Building Enduro/X On Oracle Solaris 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* Oracle Solaris platform. Document is based on Solaris 11 on x86 machine. Compiler used to Enduro/X is GCC.

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.

Enduro/X on Solaris platform is Using System V message queues.

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

The installation process will install required pen source packages from *http://www.opencsw.org*. You may install packages with different approach. This is just a sample process for getting build system working on under Solaris. For getting Enduro/X to work basically we need following packages:

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

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:

```
$ su - root
# pkgadd -d http://get.opencsw.org/now
# /opt/csw/bin/pkgutil -U
# /opt/csw/bin/pkgutil -y -i git libxml2_dev flex bison cmake gmake
```

4.2 If installing GCC...

```
$ su - root
# /opt/csw/bin/pkgutil -y -i gcc4core gcc4g++
```

4.3 If installing Sun Studio (on Solaris 11)...

- According to: https://pkg-register.oracle.com/register/product_info/6/
- Register and request access, download the and pkg.oracle.com.key.pem pkg.oracle.com.certificate.pem from Oracle to server.

Installation shows process for 12.4 version, but any later available should be installed. Also note that **.profile** and **$HOME/n-drx_home** later needs to match the path, for example **/opt/developerstudio12.6**.

```
$ su - root
# mkdir -m 0775 -p /var/pkg/ssl
# cp -i download-directory/pkg.oracle.com.key.pem /var/pkg/ssl
# cp -i download-directory/pkg.oracle.com.certificate.pem /var/pkg/ssl

# pkg set-publisher \
-k /var/pkg/ssl/pkg.oracle.com.key.pem \
-c /var/pkg/ssl/pkg.oracle.com.certificate.pem \
-G '*' -g https://pkg.oracle.com/solarisstudio/release solarisstudio

# pkg list -af 'pkg://solarisstudio/developer/solarisstudio-124/*'

# pkg install -nv solarisstudio-124

# pkg install solarisstudio-124
```

If plan to run

4.4 If installing Sun Studio (on Solaris 10)...

Firstly download the "tarfile" version from Oracle, visit http://www.oracle.com/technetwork/server-storage/solarisstudio/downloads/-tarfiles-studio-12-4-3048109.html

Next once the tar file is on your server, extract it and start the install:

```
# bzip2 -d SolarisStudio12.4-solaris-x86-bin.tar.bz2
# tar -xf SolarisStudio12.4-solaris-x86-bin.tar
# cd SolarisStudio12.4-solaris-x86-bin
# ./install_patches.sh
# mv solarisstudio12.4 /opt
```

For Solaris 10 also we need gmake to work as "make", thus

```
# ln -s /opt/csw/bin/gmake /usr/bin/make
```

This will put the compiler in "/opt/solarisstudio12.4" the path later used in this tutorial.

Also for solaris 10 we need "ar" tool which we will use from GNU package:

```
# /opt/csw/bin/pkgutil -y -i gcc4core gcc4g++
```

Chapter 5

Getting the source code

Firstly we need to add "user1" under which we will perform build actions. For test purposes we will parepare new user for which Enduro/X will built (this adds the in the path the /opt/csw/bin. You may modify that of your needs. (add /opt/solarisstudio12.4/bin if Sun studio is installed)

5.1 Adding user on Solaris 11

We add the user "user1" and also set the open file limit to 4096, by default it is 256 which is too low for unit testing.

```
$ su - root
# useradd -m user1
# passwd user1
# projadd -K "process.max-file-descriptor=(basic,10000,deny)" enduroxproject
# usermod -K "project=enduroxproject" user1
# su - user1
```

5.2 Adding user on Solaris 10

```
$ su - root
# useradd -d \$HOME -m -s /usr/bin/bash -c "User1 User1" user1
# mkdir \$HOME
# chown user1:staff \$HOME
# su - user1
```

