JPN_IWE14057: TEST SPECIFICATION

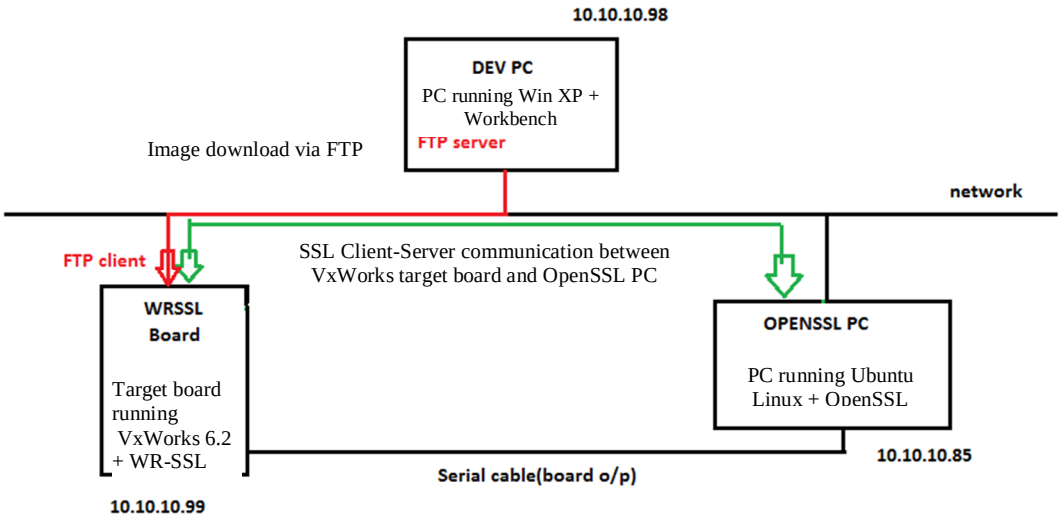
Step-by-step guide to test JPN_IWE14057 to Wind River VxWorks 6.2

7-July 15
Abiram Warrier

Contents

Development and Test Environment

Development and test environment for the JPN-IWE14057 project is as shown below.



Ensure that required hardware and software are available to realize below setup before following instructions in this document.

Document also assumes that above setup is realized with required physical connections made as shown in above diagram.

1. Installatio/build Instructions

On DEV PC:

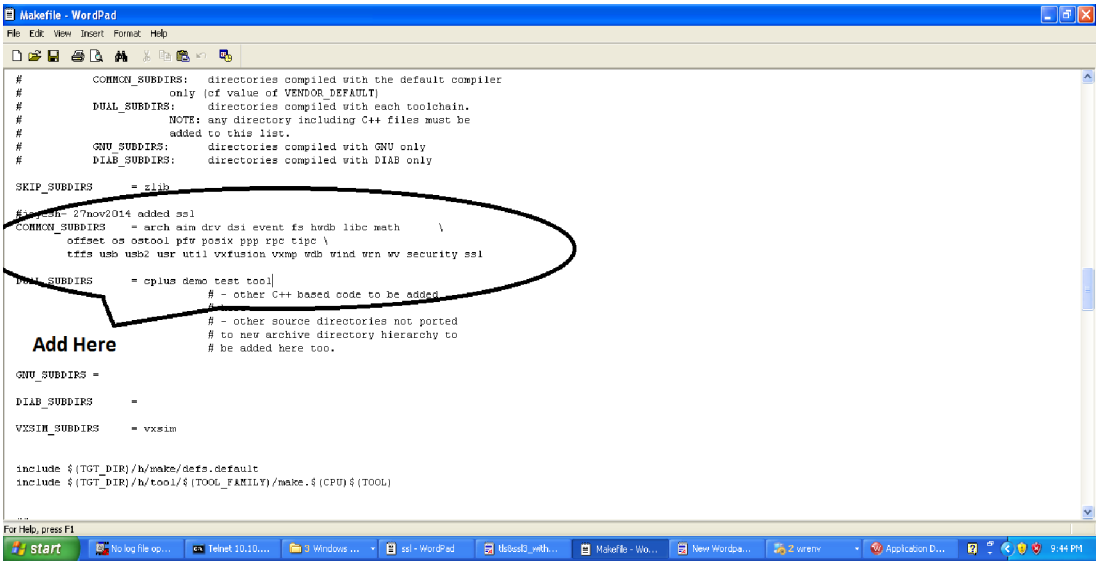
1. Install Wind River Workbench and vxWorks 6.2 and BSPs.

