Install, Configuration and Developer Guide

iPlanet Web Server Plug-in for Trustbase services

Version 1.0

October 2001

Introduction

The following chapter discusses all related documents to this guide.

Overall Layout

•

The manual Covers:

- Introduction
- Installing
- Configuring
- Developing using the API

Related Documents

- iPlanet‘ Web Server

http://docs.iplanet.com/docs/manuals/enterprise.html

- iPlanet‘ Certificate Management Server

http://docs.iplanet.com/docs/manuals/cms.html

- A SmartCard,

`http://www.gemplus.com/app/banking/gemsafe_is_mkt.htm`

- A SmartCard Reader

`http://www.gemplus.com/products/hardware/index.htm`

- GEMSafe Enterprise Workstation 1.0 software

`http://www.gemplus.com/products/software/gemsafe/index.html`

- The Hardware Security Module, CAFast

`http://www.ncipher.com`

- Identrus

http://www.identrus.com

  - Identrus Digital Signature Messaging Specification V2.0
  - Transaction Coodinator Certificate Status Check Protocol Definition
  - Identrus OCSP Responder Compliance DOcument V4.0

- RFC2560 Online certificate Status Check protocol

http://www.ietf.org/rfc/rfc2560.txt

- Access Control API

http://developer.netscape.com:80/docs/manuals/enterprise/accessapi/api.htm

- Java 2 Standard Edition (1.2.2_06 Localized) Production Release for the Solaris Operating Environment.

`http://www.sun.com/software/solaris/java/download.html`

Contents

List of Figures

Chapter 1

Introduction

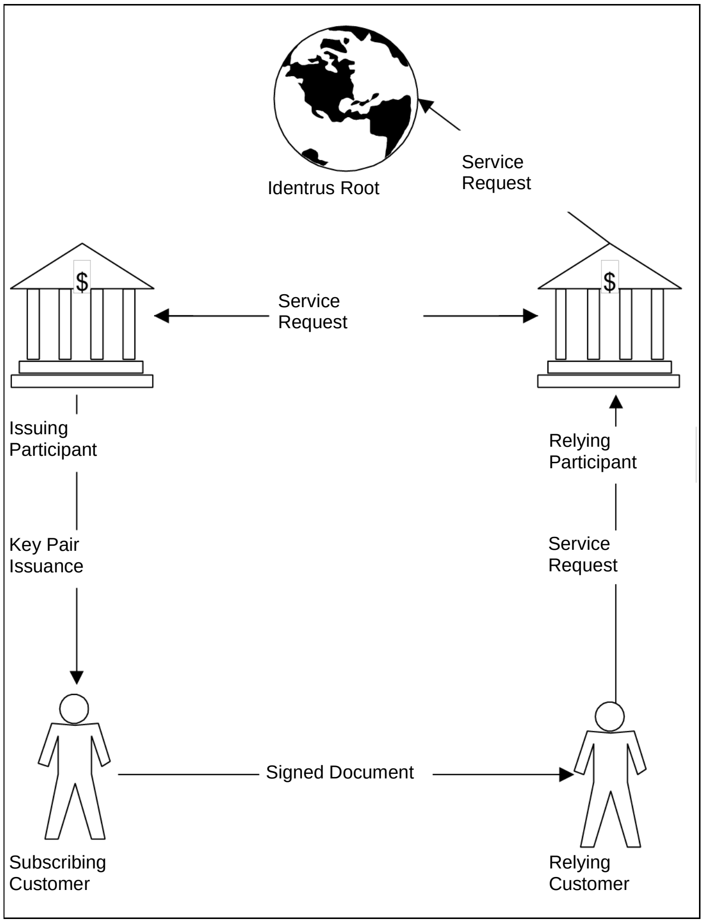
This chapter provides an overview of iPlanet Web Server: Plug-in for Trustbase services

About the Identrus Scheme

Identrus was formed to establish and operate a highly secure system for identifying parties over electronic networks, including the Internet. Identrus is comprised of regulated financial institutions coming together to combine the basic technology provided by public key cryptography and PKI with adherence to a common set of operating rules that facilitate electronic commerce.

The “Four Corner” model, as depicted below, forms the basis of the Identrus PKI network.

Figure 1-1 Identrus Four-Corner Model



The four corners consist of:

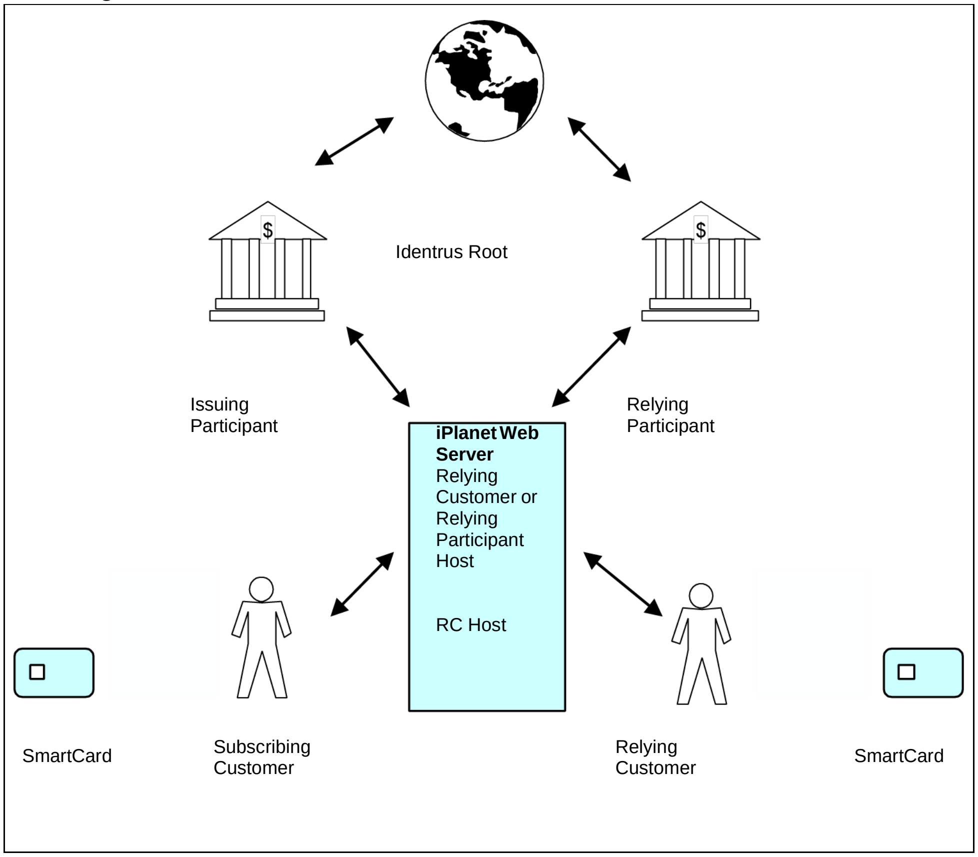
- Issuing Participant – Bank (or other financial institution) issuing smart cards containing private keys and certificates to Subscribing Customers.
- Subscribing Customer – Member of the Issuing Participant bank authorised to participate in Identrus activities.
- Relying Customer – Party with whom the Subscribing Customer initiates a signed transaction.
- Relying Participant – Bank with which the Relying Customer communicates to obtain some level of trust in the signed data received from the Subscribing Customer.

A typical transaction using the Four-Corner Model includes the following steps:

- The Subscribing Customer uses his / her SmartCard to sign a transaction request that the system sends to the Relying Customer.
- The Relying Customer verifies that the signed data came from the Subscribing Customer and was not modified in transit.
- The Relying Customer requests services from his / her bank; for example, he / she can ask its bank to perform a status check on the Subscribing Customer's certificate to make sure it is still valid (has not been revoked).
- The Relying Participant requests services of the Issuing Participant and the Identrus Root to fulfill the Relying Customer's service request.
- Based on the response from his / her bank the Relying Customer fulfills or rejects the request from the Subscribing Customer.

In order to accommodate the Identrus Scheme, the iPlanet Web Server: Plug-in for Trustbase services utilises a host system connected to the Web Server as illustrated below:

Figure 1-2 iPlanetWeb Server and the Identrus Scheme



iPlanet's Web Server Plug-in solution

The iPlanet Web Server: Plug-in for Trustbase services is designed for those Developers, Users and Administrators wishing to achieve reliable messaging when sending and receiving messages. Messages gain meaning if in some way they have:

- Authentication i.e. You know who the message came from
- Integrity i.e. You know the message has not changed
- Non-repudiation i.e. The sender cannot disavow the message

If and only if there exists both a valid digital signature and a verified certificate can we say that a message be non-repudiable. Without these mechanisms in place this cannot happen, and as such, the purpose of this document is to explain the main components associated with:

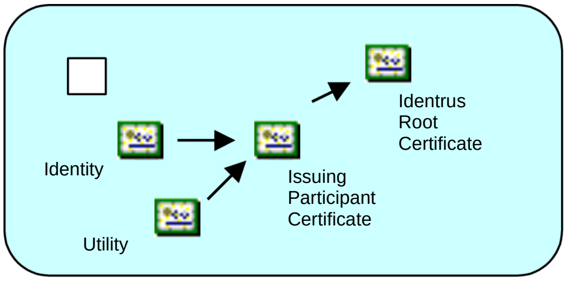
- Firstly configuring a system so that appropriate trusted certificate hierarchies are in place to authenticate messages via a certificate status check.
- Secondly how developers can deploy their own applications that are Identrus enabled

The iPlanet Web Server: Plug-in for Trustbase services achieves these aims by allowing the user the ability to determine the status of each message received and also to send messages with integrity if they so wish.

