Enabling SSL in Oracle E-Business Suite Release 12

The most significant change for Secure Sockets Layer (SSL) support in E-Business Suite Release 12 is the use of the mod_ossl module for the Oracle HTTP Server. Like mod_ssl, the mod_ossl plug-in enables strong cryptography for Oracle HTTP Server. In contrast to the OpenSSL module, mod_ossl is based on the Oracle implementation of SSL, which supports SSL 3, and is based on Certicom and RSA Security technology.

In Release 12 SSL certificates will be managed by the Oracle Wallet Manager 10g, which will be accessible via the familiar OWM graphical user interface (GUI) or the new ORAPKI command line interface (CLI). Since Release 12 will be using the Forms Listener Servlet a separate certificate is no longer needed for Forms. Forms will share the same wallet as the Oracle HTTP Server.

Note: The use of the Forms Server Listener with ConnectMode=https is not supported. ConnectMode=https only works with JInitiator which includes the Oracle SSL libraries. Release 12 uses the Sun Java Plugin and if you need to use https for the forms communication layer, you must use the servlet architecture.

Secure Sockets Layer (SSL)

SSL is a technology that defines the essential functions of mutual authentication, data encryption, and data integrity for secure transactions. Exchange of data between the client and server in such secure transactions is said to use the Secure Sockets Layer (SSL).

SSL uses 2 types of Certificates:

1. User certificates
   These are Certificates issued to servers or users to prove their identity in a public key/private key exchange.
2. Trusted certificates
   These are Certificates representing entities whom you trust - such as certificate authorities who sign the user certificates they issue.

How SSL works with Middle Tier Oracle HTTP Server:

1. The client sends a request to the server using HTTPS connection mode.
2. The server presents its certificate to the client. This certificate contains the server's identifying information.
3. The client checks its list of Trust points and compares the information in the certificate with the server's public key. If it matches, the server is authenticated as a trusted server.
4. The client sends the server a list of the encryption levels, or ciphers, that it can use.
5. The server receives the list and selects the strongest level of encryption that they have in common.
6. The client creates a session key which is used to encrypt the data and sends this session key to the server which can decrypt the data with its private key

How SSL works with Oracle Database Server:

1. The UTL_HTTP package is used for making HTTP callouts from SQL and PL/SQL to a Web Node (Oracle HTTP server).
2. When the package fetches data from a Web site using HTTPS, it specifies the location to the Oracle Wallet that resides on the database server. This wallet contains the certificate for the Certifying Authority (CA) who signed the Web node's server certificate.

Certificate Authority (CA)

A Certificate Authority is a trusted third party responsible for issuing, revoking, and renewing digital certificates. All digital certificates are signed with the Certificate Authority's private key to ensure authenticity. The Certificate Authority's Public Key is widely distributed.

Certificate Signing Request (CSR)

A Certificate Signing Request (CSR) is a digital file which contains your public key and your name. You send the CSR to a Certifying Authority (CA) to be converted into a real Certificate.

Digital Certificate (Public Key)

A digital certificate is an electronic document that binds an identity to a pair of electronic keys that can be used to encrypt and sign digital information. Certificates are issued by a trusted third party, called a Certification Authority (CA). The document is usually in a standard X509 format and contains three elements:

1. Entity attributes (information about your organization)
2. Public key (which is bound to your organization)
3. Digital signature of the trusted CA private key

Verisign (http://verisign.com/) will allow your organization to apply for a free trial certificate which will be valid for 2 weeks for testing purposes.

Private (Server) Key

The private key file is a digital file that you generate and for use to decrypt messages sent to you. The certificate request (CSR) that you send to your Certificate Authority (CA) is derived from this private key. Therefore, the resulting digital certificate (containing your public key) which is issued by your CA, is bound to this private key.

Secure Server Certificates

Secure Server Certificates are 128 bit certificates which provide 128 bit SSL encryption. If a browser has 128 bit support, then encryption is negotiated to 128 bits. However, if the browser only supports 40 bit encryption, the level of encryption, regardless of a 128 bit certificate, will be negotiated down to 40 bits.

Global Server Certificates

Global Server Certificates, also referred to as Server Gated Cryptography, are 128 bit certificates that enable all browsers to use 128 bit encryption, even if the browser only supports 40 bit encryption. A global server certificate usually has 2 parts: the certificate itself and an extra intermediate certificate which is used to provide the step-up. The marketing names of these certificates vary depending on the company that issues the certificate, for example, Thawte calls them 128 bit SuperCerts. It is not possible to get trial versions of global server certificates; therefore it is not possible to test unless one is purchased.

Secure Socket Layer Accelerators

Secure Socket Layer (SSL) Accelerators can be used to reduce the SSL traffic and workload off the web servers. Usually SSL accelerators are the primary targets for https requests from the user's desktop and thus are the initial target for all desktop client communication. They are responsible for converting "https" SSL requests to non-SSL "http" requests, directing the request to the http server which is running in non-SSL mode. Before sending the response back to the desktop they again convert the non-SSL requests to SSL requests.

Middle tier setup

The default location for the wallet in Release 12 is $INST_TOP/certs/Apache. This directory contains a wallet with demo certificates. If you wish to use these certificates for testing start with "Step -8" below to configure SSL, and then do Steps 1 through 7 when you are ready to switch to real certificates.

The demo certificates are not secure and should never be used in a production environment.

The main steps for setting up SSL on the Middle Tier are:

1. Set Your Environment.
2. Create a wallet.
3. Create a Certificate Request.
4. Submit the Certificate Request to a Certifying Authority.
5. Import your Server Certificate to the Wallet.
6. Copy the Apache Wallet to the OPMN Wallet.
7. Update the JDK Cacerts File.
8. Update the Context File.
9. Run Autoconfig.
10. Restart the middle tier services.

