needed for build:

```
# yum install flex bison libiconv curl openldap openssl zlib \
libxml2 mktemp gcc gcc-cpp libxml2-devel
```

4.1.2 Installing missing packages

The following operations will be done from root user. This will download the all open source packages to local machine to /root/rpms folder.

```
# mkdir /root/rpms
# cd /root/rpms
# ftp -i www.oss4aix.org
# ftp -i www.oss4aix.org
Connected to www.oss4aix.org.
220 FTP Server ready.
Name (www.oss4aix.org:fora): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230-
                *** Welcome to this anonymous ftp server! ***

     You are user 2 out of a maximum of 20 authorized anonymous logins.
     The current time here is Fri Jun 17 14:52:10 2016.
     If you experience any problems here, contact : michael@perzl.org


230 Anonymous login ok, restrictions apply.
ftp> bin
200 Type set to I
ftp> passive
Passive mode on.
ftp>
ftp>
ftp> cd /compatible/aix71
250 CWD command successful
ftp> mget *
227 Entering Passive Mode (178,254,6,100,136,212).
...
```

Once all packages are downloaded. We can start to install required ones. The download includes all of them, due to possible dependencies. But we need to install only two of them:

```
# rpm -i cmake-3.9.1-1.aix6.1.ppc.rpm
# rpm -i mktemp-1.7-1.aix5.1.ppc.rpm
```

4.2 Getting the source code

For test purposes we will prepare new user for which Enduro/X will built (this adds the in the path the /opt/freeware/bin and xlC version 13 compiler. You may modify that of your needs.

```
# useradd -m user1
# su - user1

$ bash
$ cat << EOF >> .profile
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:$HOME/bin:/usr/bin/X11:/sbin:.:/opt/freeware/bin:/opt ↩
    /IBM/xlC/13.1.3/bin
EOF
$ chmod +x .profile
$ source .profile
$ cd /home/user1
$ GIT_SSL_NO_VERIFY=true git clone https://github.com/endurox-dev/endurox
$ cd endurox
$ git config http.sslVerify "false"
```

4.3 Enduro/X basic Environment configuration for HOME directory

This code bellow creates *ndrx_home* executable file which loads basic environment, so that you can use sample configuration provided by Enduro/X in *sampleconfig* directory. This also assumes that you are going to install to *$HOME/endurox/dist* folder. The file bellow will override the sample configuration.

```
$ cat << EOF > $HOME/ndrx_home
#!/bin/bash

# Where app domain lives
export NDRX_APPHOME=/home/user1/endurox
# Where NDRX runtime lives
export NDRX_HOME=/home/user1/endurox/dist/bin
# Debug config too
export NDRX_DEBUG_CONF=/home/user1/endurox/sampleconfig/debug.conf

# NDRX config too.
export NDRX_CONFIG=/home/user1/endurox/sampleconfig/ndrxconfig.xml

# Access for binaries
export PATH=$PATH:/$HOME/endurox/dist/bin

# LIBPATH for .so
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/endurox/dist/lib

# UBF/FML field tables
export FLDTBLDIR=$HOME/endurox/ubftest/ubftab

################################################################################
# In case if building with Postgresql DB database testing support
# or building endurox-java with Oracle DB tests (03_xapostgres), then
# configure bellow setting (demo values provided):
# If so - uncomment bellow
################################################################################
#export EX_PG_HOST=localhost
#export EX_PG_USER=exdbtest
#export EX_PG_PASS=exdbtest1
# currently uses default port
#export EX_PG_PORT=5432
#export EX_PG_DB=xe


EOF

$ chmod +x $HOME/ndrx_home
```

4.4 Configuring PostgreSQL

If Enduro/X PostgreSQL driver is needed to be build for AIX, the PostgreSQL needs to be installed for build and test purposes. Installation will be done by using "yum" tool. It is assumed that it is configured for given OS. To install database on this system, use following commands:

```
$ su - root
# yum update
# yum install postgresql postgresql-contrib postgresql-devel postgresql-libs postgresql- ↩
    server
```