5.3 Preparing the user environment

```
$ bash
$ cat << EOF >> .profile
export PATH=\$PATH:/opt/csw/bin:/opt/solarisstudio12.4/bin
# Needed if building with postgres...
export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/opt/csw/lib/64
EOF
$ chmod +x .profile
$ source .profile
$ cd \$HOME
$ GIT_SSL_NO_VERIFY=true git clone https://github.com/endurox-dev/endurox
$ cd endurox
$ git config http.sslVerify "false"
```

5.4 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 FLDTBLDIR=\$HOME/endurox/ubftest/ubftab

export PATH=/usr/bin:/usr/sbin:/opt/csw/bin:/opt/solarisstudio12.4/bin:/opt/csw/bin:\$HOME/ ↩
    endurox/dist/bin
export LD_LIBRARY_PATH=/usr/lib/sparcv9:\$HOME/endurox/dist/lib64:/opt/solarisstudio12.4/ ↩
    lib:/usr/lib64:/opt/csw/lib/64

################################################################################
# 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

#
# If using PostgreSQL for Solaris 11 uncomment:
#
#export PATH=$PATH:/opt/csw/libexec/postgresql/93


EOF

$ chmod +x $HOME/ndrx_home
```

5.5 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. On Solaris 10, PostgreSQL comes with the operating system, thus only access rights and users needs to be configured.

Note if you plan to use ECPG mode, then ecpg pre-compiler needs to know where the ecpg libraries live. Thus LD_LIBRARY_PATH must be set during the build time.

It can be done in following way:

```
# su - user1
$ cat << EOF >> .profile
export LD_LIBRARY_PATH=/opt/csw/lib/amd64
EOF
```

after this login and log out from user1 to apply the .profile settings.

Also to active the PostgreSQL inclusion to the build, add **-DENABLE_POSTGRES=ON** flag to cmake command line (later in build section).

For Solaris 11 it needs to be installed:

```
$ su - root

-- Install with:
# /opt/csw/bin/pkgutil -y -i postgresql93

-- Install dev
# /opt/csw/bin/pkgutil -y -i postgresql_dev

-- enable for auto start
# svcadm enable cswpostgresql-93

# su - postgres

-- Create profile entry to have path to postgres binaries

$ cat << EOF > ~/.profile

#!/bin/bash

export PATH=$PATH:/opt/csw/libexec/postgresql/93/

EOF

$ chmod +x ~/.profile

-- Start postgres from Postgres user
$ /opt/csw/bin/pg_ctl-93 -D /var/opt/csw/postgresql/93 -l logfile start
server starting
```

For Solaris 10, just enable it:

```
$ su - root
# svcadm enable svc:/application/database/postgresql:version_82
```

Now create the database for Enduro/X tests (Solaris 10 & 11):

```
# 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 **postgresql.conf**, set "max_prepared_transactions" to 1000.

Solaris 10: **/var/postgres/8.2/data/postgresql.conf**

Solaris 11(.4): **/var/opt/csw/postgresql/93/postgresql.conf**

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

For access permissions and network configuration, update **pg_hba.conf**, so that it contains following:

Solaris 10: **/var/postgres/8.2/data/pg_hba.conf**

Solaris 11: **/var/opt/csw/postgresql/93/pg_hba.conf**

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

Restart PostgreSQL, Solaris 10:

```
# svcadm restart svc:/application/database/postgresql:version_82
```

Restart PostgreSQL, Solaris 11:

```
# svcadm restart cswpostgresql-93
```

Chapter 6

Building the code with GCC

It is assumed that gcc is default compiler on the system (i.e. Oracle Studio not installed), thus following cmake will pick up gcc by default:

6.1 Solaris 11

```
$ cd \$HOME/endurox
$ cmake -DCMAKE_CXX_COMPILER=g++ -DCMAKE_C_COMPILER=gcc\
-DDEFINE_DISABLEDOC=ON -DDEFINE_DISABLEGPGME=ON -DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist - ↩
    DCMAKE_LIBRARY_PATH=/opt/csw/lib/amd64 .
$ make
$ make install
```