Step 1- Set Your Environment

- Logon to the application middle tier as the OS user who owns the middle tier files.
- Source your middle tier environment file (APPS<sid_machine>.env) located in the APPL_TOP directory.
- Navigate to the $INST_TOP/ora/10.1.3 and source the <sid_machine>.env file to set your
  10.1.3 ORACLE_HOME variables.

cd $INST_TOP/ora/10.1.3

. ./OBIEE_linux1.env

echo $ORACLE_HOME

/d04/oracle/OBIEE/apps/tech_st/10.1.3

Step 2 - Create a wallet

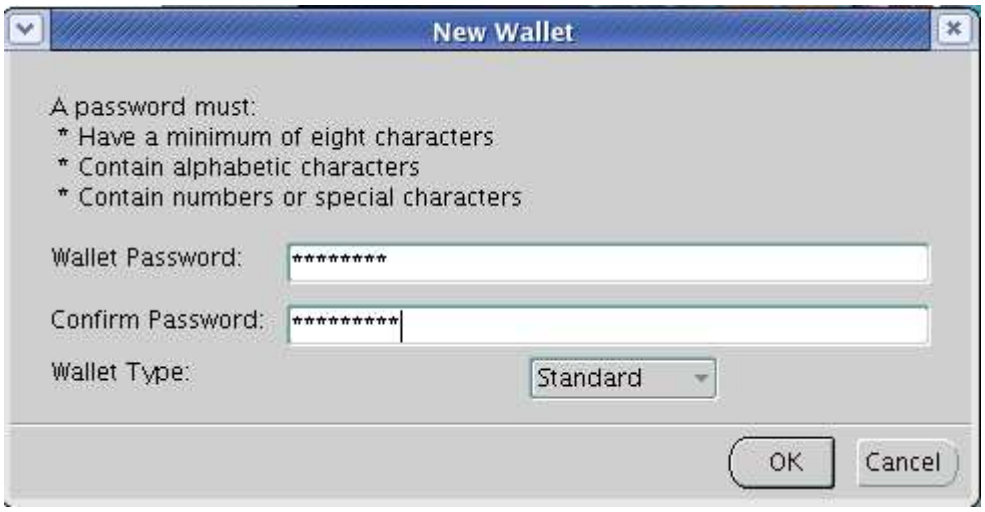
- Navigate to the $INST_TOP/certs/Apache directory.
- Move the existing wallet files to a backup directory in case you wish to use them again in the future.

cd $INST_TOP/certs/

cp -R Apache Apache_bak

- Open the Wallet manager as a background process:
  owm &

- On the Oracle Wallet Manager Menu navigate to Wallet -> New.
  Answer NO to: "Your default wallet directory doesn't exist. Do you wish to create it now?"



- The new wallet screen will now prompt you to enter a password for your wallet.

  Click YES when prompted:
  "A new empty wallet has been created. Do you wish to create a certificate request at this time?"

Step 3 - Create a Certificate Request

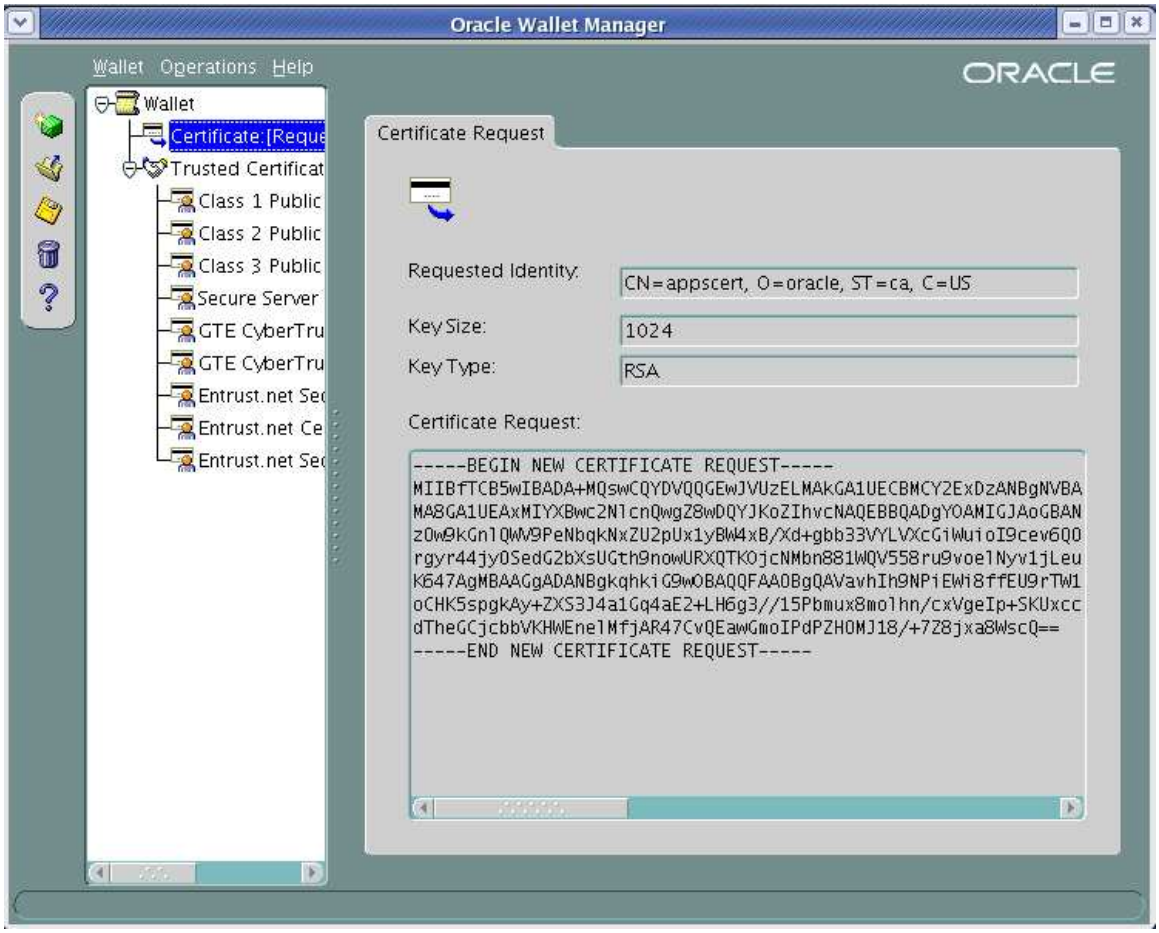
After clicking "Yes" in step 2 the Create Certificate Request Screen will pop up:

Fill in the appropriate values where:

Common Name: is the name of your server including the domain.
Organizational Unit: (optional) The unit within your organization.
Organization: is the name of your organization.

Locality/City: is your locality or city.
State/Province: is the full name of your State or Province - do not abbreviate.