On DEV-PC:

1. Modify following files as described below
   1. **File Location**
   <WorkbenchInstallationDirectory>\VxWorks-6.2\target\src\Makefile

   Changes required:
   -> Search for "COMMON_SUBDIRS" (where you will see wind,wrn,wvlisted)

-> Append "security" and "ssl" to the list of COMMON_SUBDIRS as shown in below fig.



2. File Location

<WorkbenchInstallationDirectory>\VxWorks-6.2\target\config\all\usrConfig.c

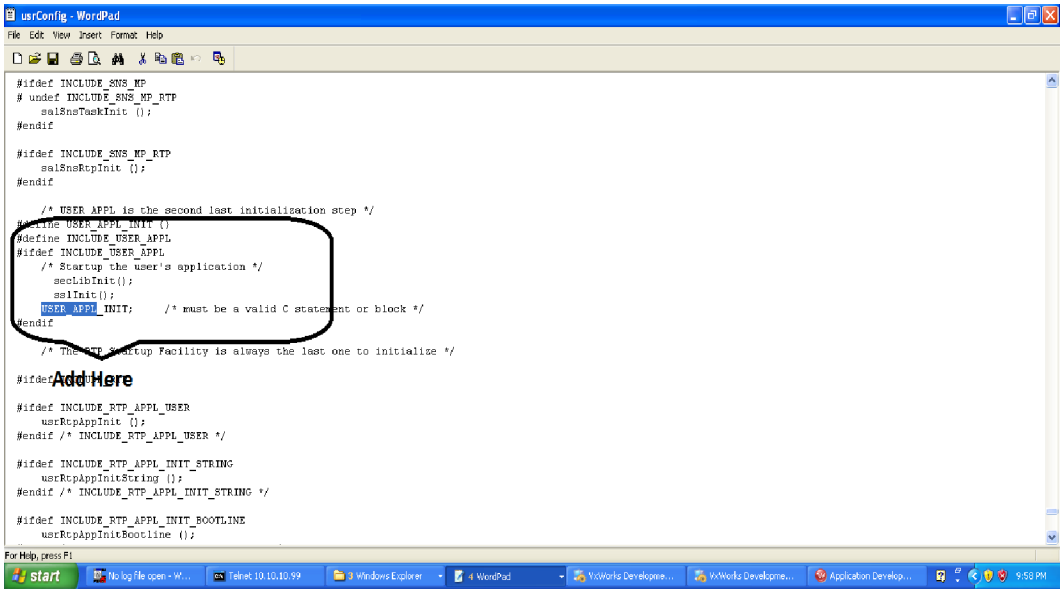
Changes required:

-> define INCLUDE_USER_APPL macro by adding below line

#define INCLUDE_USER_APPL

->In the INCLUDE_USER_APPL block add below two lines to call those functions as shown in below fig.

```
secLibInit();
sslInit();
```