6.2 Solaris 10

Also note that CC variable needs to be exported as it is used by buildclient script for view test cases.

```
$ export CC=gcc
$ cd \$HOME/endurox
$ cmake -D CMAKE_AR=/opt/csw/gnu/ar -DCMAKE_CXX_COMPILER=g++ -DCMAKE_C_COMPILER=gcc \
-DDEFINE_DISABLEDOC=ON -DDEFINE_DISABLEGPGME=ON -DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist .
$ make
$ make install
```

Chapter 7

Building the code with Solaris Studio

The compilation will be done in 64bit mode

7.1 Solaris 11

```
$ cd \$HOME/endurox
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist -DDEFINE_DISABLEGPGME=ON -DDEFINE_DISABLEDOC ↩
    =ON .
```

In case if errors like

```
ld: fatal: file /usr/lib/values-xpg6.o: wrong ELF class: ELFCLASS32
```

appears, temporary solution is to replace that particular file with 64bit version. It appears that Solaris Studio compiler ignores the "-m64" architecture flags and does not use "/usr/lib/amd64/values-xpg6.o" where it requires.

```
# su - root
# mv /usr/lib/values-xpg6.o /usr/lib/values-xpg6.o.OLD
# ln -s /usr/lib/amd64/values-xpg6.o /usr/lib/values-xpg6.o
```

7.2 Solaris 10 (NOT SUPPORTED)

This assumes that GCC is installed, and "ar" from gcc will be used.

```
$ cd \$HOME/endurox
$ cmake -D CMAKE_AR=/opt/csw/gnu/ar \
-DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist -DDEFINE_DISABLEGPGME=ON -DDEFINE_DISABLEDOC=ON .
```

The support is not available for Solaris Studio on Solaris 10 due to Thread Local Storage errors like during the linking:

```
ld: fatal: relocation error: R_SPARC_TLS_LDO_LOX10: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.first: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
ld: fatal: relocation error: R_SPARC_TLS_LDO_ADD: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.first: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
ld: fatal: relocation error: R_SPARC_TLS_LDO_HIX22: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.ostid: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
```

```
ld: fatal: relocation error: R_SPARC_TLS_LDO_LOX10: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.ostid: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
ld: fatal: relocation error: R_SPARC_TLS_LDO_ADD: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.ostid: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
ld: fatal: relocation error: R_SPARC_TLS_LDO_HIX22: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.ostid: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
ld: fatal: relocation error: R_SPARC_TLS_LDO_LOX10: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.ostid: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
ld: fatal: relocation error: R_SPARC_TLS_LDO_ADD: file CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    symbol $XBaBAqASPv3bHaz.__ndrx_debug__.ostid: bound to: CMakeFiles/nstd.dir/ndebug.c.o:  ↩
    relocation illegal when not bound to object being created
```

Thus at this time only GCC is supported for Solaris 10.

Chapter 8

Building the code

```
$ cd \$HOME/endurox
$ make
$ make install
```

This will produce binaries in *\$HOME/endurox/dist* folder.

Chapter 9

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.

9.1 UBF/FML Unit testing

```
$ cd \$HOME/endurox/sampleconfig
$ source setndrx
$ cd \$HOME/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.
```

9.2 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/endurox/atmitest
$ nohup ./run.sh &
$ tail -f \$HOME/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 10

Troubleshooting

This section lists some notes about fixing most common problems with Solaris build.

10.1 Problems with library modes

If having issues with linking particular library version, for example building in 64bit mode, but for some reason CMake is linking with 32bit libs (for example with PostgreSQL), then following flag may be applied **CMAKE_LIBRARY_PATH** to point to correct path for libraries. For example:

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=`pwd`/dist -DDEFINE_DISABLEDOC=ON\
    -DENABLE_POSTGRES=ON -DDEFINE_DISABLEGPGME=ON -DCMAKE_LIBRARY_PATH=/opt/csw/lib/amd64 .
```

Chapter 11

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 12

Additional documentation

12.1 Resources

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