Once installed, database needs to be created:

```
# su - postgres

$ initdb -D /var/lib/postgresql/data
-- and start it up...
$ pg_ctl -D /var/lib/postgresql/data -l logfile start
```

Create startup scripts:

```
# cat << EOF > /etc/rc.d/rc2.d/S99postgres
#!/bin/ksh

################################################
# name: postgres
# Starts or stops Postgresql.
# P.S. Script must be run as root!
################################################

case "\$1" in
start )
        su - postgres -c "pg_ctl -D /var/lib/postgresql/data -l logfile start"
        ;;
stop )
        su - postgres -c "pg_ctl -D /var/lib/postgresql/data -l logfile stop"
        ;;
* )
        echo "Usage: \$0 (start | stop)"
        exit 1
esac

EOF


# chmod +x /etc/rc.d/rc2.d/S99postgres

# ln -s /etc/rc.d/rc2.d/S99postgres /etc/rc.d/rc2.d/K01postgres
```

Now create the database for Enduro/X tests:

```
# su - postgres

$ createuser exdbtest

$ createdb xe

$ psql -d template1

> alter user exdbtest with encrypted password 'exdbtest1';
> grant all privileges on database xe to exdbtest;
> \q
```

Configuration files needs to be updated for authentication and distributed transactions must be enabled too.

Edit **/var/lib/postgresql/data/postgresql.conf**, set "max_prepared_transactions" to 1000.

```
max_prepared_transactions = 1000                # zero disables the feature
```

For access permissions and network configuration, update **/var/lib/postgresql/data/pg_hba.conf**, so that it contains following:

```
local   all             all                                     peer
host    all             all             127.0.0.1/32            md5
host    all             all             ::1/128                 md5
```

Restart PostgreSQL:

```
# su - root
# /etc/rc.d/rc2.d/S99postgres stop
# /etc/rc.d/rc2.d/S99postgres start
```

4.5 Building the code with xlC

It is assumed that xlC is default compiler on the system, thus following shall make the building ok:

```
$ export OBJECT_MODE=64
$ export CC=xlc
$ export CXX=xlC
$ cd /home/user1/endurox
$ cmake -DDEFINE_DISABLEPSCRIPT=ON -DDEFINE_DISABLEDOC=ON\
      -DDEFINE_DISABLEGPGME=ON -DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist .
$ make
$ make install
```

Note that we use in above snippet a exported compiler variables (CC, CXX), this is due to bug in CMake 3.9 where xlC is not detected correctly. See: https://www.ibm.com/developerworks/aix/library/au-aix-install-config-apache-subversions/index.html

4.6 Building the code with GCC

If you previously have installed gcc (C/C++) compiler open source package. Then you can build Enduro/X with GCC compiler. To prepare for GCC build, do following steps:

```
$ cd /home/user1/endurox
$ export OBJECT_MODE=64
$ export CC=gcc
$ export CXX=g++
$ export CFLAGS=-maix64
$ export CXXFLAGS=-maix64
$ cmake -DDEFINE_DISABLEPSCRIPT=ON -DDEFINE_DISABLEDOC=ON -DDEFINE_DISABLEGPGME=ON - ↩
    DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist .
$ make
$ make install
```

Chapter 5

Unit Testing

Enduro/X basically consists of two parts: . XATMI runtime; . UBF/FML buffer processing. Each of these two sub-systems have own units tests.