Inside an Identrus SmartCard

Part of setting up the iPlanet Web Server: Plug-in for Trustbase services involves issuing users with a SmartCard. Every Smart Card contains certificates that has an Identrus Hierarchy that defines which users can logon to the Web Server as illustrated below.

Figure 1-3 A SmartCard illustrating its Certificate Hierarchy



If a chain of trust can be built between a specific SmartCard certificate and a trusted CA, the certificate can be trusted (by inference). In such circumstances, any user that is an Identrus member can log onto the Web Server, provided the administratorhas constructed an entry in the webserver's LDAP user database and the correct certificate hierarchy is in place.

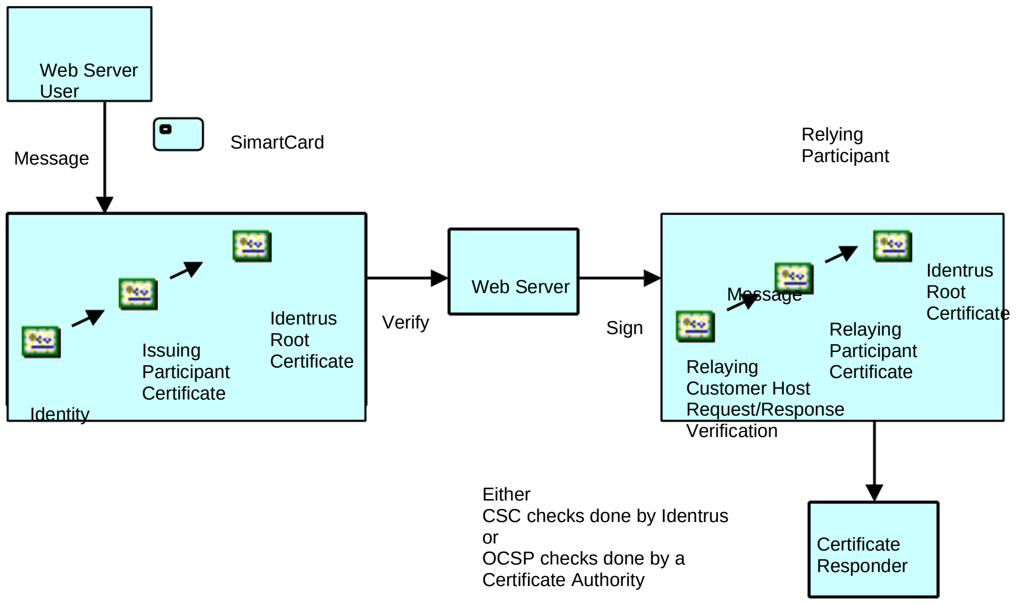
In order to become Identrus-enabled, the following certificates are needed on the Smartcard:

- An Identrus enabled Identity Certificate that provides information about who the user is.
- An Identrus enabled Utility Certificate that provides information about what access the user is authorised to have.
- An Identrus enabled Identity Certificate Authority Certificate that issues the users identity.
- An Identrus enabled Utility Certificate Authority Certificate that issues the users utility.
- Identrus Root Certificate

The Identrus Certificate Scheme

Administrators wishing to install the iPlanet Web Server: Plug-in for Trustbase services need to be aware of the main components that go into enabling users to achieve these aims. The diagram below illustrates an example of how messages typically might be validated and checked for integrity using certificates that have been issued by appropriate certificate Authorities within the Identrus Scheme.

Figure 1-4 Overview of Certificate Verification for the Identrus Scheme

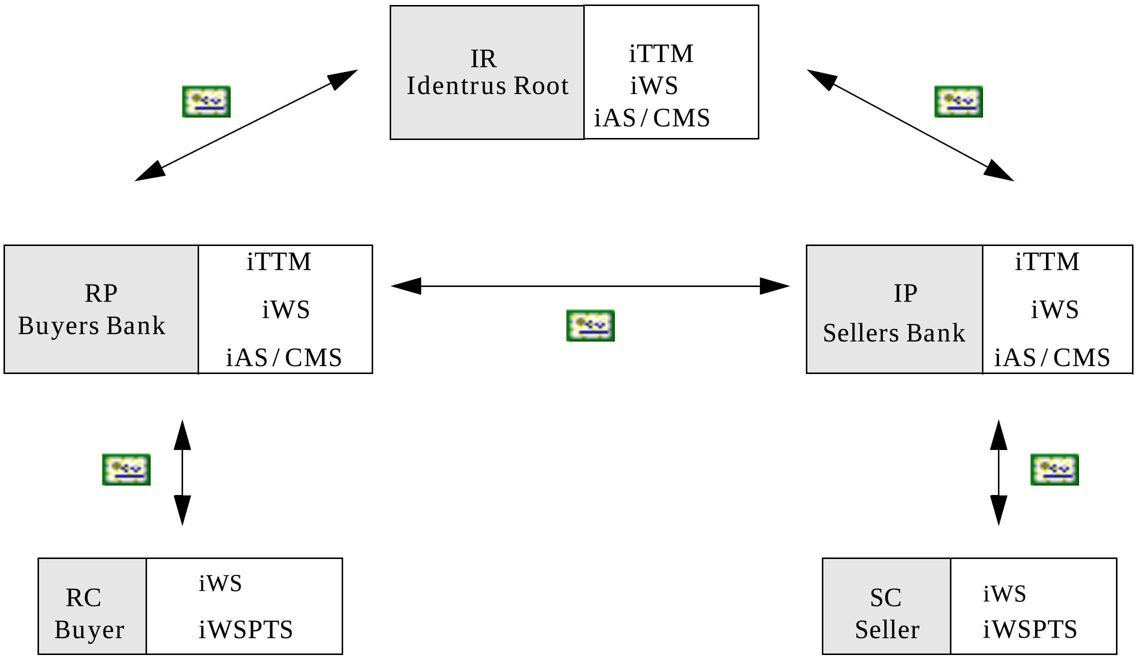

Web Server User
Message
SimartCard
Relying Participant
Identity
Issuing Participant Certificate
Identrus Root Certificate
Verify
Web Server
Sign
Message
Identrus Root Certificate
Relaying Participant Certificate
Relaying Customer Host Request/Response Verification
Either CSC checks done by Identrus or OCSP checks done by a Certificate Authority
Certificate Responder

❍

How iWSPTS fits in with iTTM

iPlanet Trustbase Transaction Manager (iTTM) processes Certificate Status checks from the Relying Participant (RP), Issuing participant (IP) and the Identrus Root (IR) They are initiated, however, by iPlanet Web Server: Plug-in for Trustbase services (iWSPTS) at the level of the Relying Customer(RC) or the Subscribing Customer (SC) as illustrated below

Figure 1-5 Initiating Certificate Status Checks



The following iPlanet products are utilised in this process

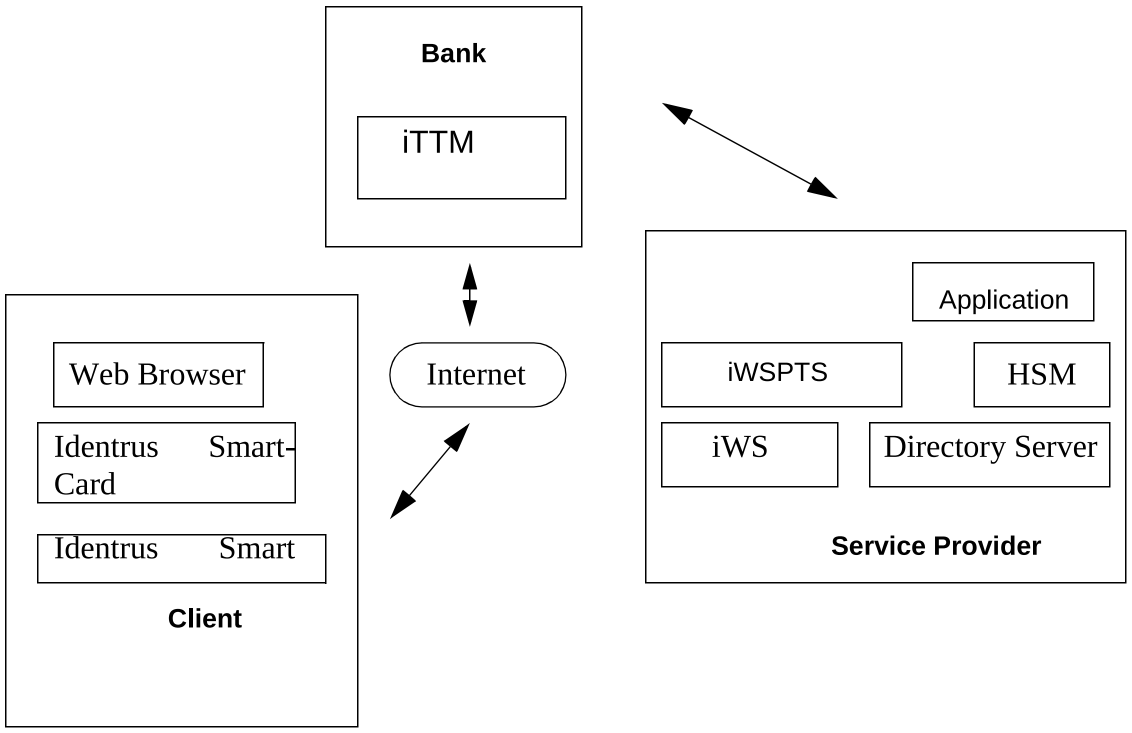
- iPlanet Web Server: Plug-in for Trustbase Services (iWSPTS)
- iPlanet Trustbase Transaction Manager (iTTM)
- iPlanet Web Server (iWS) and iPlanet Application Server (iAS)
- Certificate Management System (CMS)