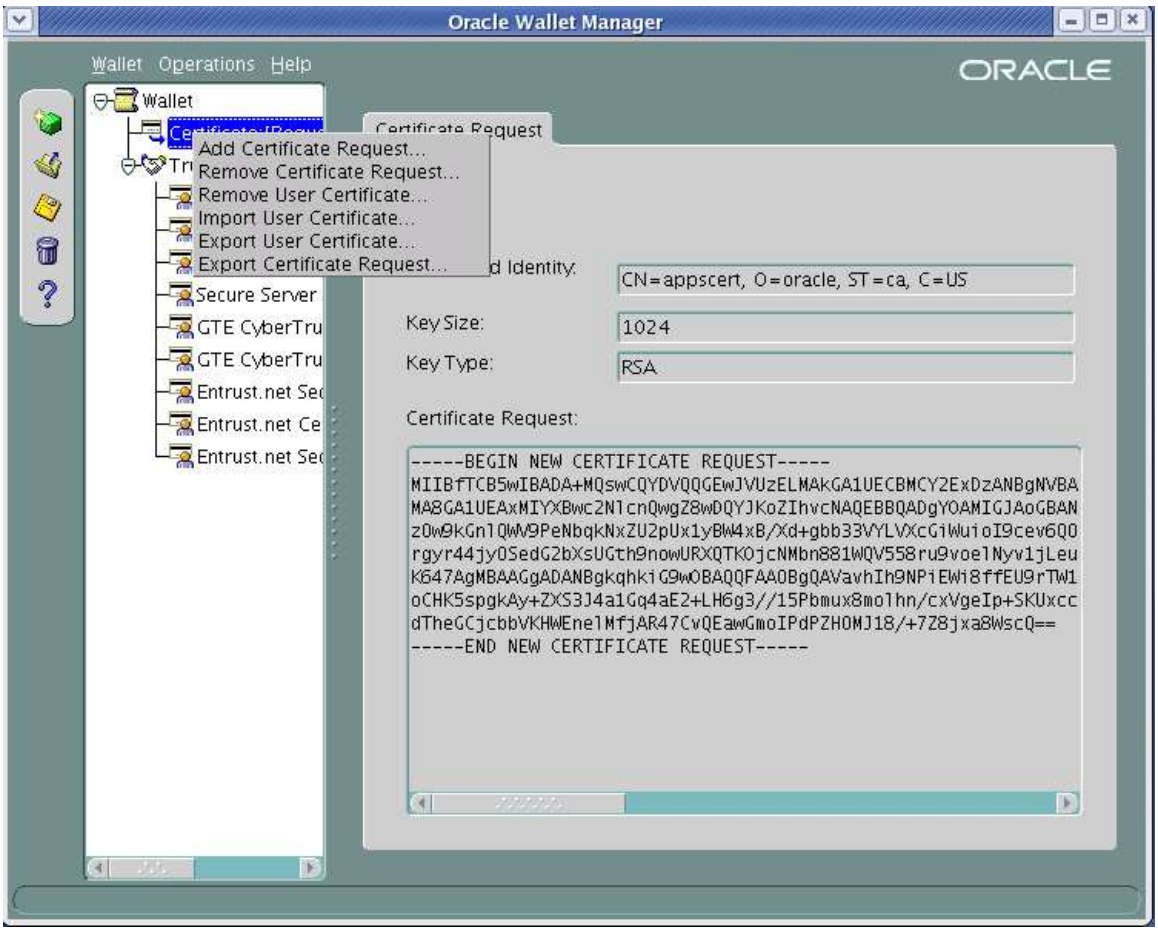
Select your Country from the drop down list.
Click OK.



Step 4 - Submit the Certificate Request to a Certifying Authority

You will need to export the Certificate Request before you can submit it to a Certifying Authority. To do so:

- Click on Certificate [Requested] to Highlight it.



- From the menu click Operations -> Export Certificate Request
- Save the file as server.csr

Press OK once you fill all information. Save wallet in $INST_TOP/certs/Apache folder with name (server_name).cer

Once saved, it will ask, “A wallet already exist, you want to override”. You can say yes. Click on certificate Request in the list provided on left side of Oracle wallet manager window and you can see your request contents on right side as seen in below screen.

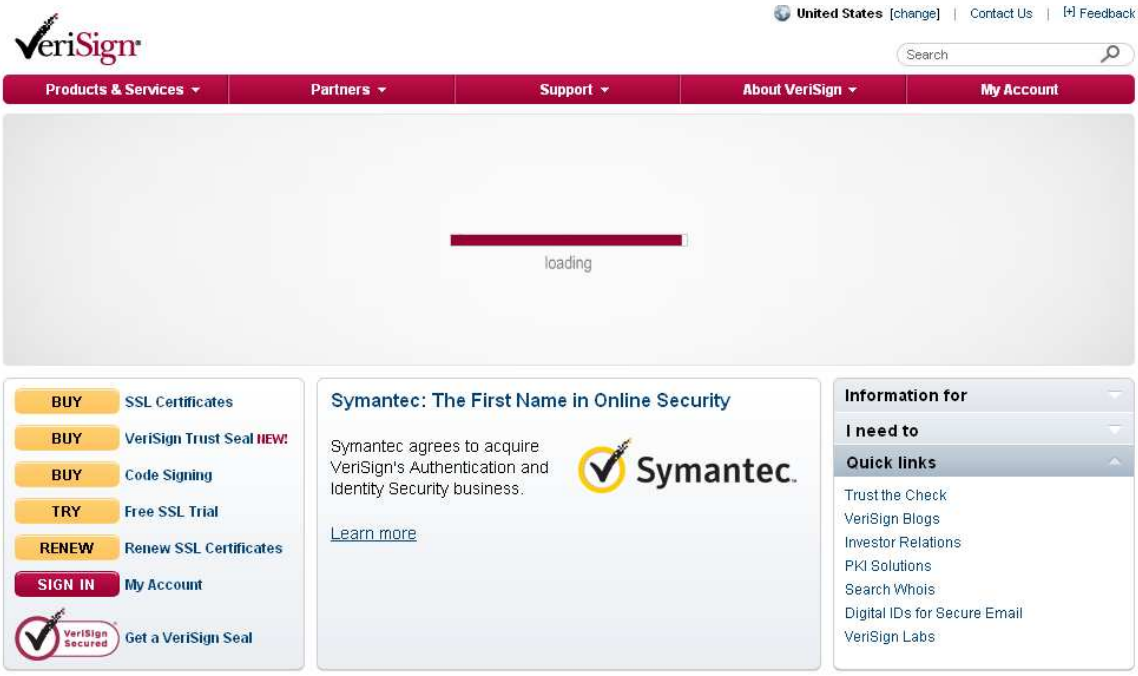
The wallet directory will now contain the following files:

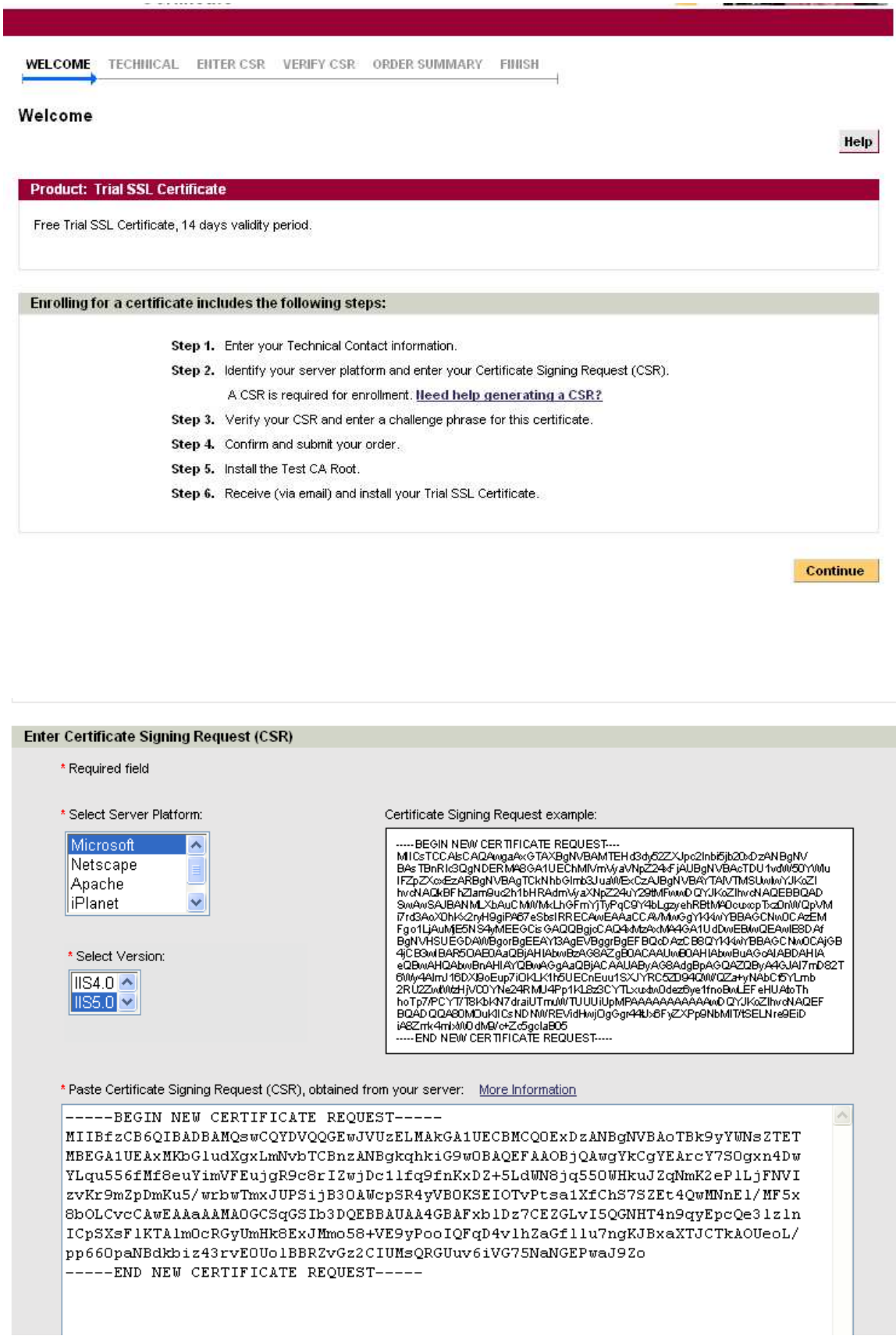
cwallet.sso
ewallet.p12
server.csr

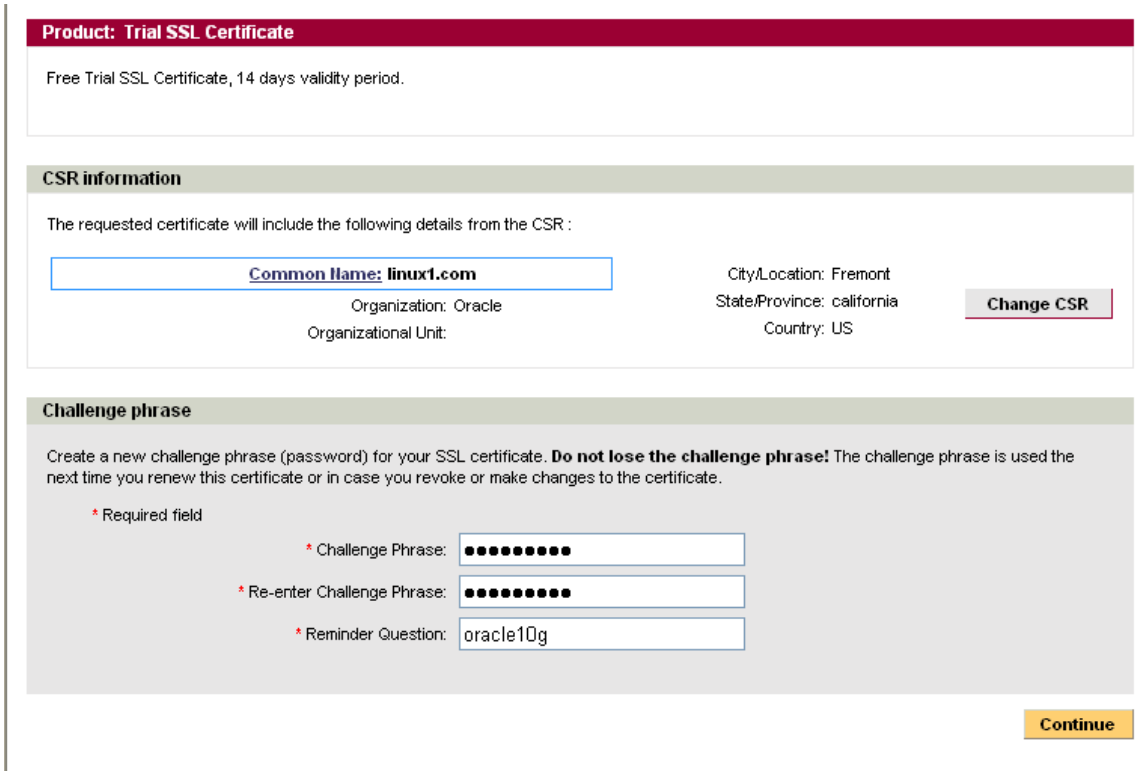
```
-rw-------  1 applmgr dba     601 Jun  4 23:04 server.csr
-rw-------  1 applmgr dba    9085 Jun  4 23:07 ewallet.p12
-rw-------  1 applmgr dba    9113 Jun  4 23:07 cwallet.sso
```