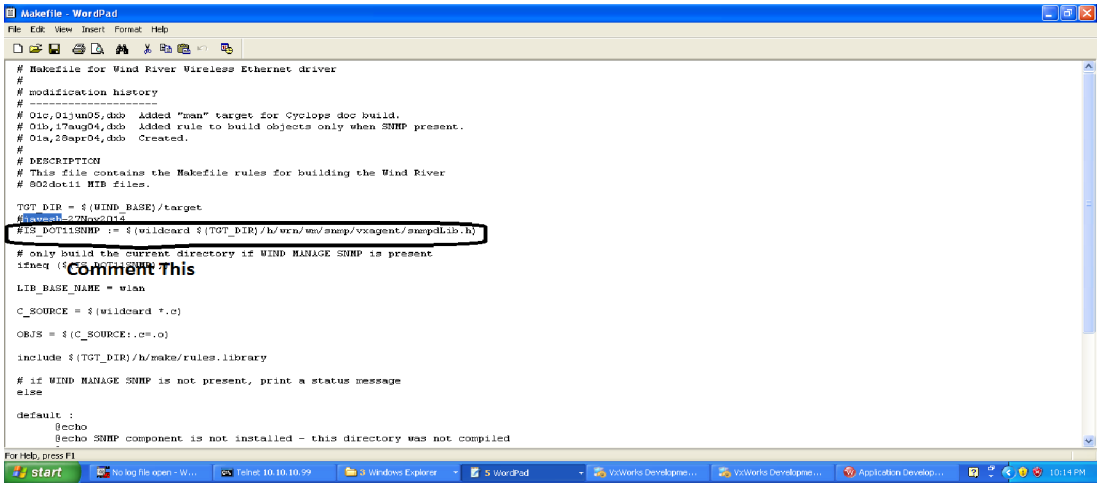


3. **File Location**

   <WorkbenchInstallationDirectory>\VxWorks-6.2\target\src\drv\wlan\management\Makefile

   Changes required:

   - search for "IS_DOT11SNMP" and comment that line as shown in below fig.

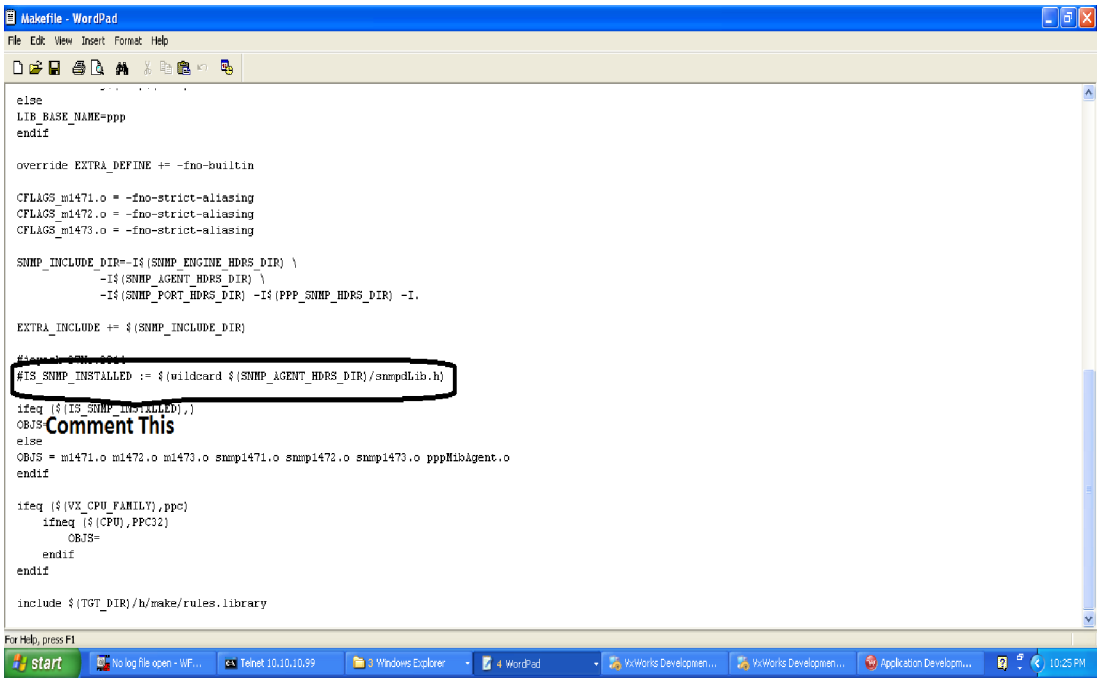


4. **File Location**

   <WorkbenchInstallationDirectory>\VxWorks-6.1\target\src\ppp\management\snmpAgent\Makefile

   Changes required:

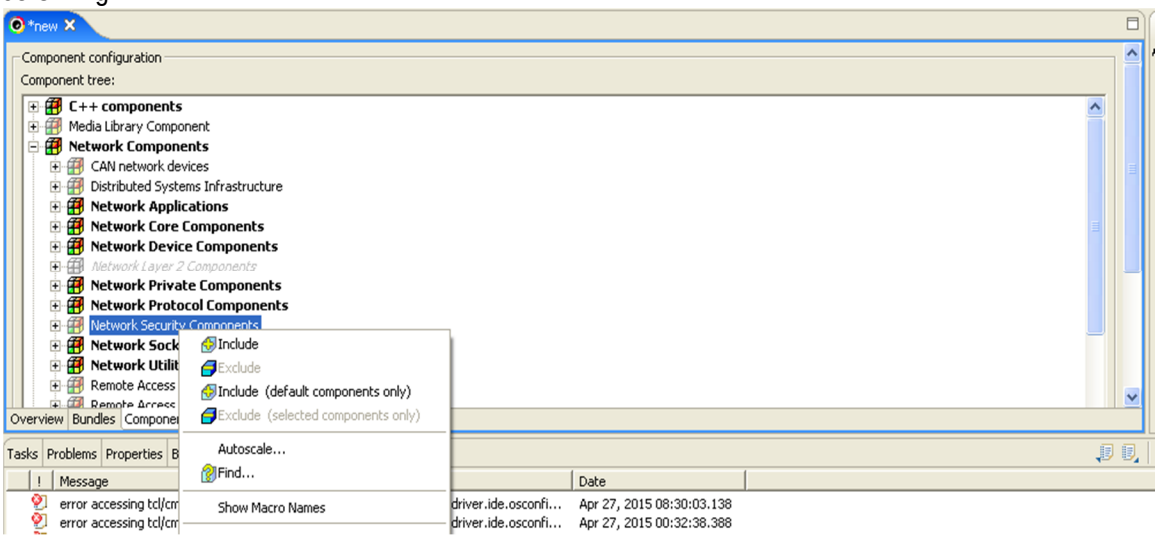
   - search for "IS_SNMP_INSTALLED" and comment that line as shown in below fig.



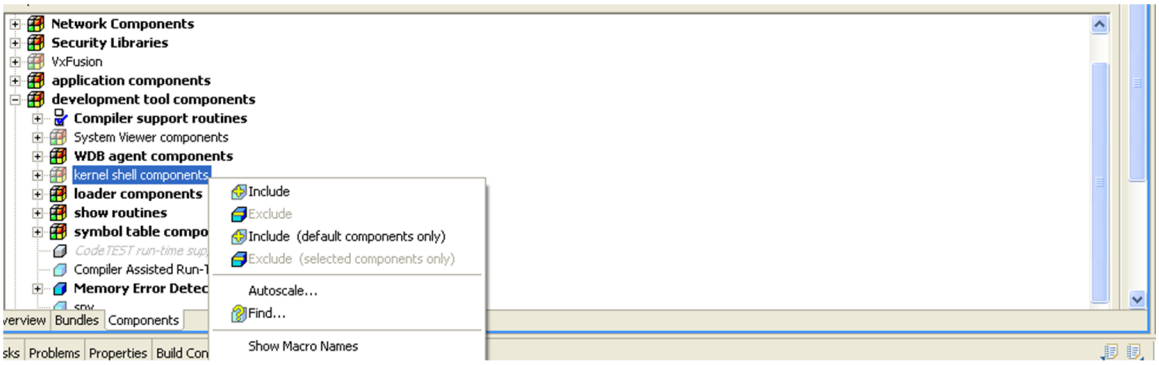
2. Build VxWorks-6.2 for specific CPU(for ex: PPC32,PENTIUM4) by following below steps:

    1. Open **VXWorksDevlopmentShell(start->all programs->windriver->vxworks-6.2->vxworks development shell).**
    2. Goto directory <WorkbenchInstallationDirectory>**\VxWorks-6.2\target\src**
    3. Build using 'make' command as make CPU=<cpu name> TOOL=<diab>
       e.g :make CPU=PPC32 TOOL=diab