iWSPTS Architecture Overview

The purpose of the IWS Plug-In is twofold:

- To provide a means of integrating Identrus CSC checks into merchant applications
- To provide users with a means of using the Identrus Smart Card as an authentication token

Figure 1-6 Archiectural Overview



The integration of CSC checks into merchant applications is provided as a Java API, and as such has no specific user interaction. The authentication process is as follows:

- The user loads the URL for the application or is redirected to it from a secured URL , and is presented with an HTML page containing a notice to place the smartcard into its reader for authentication and type in their application user name.
- The user asked to grant system access to the smart card interface applet.
- When using the Smart Card interface, the HTML page contains a signature tag in order to activate the Smart Card plug in. This presents the user with password / PIN dialogue request and the user enters the appropriate information. The user is then presented with a random string value and asked if they wish to sign the data. The user presses accept, and the form (along with the signed data is submitted)
- If the authentication fails the user is directed to an authentication failure page containing an appropriate message, the original URL value, and a link to the smartcard authentication page.

At the highest level the architectural model has three major components that must interact for the authentication system to work. A client who is normally the subject of the request, A bank that must respond to such a request and the Service Provider or Seller that normally initates the request in the first place.

Chapter 2

Installation

The plugin for the iplanet webserver comes in two pieces. The first piece of the plugin is termed the “Authentication Servlet”. This piece’s role is to perform the certificate status check that is the basis of the authentication scheme for the plugin. If the certificate status check is completed with satisfactory results a cookie is generated by the servlet and given to the browser. The second piece of the plugin is termed the “Cookie Authorisation”. The role of this piece is to check that cookie’s submitted as passports to secured area’s of the webserver are deemed “valid”. Thus the Authentication servlet performs the authentication task and the cookie authorisation performs the authorisation role. Finally there is a DSMS Java API library should you wish to integrate with existing systems. This document will detail how to install and configure both sections of the plugin.

Pre-requisites

The following Software and hardware peripherals must be installed prior to installing the iPlanet Web Server: Plug-in for Trustbase services:

- iPlanet‘ Web Server Enterprise Edition versions 4.1 and 6.0

  http://www.iplanet.com/products/iplanet_web_enterprise/

- iPlanet‘ Certificate Management Server (Optional) (See for instance `http://www.iplanet.com/downloads/download/2042.html`)

- A SmartCard, such as a credit card, which will be issued to you by a thirty party vendor. An Identrus compatible SmartCard is a mandatory requirement. iPlanet Web Server Plugin for the Identrus Network V2.0 is currently compatible with the GemPlus SmartCards GemSAFE IS 16000. See `http://www.gemplus.com/app/banking/gemsafe_is_mkt.htm`

- A SmartCard Reader with browser plug-in, which will be issued to you by a third party vendor. An Identrus compatible SmartCard Reader is a mandatory requirement. iPlanet Web Server Plugin for the Identrus Network V2.0 is currently compatible with the GemPlus Card Readers GemPC430 and GemPC410 see `http://www.gemplus.com/products/hardware/index.htm`

- GEMSafe Enterprise Workstation 1.0 software is compatible with iPlanet Web Server Plugin for the Identrus Network V2.0 see `http://www.gemplus.com/products/software/gemsafe/index.html`

- The Netscape Navigator v4.7x or above. Internet Explorer 4.0 and Internet Explorer 5.0. These should be configured automatically with the software that comes with your SmartCard and SmartCard Reader.

- The Client browser is compatible with GemPlus SmartCard and the Web Server Plug-in as such the following operating systems are supported: Windows NT 4.0 Service Pack 5 see `http://www.microsoft.com/ntserver/nts/downloads/recommended/sp5/allsp5.asp` or alternatively Windows 98 see `http://www.microsoft.com/Windows98/`

- The Hardware Security Module, CAFast (see `http://www.ncipher.com`), to accelerate cryptographic operations and securely store keys. This is an administrator requirement for Identrus member banks and does not require user intervention. Note CAFast is also referred to as nFast.

- Any SPARC Solaris version that iWS supports.

- iPlanet Directory Server 5.0

- In order to run iPlanet Web Server: Plug-in for Trustbase services you must have a relationship with a bank or an authority that is capable of acting as either an OCSP Responder or connected to the Identrus Network.
- Java 2 Standard Edition(Use 1.2.2_01 Native Threads for iWS4.1 and 1.2.2_05 (native threads) for iWS6.0)Production Release for the Solaris Operating Environment.
  `http://www.sun.com/software/solaris/java/download.html`
- Optional. Any database software such as Oracle 8.0.5 and JDBC‘ for Oracle 8.1.5 `http://www.oracle.com/java/jdbc/html/jdbc.html`

HSM Configuration

HSMs are accessed through the PKCS#11 libraries shipped by HSM vendors. In order to use an HSM, the HSM must first be correctly configured for PKCS#11 operation, and then the iPlanet Web Server Plug-in must be configured to recognise the HSM.

Configuring the HSM

An HSM should be configured according to its vendor's instructions. A brief description of the process for nCipher HSMs is provided here, along with a reference to the vendor documentation

Configuring an nCipher HSM

- Refer to the nCipher documentation for definition of terms and further instructions on Security Worlds and Operator Card Sets: Chapters 6 and 7 of the document found at `http://active.ncipher.com/documentation/PKCS11/solaris-4.01/nforce.pdf` are particularly enlightening
- Install the nCipher PKCS #11 library usually into:

```
/opt/nfast
```

- The iPlanet Web Server Plug-in requires a 1 of N Operator Card Set to use an nCipher HSM in PKCS#11 mode. One Operator Card is required for each module in the HSM. Create such an Operator Card Set as specified in the nCipher documentation. The password used must be the same as the password configured in iPlanet Web Server: Plug-in for Trustbase services.

- Create a new text file cknfastrc in the directory in which you installed the nCipher software, usually /opt/nfast, and add the lines:

```
CKNFAST_NO_UNWRAP=1
CKNFAST_LOADSHARING=1
CKNFAST_NO_ACCELERATOR_SLOTS=1

export CKNFAST_NO_UNWRAP CKNFAST_LOADSHARING
CKNFAST_NO_ACCELERATOR_SLOTS
```

- Check the installation using

```
/opt/nfast/bin/ckcheckinst
```

- In the following sections, where the vendor PKCS#11 library is referred to, take that reference to mean

```
/opt/nfast/gcc/lib/libcknfast.so
```

- Additionally, the name of the PKCS#11 token upon which private keys will be generated and stored is the name of the Operator Card Set created for use with the nCipher PKCS#11 interface

Configuring the iPlanet Web Server Plug-in

There are two steps to be taken in configuring the iPlanet Web Server Plug-in:

- Identifying the HSM vendor's PKCS#11 library to the Plug-in PKCS#11 cryptographic services
- Configuring the iPlanet Web Server: Plug-in for Trustbase services to use the HSM based PKCS#11 tokens for key storage. This may be done as part of the installation procedure, but the manual operation is detailed here to permit an HSM to be installed after installation

Identifying the vendor PKCS#11 libraries

- Change to the directory.

```
<iws_install_directory>/https-<servername_instance>/alias
```

- If the file secmod.db does not exist in the alias directory create it as follows:

```
<iws_install_directory>/bin/https/admin/bin/modutil
  -dbdir . -nocertdb -create
```

- If modutil created a secmodule.db rather than a secmod.db, move the file

```
mv secmodule.db secmod.db
```