You may now submit server.csr to your Certifying Authority to request a Server Certificate.

http://www.verisign.com/

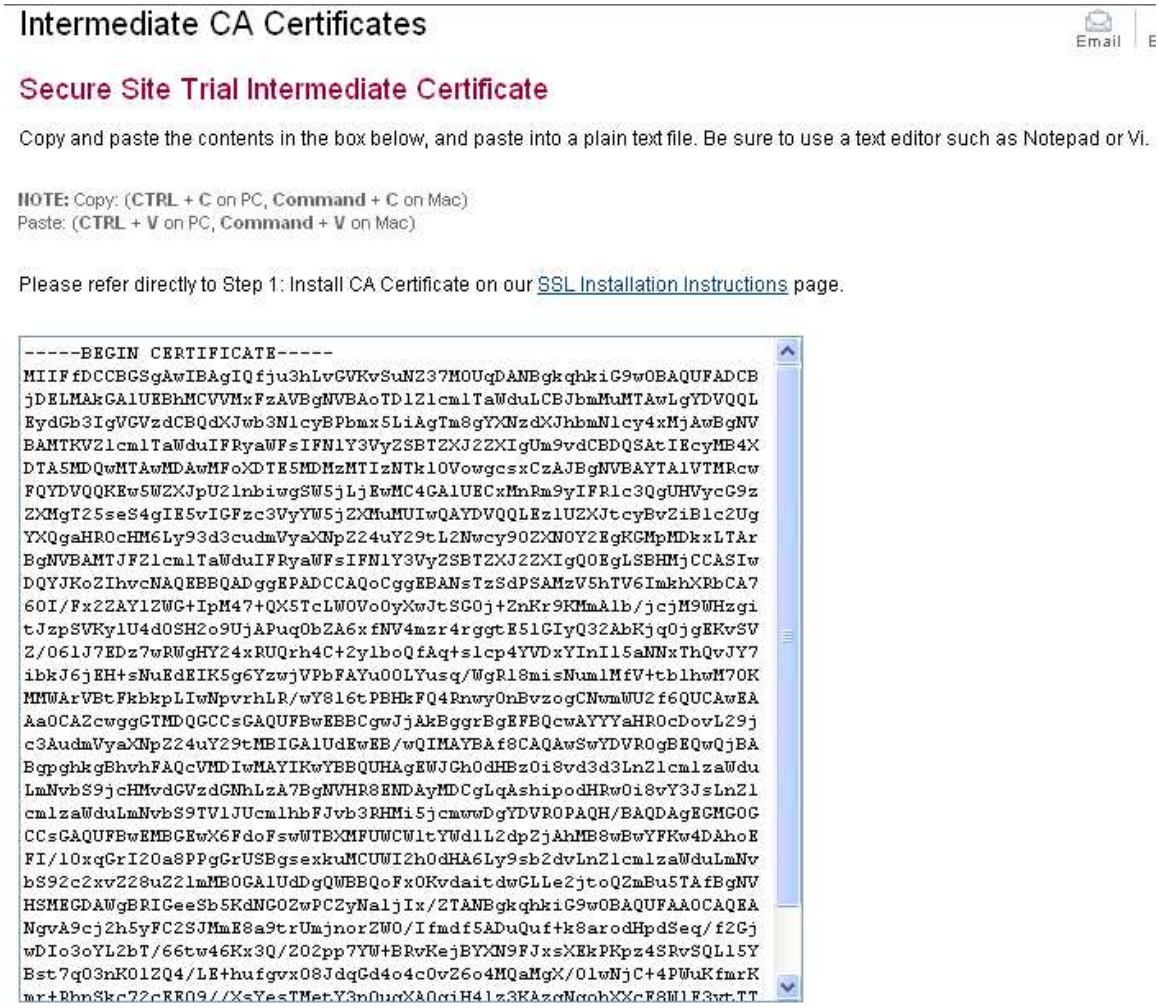



WELCOME
TECHNICAL
ENTER CSR
VERIFY CSR
ORDER SUMMARY
FINISH
Welcome
Help
Product: Trial SSL Certificate
Free Trial SSL Certificate, 14 days validity period.
Enrolling for a certificate includes the following steps:
Step 1. Enter your Technical Contact information.
Step 2. Identify your server platform and enter your Certificate Signing Request (CSR).
A CSR is required for enrollment. Need help generating a CSR?
Step 3. Verify your CSR and enter a challenge phrase for this certificate.
Step 4. Confirm and submit your order.
Step 5. Install the Test CA Root.
Step 6. Receive (via email) and install your Trial SSL Certificate.
Continue
Enter Certificate Signing Request (CSR)
* Required field
* Select Server Platform:
Microsoft
Netscape
Apache
iPlanet
* Select Version:
IIS4.0
IIS5.0
Certificate Signing Request example:
-----BEGIN NEW CERTIFICATE REQUEST----
-----END NEW CERTIFICATE REQUEST-----
* Paste Certificate Signing Request (CSR), obtained from your server: More Information
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBfzCB6QIBADBAMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExDzANBgNVBAoTBk9yYWNsZTET
MBEGA1UEAxMKbGludXgxLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEArcY7SOgxn4Dw
YLqu556fMf8euYimVFEujgR9c8rIZwjDc1lfq9fnKxDZ+5LdWN8jq55OWHkuJZqNmK2ePlLjFNVI
zvKr9mZpDmKu5/wrbwTmxJUPSijB3OAWcpSR4yVBOKSEIOTvPtsa1XfChS7SZEt4QwMNnEl/MF5x
8bOLCvcCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBAFxblDz7CEZGLvI5QGNHT4n9qyEpcQe3lzln
ICpSXsFlKTAlmOcRGyUmHk8ExJMmo58+VE9yPooIQFqD4vlhZaGfllu7ngKJBxaXTJCTkAOUeoL/
pp66OpaNBdkbiz43rvEOUolBBRZvGz2CIUMsQRGUuv6iVG75NaNGEPwaJ9Zo
-----END NEW CERTIFICATE REQUEST-----