3. Create Image Project Under WindRiver Workbench
    1. Open Workbench GUI(**start->All programs->windriver->workbench 2.4**).
    2. Create new VxWorksimage project from Workbench-2.4 menu(**file->new->vxworks image project**).
    3. Give project name and click on 'next' .
    4. Select BSP(Board Supporting Package) and tool chain and click on 'finish'. (in our case we have selected wrsbcPowerQuiccII as BSP and diab as tool chain)
    5. Goto kernel configuration of the project from project window.
    6. In component tree, enable vxWorkscomponents INCLUDE_SSL and INCLUDE_SSL_APPS by including **network security components** under **network components** as shown in below fig.



    7. Include **Kernel shell Components**under**Development Tools Components(select all)**



8. Build/Rebuild Project.
9. By default VxWorks image will be present in following directory.<WindRiverInstalledDirectory>\workspace\project_name\default\VxWorks

Bringing up Wind River Board

1. Power on the WindRiver targetboard.
2. Target board should be connected to OPENSSL PC using serial port.[use picocom for serial input/output or to see the console log of board.].
   i.e.picocom -b 9600 /dev/ttyS0
3. Make a LAN connection between Target board and system in which ftp server is running, i.e., DEV PC.[FTP server is part of windriver package and available under **start->all programs->windriver->vxworks-6.2->ftp server]**.
4. Board and DEV PC should be on same network.
5. Configure ftp server as below:
   - Launch ftp server from **start->all programs->windriver->vxworks-6.2->ftp server.**
   - Select **security->user/rights.** Pop up window as shown below fig will open.
   - Select username(if no user available then create new by pressing 'new user') and set password and home directory for that user.(set the directory where vxworks images will be present as the home directory of ftp server).

6. Switch off and switch on the board and wait for '[VxWorks Boot]'command prompt.
7. Configure board as below: (press 'c' to get configure window)
   - Select boot device(type 'help' to get the list of supported boot device).
   - set appropriate IP address for board.
   - host ip and gateway should be same as FTP server system's IP address.
   - FTP user and password field should match with FTP server configuration.
   - following fig. shows sample example

```
[VxWorks Boot]: @

boot device          : motfcc
unit number          : 0
processor number     : 0
host name            : host
file name            : vxWorks
inet on ethernet (e) : 10.10.10.99:ffffff00
host inet (h)        : 10.10.10.100
gateway inet (g)     : 10.10.10.100
user (u)             : target
ftp password (pw)    : target
flags (f)            : 0x0
other (o)            : motfcc0
```

8. Once board configuration is done, '[VxWorks Boot]' prompt will be displayed again. You may use command 'p' to ensure that configuration is correct.
9. Copy Vxworks images created to FTP home directory.

4. Test Environment Setup & Execution Procedure

A. Test Preparation

On OPENSSL-LINUX PC

1. download and install OPENSSL-1.0.1o from https://www.openssl.org/source/openssl-1.0.1o.tar.gz

2. Following certificates are to be generated and should be kept under respective folders.

md5WithRSAEncryption

```
$openssl genrsa -out md5rsakey.pem 2048
$openssl req -out md5rsa.csr -key md5rsakey.pem -new -md5
$openssl req -in md5rsa.csr -out md5rsa.pem -text
```

sha1WithRSAEncryption

```
$openssl genrsa -out sha1rsakey.pem 2048
$openssl req -out sha1rsa.csr -key sha1rsakey.pem -new -sha1
$openssl req -in sha1rsa.csr -out sha1rsa.pem -text
```

sha256WithRSAEncryption

```
$openssl genrsa -out sha256rsakey.pem 2048
$openssl req -out sha256rsa.csr -key sha256rsakey.pem -new -sha256
$openssl req -in sha256rsa.csr -out sha256rsa.pem -text
```

SHA1withECDSA

```
$openssl ecparam -name secp256k1 -genkey -param_enc explicit -out
 sha1ecdsakey.pem
$openssl req -new -x509 -key sha1ecdsakey.pem -out sha1ecdsa.pem -days
 730
```