- Add the vendor PKCS#11 library to the database of PKCS#11 modules, using an appropriate module name, e.g. nFast for an nCipher nFast module

```
<iws_install_directory>/bin/https/admin/bin/modutil
  -dbdir . -nocertdb
  -add <moduleName>
  -libfile <vendorPKCS#11Library>
  -mechanisms RSA:DSA
```

- Check that the module was installed using

```
<iws_install_directory>/bin/https/admin/bin/modutil
  -dbdir . -nocertdb -list
```

- The output should look something like this

```
Using database directory ....
Listing of PKCS #11 Modules
Listing of PKCS #11 Modules
-----------------------------------------------------------
1.<moduleName>
library name: <vendorPKCS#11Library>
slots: # slots attached
status: loaded
slot: ####-####-####-#
token: <tokenName>
slot: ####-####-####-#
token: <anotherTokenName>
...
2. Netscape Internal PKCS #11 Module
slots: 2 slots attached
status: loaded
slot: Communicator Internal Cryptographic Services Version 4.0
token: Communicator Generic Crypto Svcs
slot: Communicator User Private Key and Certificate Services
token: Communicator Certificate DB
----------------------------------------------------------
```

Configuring the iPlanet Web Server Plug-in to use a PKCS#11 token

- If the iPlanet Web Server Plug-in was configured at install time to use a PKCS#11 token, and the correct token name was chosen, then no further steps need be taken, and the Web Server Plug-in will use the HSM for key generation and storage
- If the iPlanet Web Server Plug-in was not installed to use a specific PKCS#11 token, or the token name was specified incorrectly, then these actions should be followed.
- This process must be performed after plug-in installation but before certificate application.
- For iWS4.1 Change directory to

```
cd <iws_install_directory>/docs/servlet
```

- for iWS6.0 Change Directory to

```
cd <iwstps_install_directory>/WEB-INF/lib
```

- Edit the file

```
../jssconfig/trustbase/security/jsstokenkeystore
```

- Change the key.token property line contained therein thus:

```
key.token=<tokenName>
```

- Save the file, leaving other lines untouched
- Restart the iPlanet Web Server, which is now configured to use the PKCS#11 token with the given name for key generation and storage operations

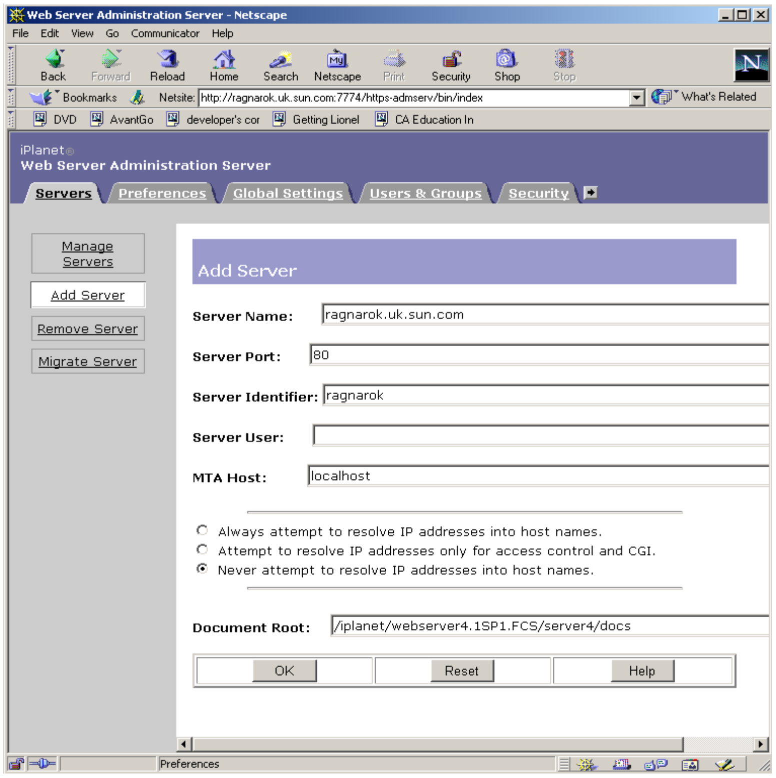
Preparing to Configure Your Web Server

For adminstrative convenience, you can install more than one Webserver instance. You should consult the WebServer guide on how to do this:

- iWS 4.1
  http://docs.iplanet.com/docs/manuals/fasttrak/41/ag/esgstart.htm#998517
- iWS 6.0
  http://docs.iplanet.com/docs/manuals/enterprise/50/ag/esgstart.htm#1003083

Under such circumstances all other steps of the installation procedure would need to be repeated for each instance.

Figure 2-1 Configuring more than one Web Server Instance


Web Server Administration Server - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://ragnarok.uk.sun.com:7774/https-admserv/bin/index What's Related
DVD AvantGo developer's cor Getting Lionel CA Education In
iPlanet
Web Server Administration Server
Servers Preferences Global Settings Users & Groups Security
Manage Servers
Add Server
Remove Server
Migrate Server
OK Reset Help
Preferences

Add Server

Server Name: ragnarok.uk.sun.com

Server Port: 80

Server Identifier: ragnarok

Server User:

MTA Host: localhost

- ○ Always attempt to resolve IP addresses into host names.
- ○ Attempt to resolve IP addresses only for access control and CGI.
- [✓ Never attempt to resolve IP addresses into host names.]

Document Root: /iplanet/webserver4.1SP1.FCS/server4/docs

Certificate Overview

Three kinds of certificates are needed for the Identrus Scheme:

1. Application Authorisation Certificates
   - Relying Participant Certificate[C5]. This is normally the Relying Participant Bank. This certificate is used for Verification purposes.
   - Identrus Root Certificate[C4]. This is normally the Identrus root itself and under such circumstances any Identrus member running applications under the Web Server can be validated.
2. Certificates for Certificate Status Checks
   - The Signing Certificate [C3]– This certificate is used by the webserver plugin to sign all its identrus transactions to the authoritative entity. During installation you will need to give this certificate a nick name.
3. Certificates for Transport Authentication and Integrity
   - The SSL Server Certificate [C1]– This is the certificate that is used by the webserver as its server certificate. It offers this certificate to clients wishing to make SSL Connections. This certificate is not given a nick name by you during the installation process because the webserver will name it – The name it uses by default is “Server-Cert”.
   - The SSL Client Certificate [C2]– This certificate is used by the webserver plugin to make SSL connections with the bank or other authoritative entity. During installation you will need to give this certificate a nick name.

Web Server Configuration Overview

The steps required to get the webserver ready for installation of the plugin are as follows :

1. Install the webserver. You may use the in built JDK of the webserver. If you want to use the plugin to authorise users during login and not just as a DSMS you will need to provide the webserver with a directory server. This directory server can either be registered at installation time or later in the "Global Settings" Tab of the administration server.

2. Initialise the Trust Database – This can be accomplished by starting on the "Servers" Tab of the administration server and clicking manage on your server. Then by clicking on the "Security" Tab for your server. Then by clicking the "Create Database" button and entering the password you want to give the database. Take Note Of This Password You Will Need It Later.

3. Change the Database access permission's – This is required because the database is created by the administration server which is normally run as root and the database is accessed by the webserver instance which is normally run as nobody. Thus you need to change the access permissions. The Database files are stored in a directory underneath the webserver install directory called *alias*. Change the permissions to read / write / execute for everyone on all the files.

4. Apply For Your Certificates – You need to apply for three Certificates for your webserver, these certificates are an SSLServer certificate , and SSLClient Certificate , and an Object Signing Certificate. Each certificate should be requested by using the "Request Certificate" button on the same page as the "Create Database" button. For Each certificate you must fill in the required details the "Key Pair File Password" is the password you noted in step 2.

5. Install Your Certificates – You need to add the certificates you have applied for into the database. Once you have the PEM format replies to your requests, you can use the "Install Certificate" button which is on the same page as the "Request Certificate" button. All three certificates you received should be entered with the type "This Server" , the "Key Pair File Password" is the password you noted in step 2. The first certificate – The SSLServer certificate should not be given a name in the "Certificate name" field. However the other two certificates should be given individual names.