Step 2. Download the Trial SSL Intermediate CA Certificate.

To download the Trial Intermediate CA on each Web server you are testing with, go to:
http://www.verisign.com/support/verisign-intermediate-ca/trial-secure-server-intermediate/index.html



Intermediate CA Certificates

Email

Secure Site Trial Intermediate Certificate

Copy and paste the contents in the box below, and paste into a plain text file. Be sure to use a text editor such as Notepad or Vi.

NOTE: Copy: (**CTRL** + **C** on PC, **Command** + **C** on Mac)
Paste: (**CTRL** + **V** on PC, **Command** + **V** on Mac)

Please refer directly to Step 1: Install CA Certificate on our SSL Installation Instructions page.

```
-----BEGIN CERTIFICATE-----
MIIFfDCCBGSgAwIBAgIQfju3hLvGVKvSuNZ37MOUqDANBgkqhkiG9w0BAQUFADCB
jDELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTAwLgYDVQQL
EydGb3IgVGVzdCBQdXJwb3NlcyBPbmx5LiAgTm8gYXNzdXJhbmNlcy4xMjAwBgNV
BAMTKVZlcmlTaWduIFRyaWFsIFNlY3VyZSBTZXJ2ZXIgUm9vdCBDQSAtIEcyMB4X
DTA5MDQwMTAwMDAwMFoXDTE5MDMzMTIzNTk1OVowgcsxCzAJBgNVBAYTAlVTMRcw
FQYDVQQKEw5WZXJpU2lnbiwgSW5jLjEwMC4GA1UECxMnRm9yIFRlc3QgUHVycG9z
ZXMgT25seS4gIE5vIGFzc3VyYW5jZXMuMUIwQAYDVQQLEzlUZXJtcyBvZiBlc2Ug
YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL2Nwcy90ZXN0Y2EgKGMpMDkxLTAr
BgNVBAMTJFZlcmlTaWduIFRyaWFsIFNlY3VyZSBTZXJ2ZXIgQ0EgLSBHMjCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANsTzSdPSAMzV5hTV6ImkhXRbCA7
6OI/Fx2ZAYlZWG+IpM47+QX5TcLWOVo0yXwJtSGOj+ZnKr9KMmAlb/jcjM9WHzgi
tJzpSVKylU4dOSH2o9UjAPuqObZA6xfNV4mzr4rggtE5lGIyQ32AbKjqOjgEKvSV
Z/O6lJ7EDz7wRWgHY24xRUQrh4C+2ylboQfAq+slcp4YVDxYInIl5aNNxThQvJY7
ibkJ6jEH+sNuEdEIK5g6YzwjVPbFAYu0OLYusq/WgRl8misNumlMfV+tblhwM7OK
MMWArVBtFkbkpLIwNpvrhLR/wY816tPBHkFQ4RnwyOnBvzogCNwmWU2f6QUCAwEA
AaOCAZcwggGTMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYYYaHR0cDovL29j
c3AudmVyaXNpZ24uY29tMBIGA1UdEwEB/wQIMAYBAf8CAQAwSwYDVR0gBEQwQjBA
BgpghkgBhvhFAQcVMDIwMAYIKwYBBQUHAgEWJGh0dHBzOi8vd3d3LnZlcmlzaWdu
LmNvbS9jcHMvdGVzdGNhLzA7BgNVHR8ENDAyMDCgLqAshipodHRwOi8vY3JsLnZl
cmlzaWduLmNvbS9TVlJUcmlhbFJvb3RHMi5jcmwwDgYDVR0PAQH/BAQDAgEGMG0G
CCsGAQUFBwEMBGEwX6FdoFswWTBXMFUWCWltYWdlL2dpZjAhMB8wBwYFKw4DAhoE
FI/l0xqGrI2Oa8PPgGrUSBgsexkuMCUWI2h0dHA6Ly9sb2dvLnZlcmlzaWduLmNv
bS92c2xvZ28uZ2lmMB0GA1UdDgQWBBQoFx0KvdaitdwGLLe2jtoQZmBu5TAfBgNV
HSMEGDAWgBRIGeeSb5KdNGOZwPCZyNaljIx/ZTANBgkqhkiG9w0BAQUFAAOCAQEA
NgvA9cj2h5yFC2SJMmE8a9trUmjnorZWO/Ifmdf5ADuQuf+k8arodHpdSeq/f2Gj
wDIo3oYL2bT/66tw46Kx3Q/ZO2pp7YW+BRvKejBYXN9FJxsXEkPKpz4SRvSQL15Y
Bst7q03nKOlZQ4/LE+hufgvx08JdqGd4o4c0vZ6o4MQaMgX/OlwNjC+4PWuKfmrK
mr+RhnSkc72cRE09//XsYesTMetY3nOuqXA0giH4lz3KAzqNqobXXcE8WlE3xtTT
```

Creating your certifying authority's certificate

- Copy/ftp (binary mode) .crt to the your PC desktop
- Double-click the file and go to Certification Path tab
- Double click on VeriSign/RSA Secure Server CA
- Go to Details tab and press Copy to File…
- Press Next and select "Base-64 encoded X.509(.CER)" and press next
- Give the name as server.crt
- Press Finish

A new server.crt file will be present in your local PC. FTP back ca.cer file to your UNIX host.

Step 5 - Import your Server Certificate to the Wallet.

