Building Enduro/X On FreeBSD Platform

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1.0	2016-06	Initial draft	MV

Contents

Chapter 1

About manual

This manual describes how to build *Enduro/X* FreeBSD platform. Document is based on FreeBSD 10.3 on amd64 machine. Default FreeBSD C/C++ compiler is used.

Chapter 2

Overview

This manual includes basic installation of Enduro/X which does not include building of documentation and does not use GPG-ME encryption for bridges.

Chapter 3

Operating System Configuration

For OS configuration settings see ex_adminman(guides)(Enduro/X Administration Manual, Setup System chapter). This step is mandatory be executed, before continuing.

Chapter 4

Installation process

For getting Enduro/X to work basically we need following packages:

1. git
2. cmake
3. flex
4. bison
5. libxml2
6. gcc/g++
7. bash

4.1 Packages to be installed

The following operations will be done from root user. This will download and install open source packages to local machine:

```
# pkg install cmake flex bison libxml2 git bash
```

4.2 Getting the source code

For test purposes we will parepare new user for which Enduro/X will built.

```
# adduser
Username: user1
Full name: Test user
Uid (Leave empty for default):
Login group [user1]:
Login group is user1. Invite user1 into other groups? []:
Login class [default]:
Shell (sh csh tcsh git-shell nologin) [sh]:
Home directory [/home/user1]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
```

```
Enter password again:
Lock out the account after creation? [no]:
Username   : user1
Password   : *****
Full Name  : Test user
Uid        : 1002
Class      :
Groups     : user1
Home       : /home/user1
Home Mode  :
Shell      : /bin/sh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (user1) to the user database.
Add another user? (yes/no): no
Goodbye!

-- Set bash as shell
# chsh -s /usr/local/bin/bash root
# su - user1
$ bash
$ git clone https://github.com/endurox-dev/endurox
$ cd endurox
```

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

echo "Loading ndrx_home..."
# Where app domain lives
export NDRX_APPHOME=$HOME/endurox
# Where NDRX runtime lives
export NDRX_HOME=$HOME/endurox/dist/bin
# Debug config too
export NDRX_DEBUG_CONF=$HOME/endurox/sampleconfig/debug.conf
# NDRX config too.
export NDRX_CONFIG=$HOME/endurox/sampleconfig/ndrxconfig.xml

export PATH=$PATH:$HOME/projects/endurox/dist/bin

export FLDTBLDIR=$HOME/endurox/ubftest/ubftab

export PATH=$PATH:/opt/csw/bin:$HOME/endurox/dist/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/endurox/dist/lib

# Where the queues live:
export NDRX_QPATH=/mnt/mqueue

################################################################################
# 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 FreeBSD, the PostgreSQL needs to be installed for build and test purposes. To install database on this system, use following commands:

```
$ sudo pkg install postgresql96-server postgresql96-client postgresql96-contrib

-- Add postgres to startup:
$ sudo sysrc postgresql_enable=yes
```

Once PostgreSQL is installed, update the configuration and create the database. Create database first and start the server:

```
$ sudo service postgresql initdb
$ sudo service postgresql start
```

Now create the database for Enduro/X tests:

```
$ sudo -s

$ 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/db/postgres/data96/postgresql.conf**, set "max_prepared_transactions" to 1000.

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

For access permissions and network configuration, update **/var/db/postgres/data96/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:

```
$ sudo service postgresql restart
```

4.5 Building the code

```
$ cd /home/user1/endurox
$ cmake -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 (about ~10 GB).

```
$ 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

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 7

Additional documentation

7.1 Resources

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