5.1 UBF/FML Unit testing

```
$ cd /home/user1/endurox/sampleconfig
$ source setndrx
$ cd /home/user1/endurox/ubftest
$ ./ubfunit1 2>/dev/null
Running "main" (76 tests)...
Completed "ubf_basic_tests": 198 passes, 0 failures, 0 exceptions.
Completed "ubf_Badd_tests": 225 passes, 0 failures, 0 exceptions.
Completed "ubf_genbuf_tests": 334 passes, 0 failures, 0 exceptions.
Completed "ubf_cfchg_tests": 2058 passes, 0 failures, 0 exceptions.
Completed "ubf_cfget_tests": 2232 passes, 0 failures, 0 exceptions.
Completed "ubf_fdel_tests": 2303 passes, 0 failures, 0 exceptions.
Completed "ubf_expr_tests": 3106 passes, 0 failures, 0 exceptions.
Completed "ubf_fnext_tests": 3184 passes, 0 failures, 0 exceptions.
Completed "ubf_fproj_tests": 3548 passes, 0 failures, 0 exceptions.
Completed "ubf_mem_tests": 4438 passes, 0 failures, 0 exceptions.
Completed "ubf_fupdate_tests": 4613 passes, 0 failures, 0 exceptions.
Completed "ubf_fconcat_tests": 4768 passes, 0 failures, 0 exceptions.
Completed "ubf_find_tests": 5020 passes, 0 failures, 0 exceptions.
Completed "ubf_get_tests": 5247 passes, 0 failures, 0 exceptions.
Completed "ubf_print_tests": 5655 passes, 0 failures, 0 exceptions.
Completed "ubf_macro_tests": 5666 passes, 0 failures, 0 exceptions.
Completed "ubf_readwrite_tests": 5764 passes, 0 failures, 0 exceptions.
Completed "ubf_mkfldhdr_tests": 5770 passes, 0 failures, 0 exceptions.
Completed "main": 5770 passes, 0 failures, 0 exceptions.
```

5.2 XATMI Unit testing

ATMI testing might take some time. Also ensure that you have few Gigabytes of free disk space, as logging requires some space. Also for AIX there are small default limits of max file size. It is recommended to increase it to some 10 GB or so. To run the ATMI tests do following:

```
$ cd /home/user1/endurox/atmitest
$ nohup ./run.sh &
$ tail -f /home/user1/endurox/atmitest/test.out
```

```
...
************ FINISHED TEST: [test028_tmq/run.sh] with 0 ************
Completed "atmi_test_all": 28 passes, 0 failures, 0 exceptions.
Completed "main": 28 passes, 0 failures, 0 exceptions.
```

Chapter 6

Trouble shooting

6.1 Problems with libxml2

You may experience issues with libxml2 version between free-ware and AIX system provided. The error looks like:

```
$ ./cpmsrv
exec(): 0509-036 Cannot load program ./cpmsrv because of the following errors:
        0509-150   Dependent module /opt/freeware/lib/libxml2.a(libxml2.shr_64.o) could not ↩
            be loaded.
        0509-152   Member libxml2.shr_64.o is not found in archive
```

It seems that linker is using /ccs/lib/libxml2.a but at runtime picks up /opt/freeware/lib/libxml2.a. One way to solve this is to replace freeware version with system provided file. That could be done in following way:

```
# cd /opt/freeware/lib
# mv libxml2.a backup.libxml2.a
# ln -s /usr/ccs/lib/libxml2.a .
```

6.2 Rebuilding with other compiler

To switch the compilers, it is recommended to clean up CMake cached files before doing configuration for other compiler, for example (switching from xlC to GCC):

```
$ rm -rf CMakeCache.txt Makefile CMakeFiles/
$ export OBJECT_MODE=64
$ export CC=gcc
$ gcc
gcc: fatal error: no input files
compilation terminated.
$ export CXX=g++
$ export CFLAGS=-maix64
$ export CXXFLAGS=-maix64
$ cmake -DDEFINE_DISABLEPSCRIPT=ON -DDEFINE_DISABLEDOC=ON -DDEFINE_DISABLEGPGME=ON - ↩
    DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist .
-- The C compiler identification is GNU 4.8.3
-- The CXX compiler identification is GNU 4.8.3
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
```

```
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/g++
-- Check for working CXX compiler: /usr/bin/g++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
...
```

6.3 Thread local storage issues

On AIX 6.1 there with gcc version 4.8.3 works on with *thread flag. However, it looks like On AIX 7.1 with the same gcc version* thread local storage is not working. The symptoms are that various test cases fail, for example test028 (tmqueue). While this happens it is recommended to use xlC compiler.

Chapter 7

Conclusions

At finish you have a configured system which is read to process the transactions by Enduro/X runtime. It is possible to copy the binary version (*dist*) folder to other same architecture machines and run it there with out need of building.

Chapter 8

Additional documentation

8.1 Resources

[1] [BINARY_INSTALL] See Enduro/X binary_install manual.