After you receive your Server Certificate from your Certifying Authority you will need to import it into your wallet. Copy the certificate to server.crt in the wallet directory on your server by one of the following methods:

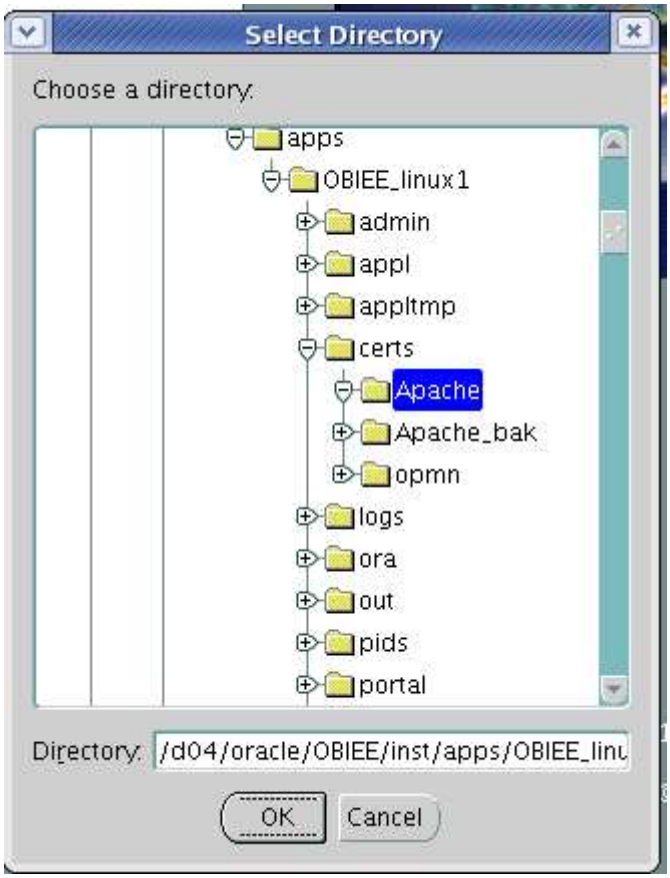
1. ftp the certificate (in binary mode)
2. copy and paste the contents into server.crt

Follow these steps to import server.crt into your wallet:

- Open the Wallet Manager as a background process:
  owm &

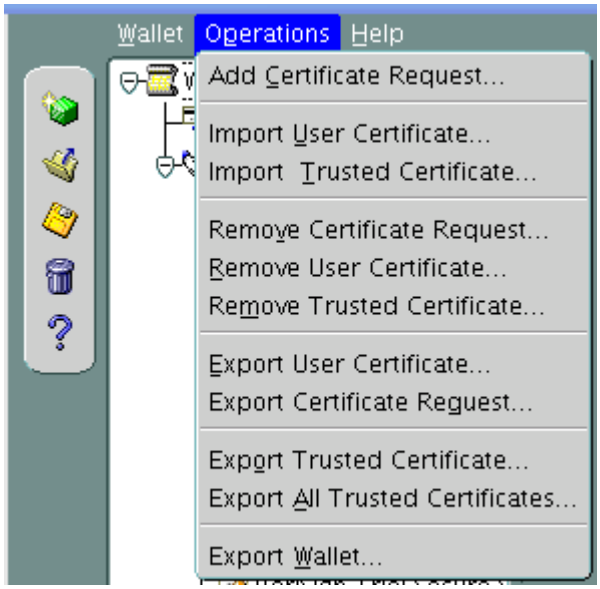
- From the menu click Wallet then Open.
- Answer Yes when prompted:
  Your default wallet directory does not exist.
  Do you want to continue?

- On the Select Directory screen change the Directory to your fully qualified wallet directory and click OK

- 

- Enter your wallet password and click OK.
- 

- 



Operations – Import Trusted Certificate

- On the Oracle Wallet Manager Menu navigate to Operations - Import User Certificate. Server certificates are a type of user certificate. Since the Certifying Authority issued a certificate for the server, placing its distinguished name (DN) in the Subject field, the server is the certificate owner, thus the "user" for this user certificate.

- Click OK.
- Double Click on server.crt to import it.
- Save the wallet:
  On the Oracle Wallet Manager Menu click Wallet.
  Verify the Auto Login box is checked.
  Click Save.

Step 6 - Modify the OPMN wallet.

The Oracle Applications Rapid Install process creates a default "demo" opmn wallet in the $INST_TOP/certs/opmn directory that can be used in test instances for basic SSL testing. Now that the Apache wallet has been created you will need to to use these same certificates for opmn. Use the following steps to backup and copy the wallets:

- Navigate to the $INST_TOP/certs/opmn directory.
- Create a new directory named BAK
- Move the ewallet.p12 and cwallet.sso files to the BAK directory just created.
- Copy the ewallet.p12 and cwallet.sso files from the $INST_TOP/certs/Apache directory to the $INST_TOP/certs/opmn directory.

Step 7 - Update the JDK Cacerts File.

Oracle Web Services requires the Certificate of the Certifying Authority who issued your server certificate (ca.crt from the previous step) to be present in the JDK cacerts file. In addition, some features of XML Publisher and BI Publisher require the server certficate (server.crt from previous step) to be present, Follow these steps to be sure these requirements are met:

- Navigate to the $OA_JRE_TOP/lib/security directory
- Backup the existing cacerts file.
- Copy your ca.crt and server.crt files to this directory
  Issue the following command to insure that cacerts has write permissions:

  chmod u+w cacerts

- Add your Apache ca.crt and server.crt to cacerts:

keytool -import -alias ApacheRootCA -file ca.crt -trustcacerts -v -keystore cacerts
keytool -import -alias ApacheServer -file server.crt -trustcacerts -v -keystore cacerts

When prompted enter the keystore password (default password is changeit).

Step 8 - Update the Context File.

