

# Building Enduro/X On FreeBSD Platform

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

# Contents

<b>1</b>	<b>About manual</b>	<b>1</b>
<b>2</b>	<b>Overview</b>	<b>2</b>
<b>3</b>	<b>Installation process</b>	<b>3</b>
3.1	Packages to be installed . . . . .	3
3.2	Configuring the system . . . . .	3
3.3	Getting the source code . . . . .	4
3.4	Enduro/X basic Environment configuration for HOME directory . . . . .	4
3.5	Building the code . . . . .	5
<b>4</b>	<b>Unit Testing</b>	<b>6</b>
4.1	UBF/FML Unit testing . . . . .	6
4.2	XATMI Unit testing . . . . .	6
<b>5</b>	<b>Conclusions</b>	<b>8</b>
<b>6</b>	<b>Additional documentation</b>	<b>9</b>
6.1	Resources . . . . .	9

## 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

# 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

### 3.1 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
```

### 3.2 Configuring the system

Queue file system must be mounted when OS starts. Firstly we need a folder */mnt/mqueue* where the queues are mount. And secondly we will add the automatic mount at system startup in */etc/fstab*.

```
# mkdir /mnt/mqueue
# cat << EOF >> /etc/fstab
null    /mnt/mqueue    mqueuefs      rw          0            0
EOF
# mount /mnt/mqueue
```

You also need to change the queue paramters:

```
# cat << EOF >> /etc/sysctl.conf

# kernel tunables for Enduro/X:
kern.mqueue.curmq=1
kern.mqueue.maxmq=30000
```

```
kern.mqueue.maxmsgsize=64000
kern.mqueue.maxmsg=1000

EOF

# sysctl -f /etc/sysctl.conf
```

Enduro/X testing framework uses */bin/bash* in scripting, thus we must get it working:

```
# ln -s /usr/local/bin/bash /bin/bash
```

**reboot** to apply new settings (limits & mqueue mount)

### 3.3 Getting the source code

For test purposes we will prepare 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!

# su - user1
$ bash
$ git clone https://github.com/endurox-dev/endurox
$ cd endurox
```

### 3.4 Enduro/X basic Environment configuration for HOME directory

This code below 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 below 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

EOF

$ chmod +x $HOME/ndrx_home
```

### 3.5 Building the code

```
$ cd /home/user1/endurox
$ cmake -DDEFINE_DISABLEDDOC=ON -DDEFINE_DISABLEGPGME=ON -DCMAKE_INSTALL_PREFIX:PATH=`pwd`/ ←
    dist .
$ make
$ make install
```



## Chapter 4

# 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.

### 4.1 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.
```

### 4.2 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 5

# 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 6

# Additional documentation

### 6.1 Resources

[1] [BINARY\_INSTALL] See Enduro/X binary\_install manual.