Note: You may verify the key and the certificate contents using

```
$openssl ecparam -in sha1ecdsakey.pem -text –noout
$openssl x509 -in sha1ecdsa.pem -text -noout
```

sha256withECDSA

```
$openssl ecparam -name secp256k1 -genkey -param_enc explicit -out
 sha256ecdsakey.pem
$openssl req -new -x509 -key sha256ecdsakey.pem -out sha256ecdsa.pem -
 days 730
```

Note: You may verify the key and the certificate contents using

```
$openssl ecparam -in sha256ecdsakey.pem -text -noout
$openssl x509 -in sha256ecdsa.pem -text -noout
```

3. Copy above generated certificate and key files to openssl-1.0.1o folder on OPENSSL-LINUX PC
4. Transfer above generated certificate and key files (via FTP or any other method) to home directory of FTP server on DEV-PC.

B. Test Procedure

CASE: A01

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "md5rsa.pem", "-key", "md5rsakey.pem","-
       ssl2")
```

2. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl2 -cipher DES-CBC3-MD5 -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A02

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

   e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert md5rsa.pem –key
  md5rsakey.pem -ssl2 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

2. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-ssl2","-cipher","DES-CBC3-MD5","-
       connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A03

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
->nm_server_main("5","-cert", "md5rsa.pem", "-key", "md5rsakey.pem","-
       ssl2")
```

On OPENSSL-LINUX PC

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

```
$ cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl2 -cipher RC4-MD5 -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A04

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

```
$ cd /home/user1/openssls/openssl-1.0.1o
$ ./apps/openssl s_server -accept 4433 -cert md5rsa.pem -key
  md5rsakey.pem -ssl2 -msg
```

2. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-ssl2","-cipher","RC4-MD5","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A05

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal

```
-> nm_server_main("5","-cert", "sha1rsa.pem", "-key",
"sha1rsakey.pem","-ssl3")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl3 -cipher ECDHE-RSA-AES256-SHA -connect
       <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A06

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

```
$ cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha1rsa.pem -key
       sha1rsakey.pem -ssl3 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
->nm_client_main("5","-ssl3","-cipher","ECDHE-RSA-AES256-SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A07

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha1ecdsa.pem", "-key",
"sha1ecdsakey.pem","-ssl3")
```

On OPENSSL-LINUX -PC:

2. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$ cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl3 -cipher ECDHE-ECDSA-AES256-SHA -connect
       <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A08

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below
   e.g.
   ```
   $cd /home/user1/openssls/openssl-1.0.1o
   $./apps/openssl s_server -accept 4433 -cert sha1ecdsa.pem -key
          sha1ecdsakey.pem -ssl3 -msg
   ```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.
   ```
   -> nm_client_main("5","-ssl3","-cipher","ECDHE-ECDSA-AES256-SHA","-
   connect","<OPENSSL_PC_IPAddr>:4433")
   ```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A09

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.
   ```
   -> nm_server_main("5","-cert", "sha1rsa.pem", "-key",
   "sha1rsakey.pem","-ssl3")
   ```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below
   ```
   $ cd /home/user1/openssls/openssl-1.0.1o
   $./apps/openssl s_client -ssl3 -cipher ECDHE-RSA-DES-CBC3-SHA -connect
          <host>:4433
   ```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A10

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

   e.g.

```
$ cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha1rsa.pem -key
       sha1rsakey.pem -ssl3 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL PC picocom terminal.

```
-> nm_client_main("5","-ssl3","-cipher","ECDHE-RSA-DES-CBC3-SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A11

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
->nm_server_main("5","-cert", "sha1rsa.pem", "-key",
"sha1rsakey.pem","-ssl3")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$ cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl3 -cipher DES-CBC3-SHA -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A12