   I

NOTE Make a note of these names. You will need them later

6. Install the Identrus Chain Certificates – You need to add the certificates that complete the trust chain between the certificates you requested and the Identrus Root. You must install all the certificates including the Identrus Root. You can do this using the same page “Install Certificate” youwill need to use the type “Trusted Certificate Authority”.
7. Secure Your Webserver – By enabling SSL and configuring it to require a client certificate. This can be accomplished by starting on the “Servers” Tab of the administration server and clicking manage on your server. Then by clicking the “Preferences” Tab of your webserver. Then by clicking the “Encryption On / Off button”, You can then select encryption on and save your changes. Now Click on the “Encryption Preferences” button and enable “Require Client Certificate”.
8. You need to add a client certificate for DSMSDemo

Configuring iPlanet Web Server

The Web server must be installed and running

1. (See the iPlanet Web Server Documentation for details on how to install .

    a. iWS 4.1

http://docs.iplanet.com/docs/manuals/fasttrak/41/ig/contents.htm

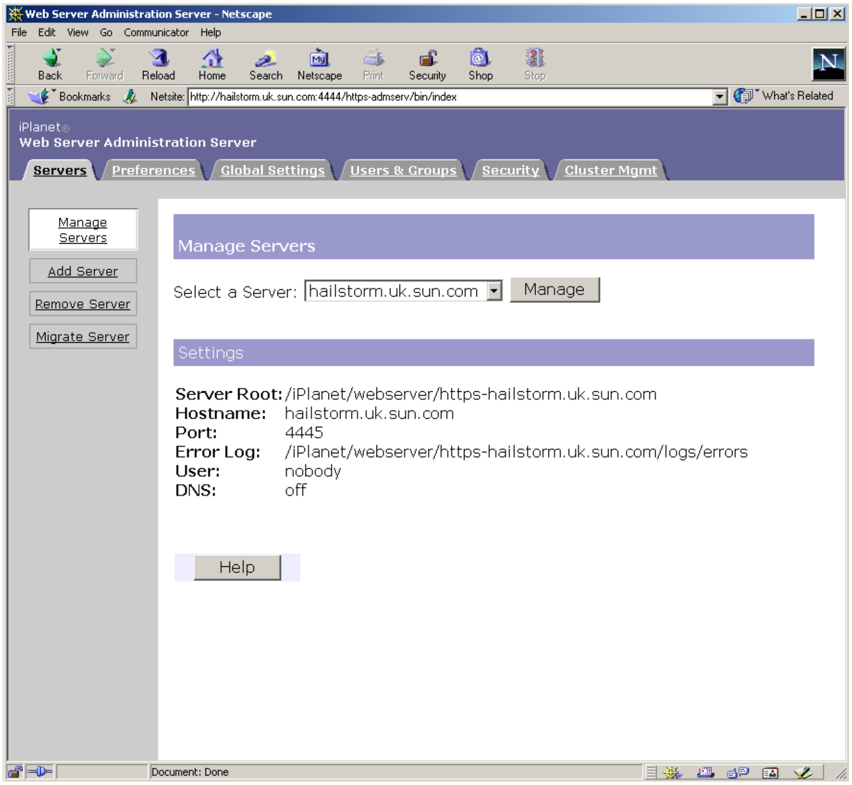
    b. iWS 6.0

http://docs.iplanet.com/docs/manuals/enterprise/50/ig/contents.htm

2. We now Illustrate this for iWS4.1. The steps for iWS 6.0 are similar. To Configure Securely:

    a. Open the Admin Web Server at http://hailstorm.uk.sun.com:4444

Figure 2-2 Opening the Administrator Web Server Screen


Web Server Administration Server - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks
Netsite: http://hailstorm.uk.sun.com:4444/https-admserv/bin/index
What's Related
iPlanet
Web Server Administration Server
Servers
Preferences
Global Settings
Users & Groups
Security
Cluster Mgmt
Manage Servers
Add Server
Remove Server
Migrate Server
Manage
Help
Document: Done

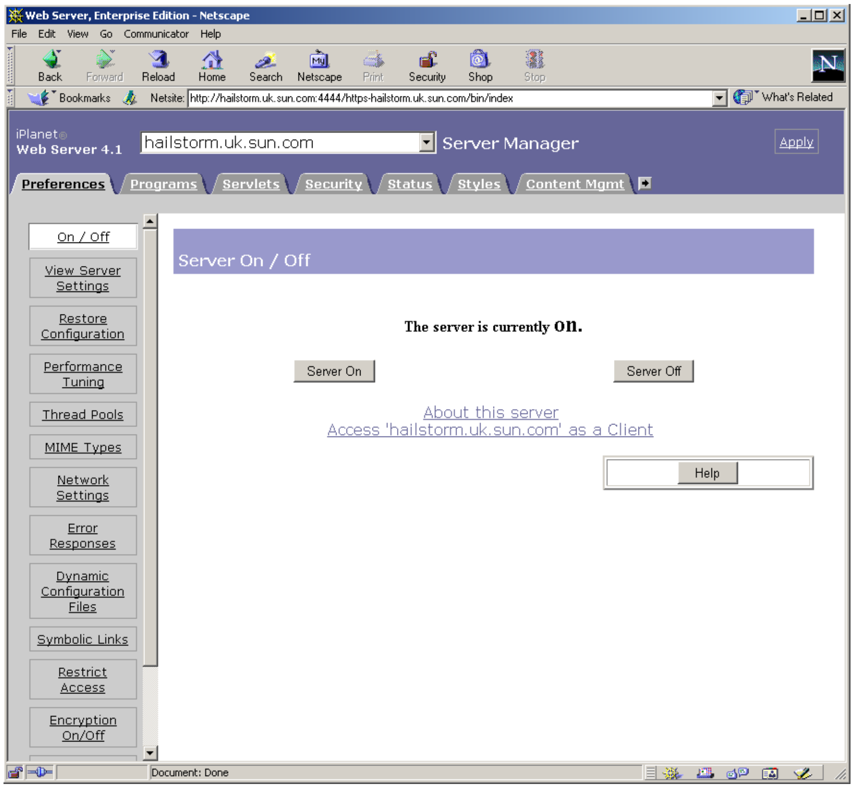
Manage Servers

Select a Server: hailstorm.uk.sun.com

Settings

Server Root:	/iPlanet/webserver/https-hailstorm.uk.sun.com
Hostname:	hailstorm.uk.sun.com
Port:	4445
Error Log:	/iPlanet/webserver/https-hailstorm.uk.sun.com/logs/errors
User:	nobody
DNS:	off

b. Select <manage>

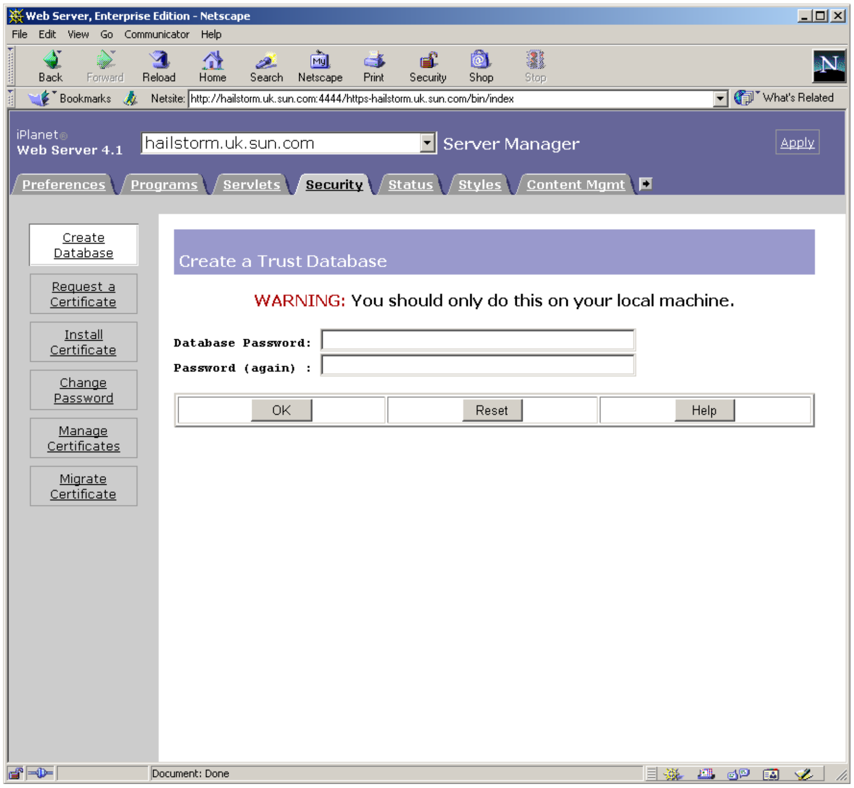

Web Server, Enterprise Edition - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://hailstorm.uk.sun.com:4444/https-hailstorm.uk.sun.com/bin/index
What's Related
iPlanet Web Server 4.1
hailstorm.uk.sun.com
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt
On / Off
View Server Settings
Restore Configuration
Performance Tuning
Thread Pools
MIME Types
Network Settings
Error Responses
Dynamic Configuration Files
Symbolic Links
Restrict Access
Encryption On/Off
Server On / Off
Server On
Server Off
About this server
Access 'hailstorm.uk.sun.com' as a Client
Help
Document: Done

The server is currently on.

c. Select security

d. choose a password and select ok

e. Select <Manage Certificates>


Web Server, Enterprise Edition - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://hailstorm.uk.sun.com:4444/https-hailstorm.uk.sun.com/bin/index
What's Related
iPlanet Web Server 4.1
hailstorm.uk.sun.com
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt
Create Database
Request a Certificate
Install Certificate
Change Password
Manage Certificates
Migrate Certificate
OK Reset Help
Document: Done