Use the Oracle Applications Manager (OAM) Context Editor to change the SSL related variables as shown in this table:

SSL Related Variables in the Context File

Variable	Non-SSL Value	SSL Value
s_url_protocol	http	https
s_local_url_protocol	http	https
s_webentryurlprotocol	http	https
s_active_webport	same as s_webport	same as s_webssl_port
s_webssl_port	not applicable	default is 4443
s_https_listen_parameter	not applicable	same as s_webssl_port
s_help_web_agent	url constructed with http protocol and s_webport	url constructed with https protocol and s_webssl_port
s_login_page	url constructed with http protocol and s_webport	url constructed with https protocol and s_webssl_port
s_external_url	url constructed with http protocol and s_webport	url constructed with https protocol and s_webssl_port

Changes when using an SSL Accelerator

Variable	Non-SSL Value	SSL Value
s_url_protocol	http	http
s_local_url_protocol	http	http
s_webentryurlprotocol	http	https
s_active_webport	same as s_webport	value of the SSL Accelerator's external interfacing port
s_webentryhost	same as s_webhost	SSL Accelerator hostname
s_webentrydomain	same as s_domainname	SSL Accelerator domain name
s_enable_sslterminator	#	remove the '#' to use ssl_terminator.conf in ssl terminated environments
s_login_page	url constructed with http protocol and s_webport	url constructed with https protocol, s_webentryhost, s_webentrydomain, s_active_webport
s_external_url	url constructed with http protocol and s_webport	url constructed with https protocol, s_webentryhost, s_webentrydomain, s_active_webport

s_url_protocol - https

s_local_url_protocol - https

s_webentryurlprotocol - https

s_active_webport - 443

s_webssl_port - 443

s_https_listen_parameter - 443

s_login_page - https://linux1.com:443/OA_HTML/AppsLogin

s_external_url - https://linux1.com:443

Step 9 - Run Autoconfig

Autoconfig can be run by using the adautocfg.sh script in the Middle Tier $ADMIN_SCRIPTS_HOME directory.

Error Faced – Apache is not starting

sh adapcctl.sh start

You are running adapcctl.sh version 120.7.12010000.2

Starting OPMN managed Oracle HTTP Server (OHS) instance ...

adapcctl.sh: exiting with status 204

adapcctl.sh: check the logfile /d04/oracle/OBIEE/inst/apps/OBIEE_linux1/logs/appl/admin/log/adapcctl.txt for more information

cat /d04/oracle/OBIEE/inst/apps/OBIEE_linux1/logs/appl/admin/log/adapcctl.txt

06/05/10-23:47:06 :: adapcctl.sh: starting OPMN managed OHS instance

opmnctl: starting opmn managed processes...

==========
opmn id=linux1.com:6210

0 of 1 processes started.

ias-instance id=OBIEE_linux1.linux1.com

ias-component/process-type/process-set:

HTTP_Server/HTTP_Server/HTTP_Server/

Error

--> Process (index=1,uid=452071893,pid=3619)

failed to start a managed process after the maximum retry limit

Log:

/d04/oracle/OBIEE/inst/apps/OBIEE_linux1/logs/ora/10.1.3/opmn/HTTP_Server~1.log

06/05/10-23:47:12 :: adapcctl.sh: exiting with status 204

$LOG_HOME/appl/admin/log/adapcctl.txt

cat /d04/oracle/OBIEE/inst/apps/OBIEE_linux1/logs/ora/10.1.3/opmn/HTTP_Server~1.log

cd $LOG_HOME/ora/10.1.3/Apache/error_log.[number] & access_log

[Sat Jun 5 23:47:11 2010] [crit] (13)Permission denied: make_sock: could not bind to port 443

Need to fix it

Step 10 - Customizations (optional)

In Release 12 we keep a non-ssl port open for those products which need to access some of their pages via the http protocol, as well as the Oracle Applications Help System. If you wish to disable the http port and force all users to access your pages via the https protocol you can add a redirect rule to $INST_TOP/ora/10.1.3/Apache/Apache/conf/custom.conf file.
RewriteRule ^/$ https://<servername.domain:<port>/OA_HTML/AppsLogin [R,L]:

RewriteRule ^/$ https://<servername.domain:<port>/OA_HTML/AppsLogin [R,L]

Any updates you make to the custom.conf file will be preserved when Autoconfig is run.

Step 11 - Restart the middle tier services.

Use the adapcctl.sh script in the $ADMIN_SCRIPTS_HOME directory to stop and restart the middle tier Apache services.

Section 4: Database Tier Setup

Oracle products such as Oracle Configurator, Order Management, iStore, Order Capture, Quoting, iPayment, iStore, and Pricing access data over the Internet in HTTP or HTTPS connection mode. The implementation of SSL for the Oracle Database Server (which acts as a client sending requests to the Web server) makes use of the Oracle Wallet Manager for setting up an Oracle wallet.

To enable SSL on the Database Tier you need only create a wallet. You do not need a server certificate for this wallet. If you were required to import your ca.crt into the middle tier wallet you will need to do it for this wallet also.

- After setting your environment for the database tier, navigate to the $ORACLE_HOME/appsutil directory.
- Create a new wallet directory named: *wallet*
- Navigate to the newly created wallet directory.
- Open the Wallet Manager as a background process:
  owm &
- On the Oracle Wallet Manager Menu navigate to Wallet -> New.
  Answer NO to: "Your default wallet directory doesn't exist. Do you wish to create it now?"
  The new wallet screen will now prompt you to enter a password for your wallet.

  Click NO when prompted:
  "A new empty wallet has been created. Do you wish to create a certificate request at this time?"
- If you need to import ca.crt:
  On the Oracle Wallet Manager menu navigate to Operations -> Import Trusted Certificate.
  Click OK.
  Double click on ca.crt to import it.
- Save the wallet:
  On the Oracle Wallet Manager Menu click Wallet.
  Verify the Auto Login box is checked.
  Click Save.

To test that the wallet is properly set up and accessible, login to SQLPLUS as the apps user and execute the following:

```
select utl_http.request('[address to access]', '[proxy address]',
'file:[full path to wallet directory]', null) from dual;
```

where:

'[address to access]' = the url for your Oracle Applications Rapid Install Portal.

'[proxy address]' = the url of your proxy server, or NULL if not using a proxy server.

'file:[full path to wallet directory]' = the location of your wallet directory.

The final parameter is the wallet password, which is set to null by default.

References:

http://avdeo.com/enabling-ssl-in-oracle-e-business-suite-release-12/
Metalink Note -ID **376700.1** - Enabling SSL in Oracle Applications Release 12
Note 376694.1: Using the Oracle Wallet Manager Command Line Interface in Release 12.
http://kannankumara.blogspot.com/2008/02/self-signed-certificate-for-oracle-as.html
http://tylermuth.wordpress.com/2007/07/27/oracle-wallet-w-self-signed-certificate/