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

```
$ cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha1rsa.pem -key
       sha1rsakey.pem -ssl3 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-ssl3","-cipher","DES-CBC3-SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client

terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A13

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha1rsa.pem", "-key",
"sha1rsakey.pem","-ssl3")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl3 -cipher ECDHE-RSA-RC4-SHA -connect
       <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A14

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha1rsa.pem -key
       sha1rsakey.pem -ssl3 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
➔ nm_client_main("5","-ssl3","-cipher","ECDHE-RSA-RC4-SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

Case:A15

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
->nm_server_main("5","-cert", "sha1ecdsa.pem", "-key",
"sha1ecdsakey.pem","-ssl3")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -ssl3 -cipher ECDHE-ECDSA-RC4-SHA -connect
       <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A16

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha1ecdsa.pem -key
       sha1ecdsakey.pem -ssl3 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-ssl3","-cipher","ECDHE-ECDSA-RC4-SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A17

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key",
"sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher TLS_RSA_WITH_RC4_128_SHA -connect
       <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A18

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below
   e.g.
   ```
   $cd /home/user1/openssls/openssl-1.0.1o
   $./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
          sha256rsakey.pem -tls1 -msg
   ```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.
   ```
   -> nm_client_main("5","-tls1","-
   cipher","TLS_RSA_WITH_RC4_128_SHA","-
   connect","<OPENSSL_PC_IPAddr>:4433")
   ```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A19

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.
   ```
   -> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
   sha256rsakey.pem","-tls1")
   ```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below
   ```
   $cd /home/user1/openssls/openssl-1.0.1o
   $./apps/openssl s_client -tls1 -cipher TLS_RSA_WITH_DES_CBC_SHA  -
          connect <host>:4433
   ```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A20

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
```

```
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_RSA_WITH_DES_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A21

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher TLS_RSA_WITH_3DES_EDE_CBC_SHA
-connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A22

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_RSA_WITH_3DES_EDE_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A23

On OPENSSL-LINUX -PC:

2. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below e.g.

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher TLS_RSA_WITH_AES_256_CBC_SHA
-connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A24

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_RSA_WITH_AES_256_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A25

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

```
$cd  /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_RSA_WITH_RC4_128_SHA -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A26

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_RSA_WITH_RC4_128_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A27

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A28

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A29

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA  -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client.

Connection should get closed without problems.

CASE: A30

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A31

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256ecdsa.pem", "-key",
"sha256ecdsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A32

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
```

```
$./apps/openssl s_server -accept 4433 -cert sha256ecdsa.pem -key
sha256ecdsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_ECDSA_WITH_RC4_128_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A33

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256ecdsa.pem", "-key", "
sha256ecdsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A34

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256ecdsa.pem -key
sha256ecdsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A35

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256ecdsa.pem", "-key", "
sha256ecdsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA  -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A36

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

   e.g.

```
$cd  /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256ecdsa.pem -key
sha256ecdsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A37

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_RSA_WITH_AES_256_CBC_SHA256  -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A38

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

   e.g.

```
$cd  /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 –msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_RSA_WITH_AES_256_CBC_SHA256","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A39

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_RSA_WITH_AES_256_GCM_SHA384 -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A40

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below e.g.

```
$cd  /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
nm_client_main("5","-tls1","-
cipher","TLS_RSA_WITH_AES_256_GCM_SHA384","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A41

On OPENSSL-LINUX -PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A42

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A43

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256rsa.pem", "-key", "
sha256rsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A44

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256rsa.pem -key
sha256rsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A45

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256ecdsa.pem", "-key", "
sha256ecdsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A46

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below e.g.

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256ecdsa.pem -key
sha256ecdsakey.pem -tls1 –msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A47

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

```
-> nm_server_main("5","-cert", "sha256ecdsa.pem", "-key", "
sha256ecdsakey.pem","-tls1")
```

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

```
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_client -tls1 -cipher
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384  -connect <host>:4433
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

CASE: A48

On OPENSSL-LINUX -PC:

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

```
e.g.
$cd /home/user1/openssls/openssl-1.0.1o
$./apps/openssl s_server -accept 4433 -cert sha256ecdsa.pem -key
sha256ecdsakey.pem -tls1 -msg
```

On WRSSL-Board and picocom on OPENSSL-PC:

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

```
-> nm_client_main("5","-tls1","-
cipher","TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384","-
connect","<OPENSSL_PC_IPAddr>:4433")
```

Expected result:

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

/EOD