Create a Trust Database

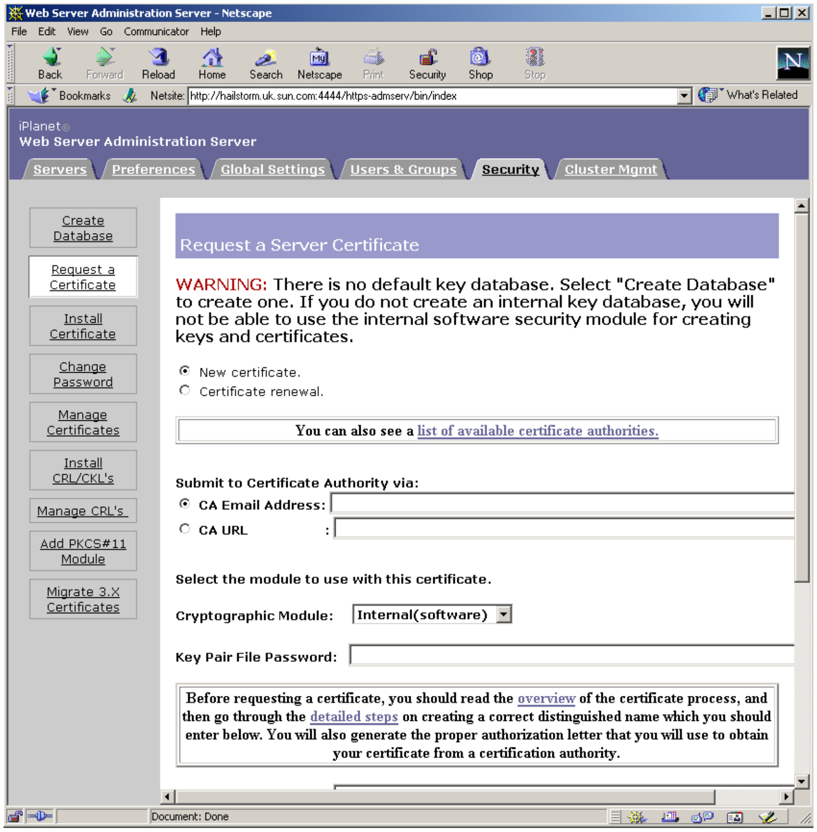
WARNING: You should only do this on your local machine.

Database Password:

Password (again) :

f. Select <request certificate>

Figure 2-3 Request Server Certificate



Request a Server Certificate

WARNING: There is no default key database. Select "Create Database" to create one. If you do not create an internal key database, you will not be able to use the internal software security module for creating keys and certificates.

[✓ New certificate.]
Certificate renewal.

You can also see a list of available certificate authorities.

Submit to Certificate Authority via:

[✓ CA Email Address:]
CA URL :

Select the module to use with this certificate.

Cryptographic Module: Internal(software)

Key Pair File Password:

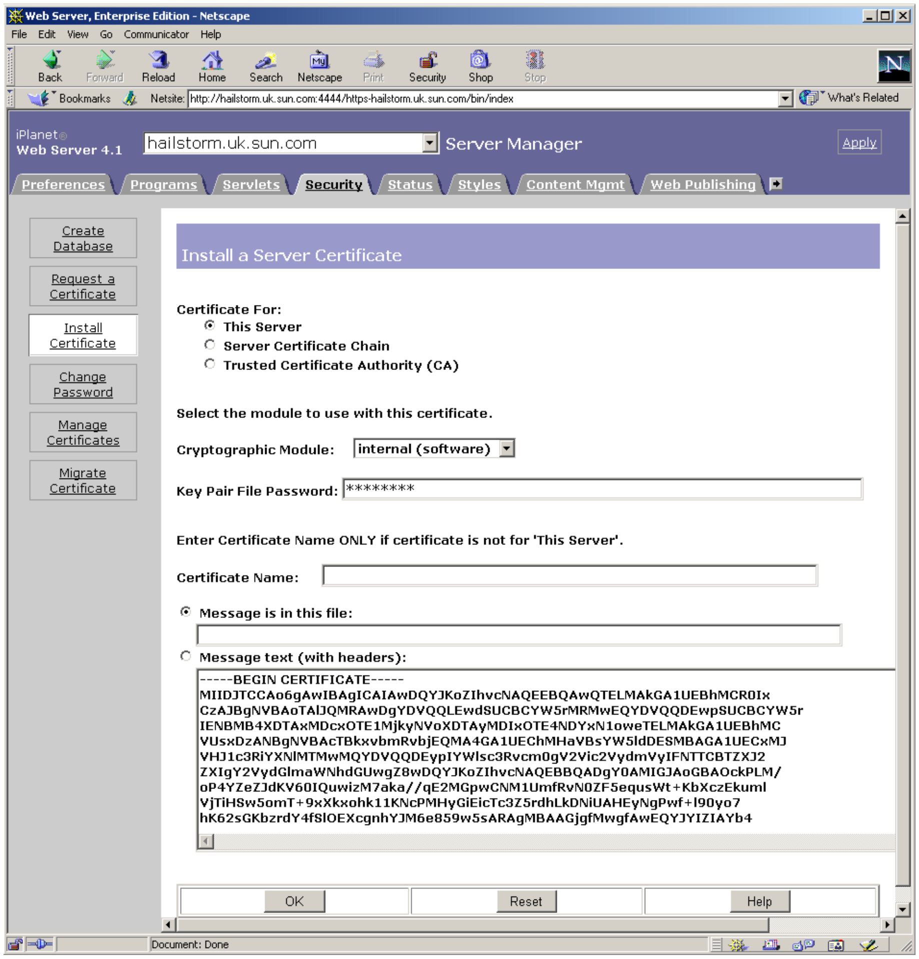
Before requesting a certificate, you should read the overview of the certificate process, and then go through the detailed steps on creating a correct distinguished name which you should enter below. You will also generate the proper authorization letter that you will use to obtain your certificate from a certification authority.

g. fill in form and collect request

h. Give it to your BankCA who will send you a PEM format response

i. Select <Install Certificate>

Figure 2-4 Install Server Certificate


Web Server, Enterprise Edition - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://hailstorm.uk.sun.com:4444/https-hailstorm.uk.sun.com/bin/index
What's Related
iPlanet Web Server 4.1
hailstorm.uk.sun.com
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt Web Publishing
Create Database
Request a Certificate
Install Certificate
Change Password
Manage Certificates
Migrate Certificate
OK Reset Help
Document: Done

Install a Server Certificate

Certificate For:
- [● This Server]
- ○ Server Certificate Chain
- ○ Trusted Certificate Authority (CA)

Select the module to use with this certificate.

Cryptographic Module: internal (software)

Key Pair File Password: *********

Enter Certificate Name ONLY if certificate is not for 'This Server'.

Certificate Name:

[● Message is in this file:]

○ Message text (with headers):

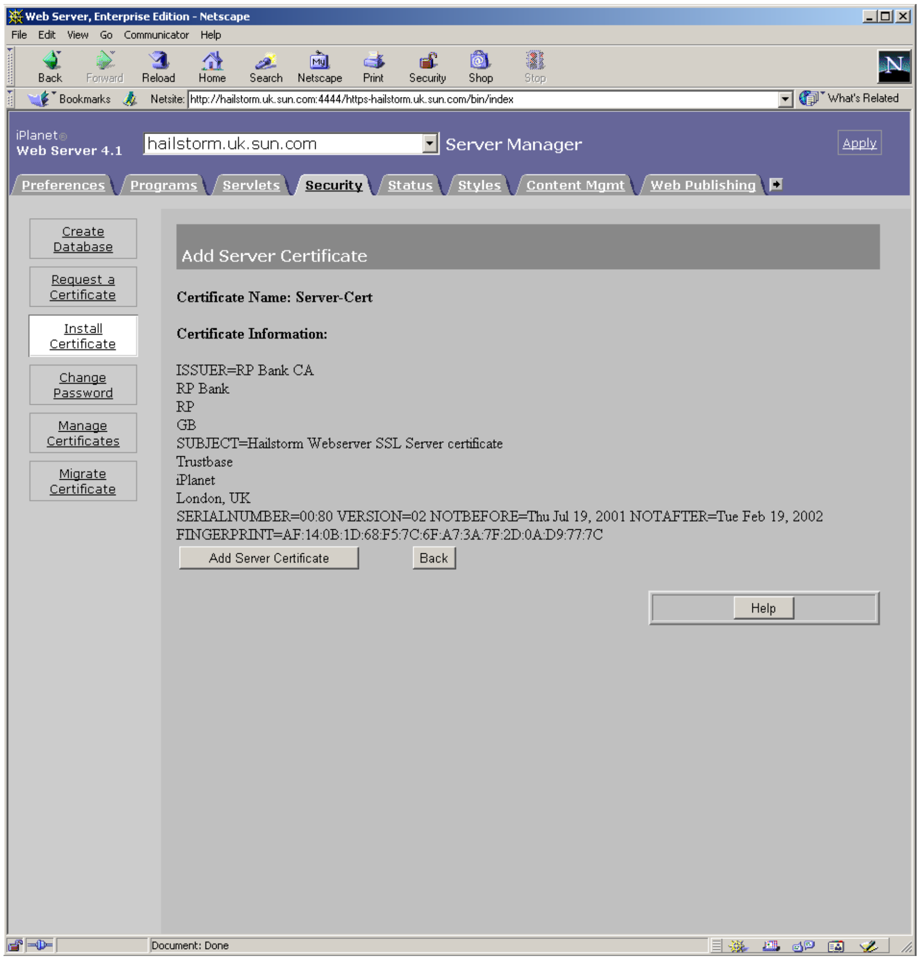
```
-----BEGIN CERTIFICATE-----
MIIDJTCCAo6gAwIBAgICAIAwDQYJKoZIhvcNAQEEBQAwQTELMAkGA1UEBhMCR0Ix
CzAJBgNVBAoTAlJQMRAwDgYDVQQLEwdSUCBCYW5rMRMwEQYDVQQDEwpSUCBCYW5r
IENBMB4XDTAxMDcxOTE1MjkyNVoXDTAyMDIxOTE4NDYxN1oweTELMAkGA1UEBhMC
VUsxDzANBgNVBAcTBkxvbmRvbjEQMA4GA1UEChMHaVBsYW5ldDESMBAGA1UECxMJ
VHJ1c3RiYXNlMTMwMQYDVQQDEypIYWlsc3Rvcm0gV2Vic2VydmVyIFNTTCBTZXJ2
ZXIgY2VydGlmaWNhdGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOckPLM/
oP4YZeZJdKV60IQuwizM7aka//qE2MGpwCNM1UmfRvN0ZF5equsWt+KbXczEkuml
VjTiHSw5omT+9xXkxohk11KNcPMHyGiEicTc3Z5rdhLkDNiUAHEyNgPwf+l90yo7
hK62sGKbzrdY4fSlOEXcgnhYJM6e859w5sARAgMBAAGjgfMwgfAwEQYJYIZIAYb4
```

j. Select <this Server> for [C1],[C2],[C3] <Trusted Certificate Authority> for [C4] and [C5]

k. Select <Message text>

l. Select <Add Certificate>

Figure 2-5 Add Server Certificate


Web Server, Enterprise Edition - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://hailstorm.uk.sun.com:4444/https-hailstorm.uk.sun.com/bin/index
What's Related
iPlanet Web Server 4.1
hailstorm.uk.sun.com
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt Web Publishing
Create Database
Request a Certificate
Install Certificate
Change Password
Manage Certificates
Migrate Certificate
Add Server Certificate
Back
Help
Document: Done

Add Server Certificate

Certificate Name: Server-Cert

Certificate Information:

ISSUER=RP Bank CA
RP Bank
RP
GB
SUBJECT=Hailstorm Webserver SSL Server certificate
Trustbase
iPlanet
London, UK
SERIALNUMBER=00:80 VERSION=02 NOTBEFORE=Thu Jul 19, 2001 NOTAFTER=Tue Feb 19, 2002
FINGERPRINT=AF:14:0B:1D:68:F5:7C:6F:A7:3A:7F:2D:0A:D9:77:7C

m. Repeat for all three certs [C1,C2, C3]

n. Add Identrus Root Certificate from your CA[C4]

o. Add RPCA from your CA [C5]

p. To check that all certificates are present and correct

- Select <Security>
- Select <Manage Certificates>

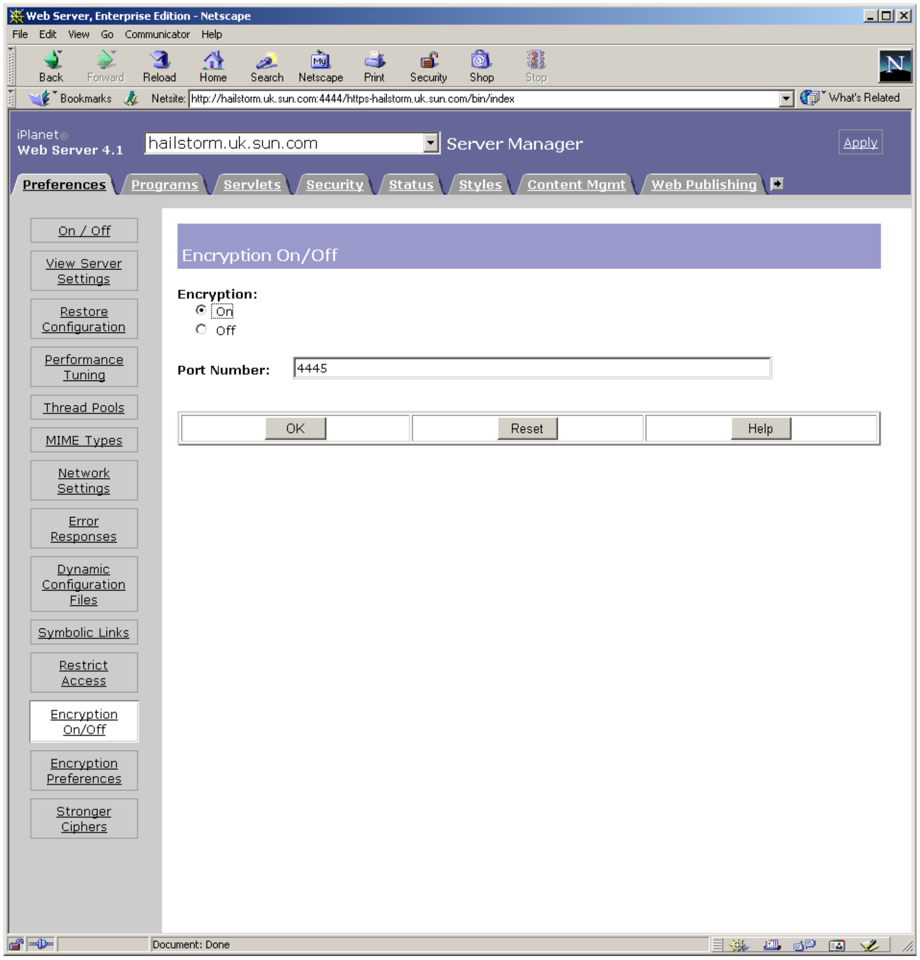
q. To Secure your Webserver

r. Select <preferences>

s. select <encryption on / off>

t. Select <on>

Figure 2-6 Set Encyption



Encryption On/Off

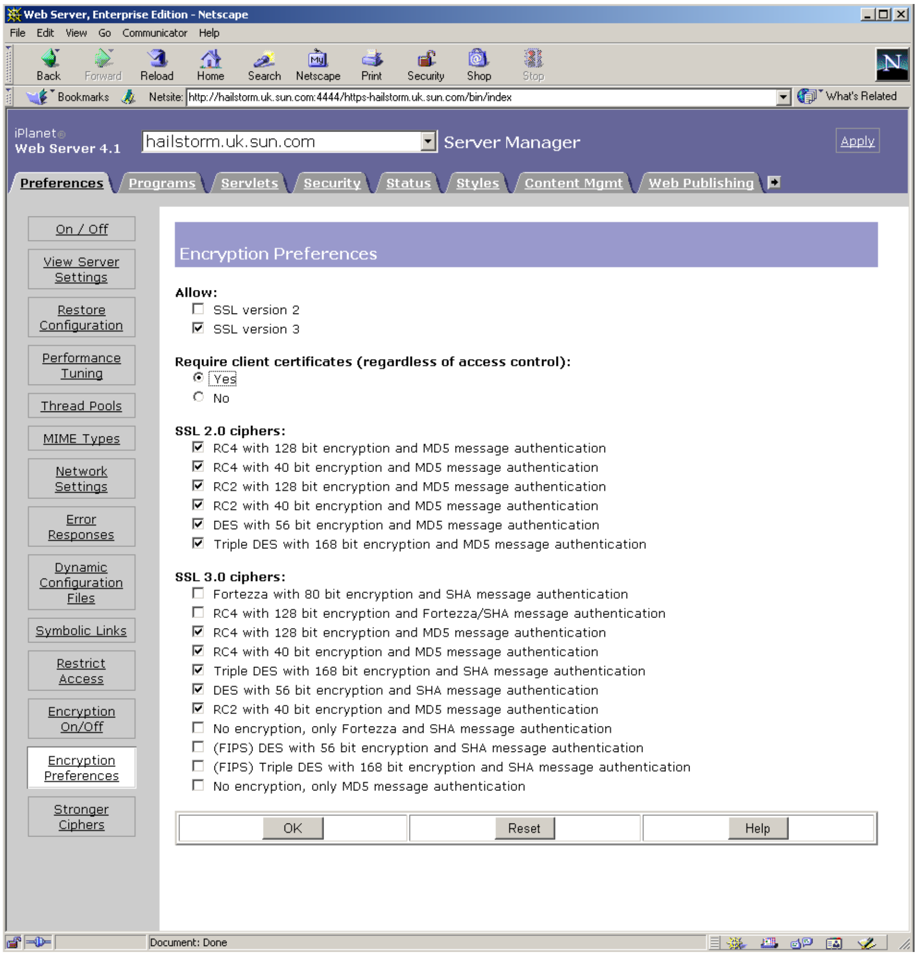
Encryption: [✓ On] Off

Port Number: 4445

u. Select <ok> followed by <Save Changes>

v. Select <encryption preferences>

Figure 2-7 Encryption Settings


Web Server, Enterprise Edition - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: http://hailstorm.uk.sun.com:4444/https-hailstorm.uk.sun.com/bin/index
What's Related
iPlanet Web Server 4.1
hailstorm.uk.sun.com
Server Manager
Apply
Preferences Programs Servlets Security Status Styles Content Mgmt Web Publishing
On / Off
View Server Settings
Restore Configuration
Performance Tuning
Thread Pools
MIME Types
Network Settings
Error Responses
Dynamic Configuration Files
Symbolic Links
Restrict Access
Encryption On/Off
Encryption Preferences
Stronger Ciphers
Document: Done

Encryption Preferences

Allow:

- ☐ SSL version 2
- ☑ SSL version 3

Require client certificates (regardless of access control):

- [✓ Yes]
- No

SSL 2.0 ciphers:

- ☑ RC4 with 128 bit encryption and MD5 message authentication
- ☑ RC4 with 40 bit encryption and MD5 message authentication
- ☑ RC2 with 128 bit encryption and MD5 message authentication
- ☑ RC2 with 40 bit encryption and MD5 message authentication
- ☑ DES with 56 bit encryption and MD5 message authentication
- ☑ Triple DES with 168 bit encryption and MD5 message authentication

SSL 3.0 ciphers:

- ☐ Fortezza with 80 bit encryption and SHA message authentication
- ☐ RC4 with 128 bit encryption and Fortezza/SHA message authentication
- ☑ RC4 with 128 bit encryption and MD5 message authentication
- ☑ RC4 with 40 bit encryption and MD5 message authentication
- ☑ Triple DES with 168 bit encryption and SHA message authentication
- ☑ DES with 56 bit encryption and SHA message authentication
- ☑ RC2 with 40 bit encryption and MD5 message authentication
- ☐ No encryption, only Fortezza and SHA message authentication
- ☐ (FIPS) DES with 56 bit encryption and SHA message authentication
- ☐ (FIPS) Triple DES with 168 bit encryption and SHA message authentication
- ☐ No encryption, only MD5 message authentication

OK | Reset | Help

w. Select <Require Client Certificate> = “Yes”

x. Select <ok><Save Change>

y. Start the server up again

z. the server should now be secure

aa. Type https://hailstorm.uk.sun.com:4445 to verify

ab. You may need to get a SSL Client cert for the RPCA [C5]

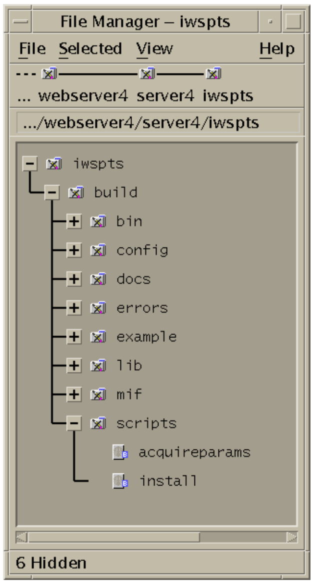
Installing the Plug-in

You will have been provided with a single compressed tar file which contains the installation of the plugin. The compressed tar file which is normally called *iwstps.tar.Z* should be copied to a directory of your choice and then unpacked. If you are using the c-shell an example as to how unpack the compressed tar file would be :

```
gzip -c -d iwstps-1.0.tar.gz | ( mkdir iwstps; cd iwtps ; tar xvf - )
```

Once you have unpacked the distribution it will have created several directories below the one you are currently in. The scripts directory contains the install scripts and data they need. The errors directory contains html files that the plugin will use when in operation. It also contains an applet that performs some of the required smart card interactions. The bin and lib directories contain the actual program code that represent the plugin.

Figure 2-8 Example Unzipped iwspts directory for iWS4.1



```
iwspts
  build
    bin
    config
    docs
    errors
    example
    lib
    mif
    scripts
      acquireparams
      install
```

iWS 4.1 Plug-in for Trustbase Services Installation Overview

Before installing the plugin you should have several pieces of information that it will require ready to hand. The information that the plugin will need is as follows. It should be noted that the defaults that are named for explanatory purposes and are not used by the install process.

- **The webserver install directory** This is the full path to the webserver install directory. This directory is by default */usr/netscape/server4*. The directory should contain the *startconsole* program and several subdirectories.
- **The webserver label name** : This is the full label name you have given to the instance of the webserver you want to install the plugin for. The label name is typically https-machine-domainname so it may look something like https-ragnarok-uk.sun.com.
- **The full path to the webserver documents directory** : This should be the full pathname to the webserver instance's documents directory. This is by default */usr/netscape/server4/docs*.
- **The full path to the webserver servlet directory** : This should the full pathname to the webserver instance's servlet directory. This is by default */usr/netscape/server4/docs/servlet*. Although it is important to note that this directory is not created when you install the webserver.
- **The keystore password** : This is the password you gave when you created the keystore database.
- **The nick name of your server's signing certificate** : This is the nick name of the certificate you wish the webserver to use when signing outgoing identrus transactions. This will be the nick name of the certificate you requested for object signing.
- **The nick name of your server's SSL Signing certificate** : This is the nick name of the certificate you wish the webserver to use when connecting as a client in SSL connections. This will be the nick name of the certificate you requested for SSL client usage.

- **The domain within which the server operates** : This is for the authorisation system – it controls how wide the domain for the cookie authorisation is and it must be wide enough to include this host. Thus it must be at least one step above this host , eg if your host is called blizzard.uk.sun.com the domain must be at least uk.sun.com or wider.
- **The nick name of the verification certificate** : This is the nick name of the certificate you wish to use as an authoritative root certificate or "Trust Anchor". In this case the Identrus Root.
- The following LDAP settings, obtained from <Global Settings>
  - **The fully qualified hostname of your LDAP database server** : This is the hostname and domain name of the machine that has the LDAP database that store the user profiles that the webserver uses.
  - **The port number that your LDAP database is running on** : This is the port number that the LDAP database is listening on.
  - **The Base DN of your LDAP Database** : This is the base DN for user searches of your LDAP database.

iWS 6.0 Plug-in for Trustbase Services Installation Overview

Before installing the plugin you should have several pieces of information that it will require ready to hand. The information that the plugin will need is as follows. It should be noted that the defaults that are named for explanatory purposes and are not used by the install process.

- **The webserver install directory** This is the full path to the webserver install directory. This directory is by default */usr/netscape/server4*. The directory should contain the *startconsole* program and several subdirectories.
- **The webserver instance name** : This is the name of the instance of the webserver you wish to install the plug-in for. By default this is <machine_name>.<domain_name>
- **The Deployment Directory** : This is the full path tothe directory you wish to employ the plug-in application to.
- **The Virtual Server Name :** Allows you to serve different domains with the same Web Server. This, by default, would be the https-<instance-name>
- **The keystore password** : This is the password you gave when you created the keystore database.
- **The nick name of your server's signing certificate** : This is the nick name of the certificate you wish the webserver to use when signing outgoing identrus transactions. This will be the nick name of the certificate you requested for object signing.
- **The nick name of your server's SSL Signing certificate** : This is the nick name of the certificate you wish the webserver to use when connecting as a client in SSL connections. This will be the nick name of the certificate you requested for SSL client usage.
- **The domain within which the server operates** : This is for the authorisation system – it controls how wide the domain for the cookie authorisation is and it must be wide enough to include this host. Thus it must be at least one step above this host , eg if your host is called blizzard.uk.sun.com the domain must be at least uk.sun.com or wider.
- **The nick name of the verification certificate** : This is the nick name of the certificate you wish to use as an authoritative root certificate or "Trust Anchor". In this case the Identrus Root.

- The following LDAP settings, obtained from <Global Settings>
    - **The fully qualified hostname of your LDAP database server** : This is the hostname and domain name of the machine that has the LDAP database that store the user profiles that the webserver uses.
    - **The port number that your LDAP database is running on** : This is the port number that the LDAP database is listening on.
    - **The Base DN of your LDAP Database** : This is the base DN for user searches of your LDAP database.
    -

Performing the iWS4.1 Plug-in for Trustbase Services Installation

Once you have gathered all the information you are ready to proceed with the installation. Inside the scripts directory you unpacked earlier there is a script called "install". If you run this script it will ask you all the questions detailed above. Once the questions have been answered the script will proceed to copy all the binaries required and change the configuration files for the plugin's operation. Example of what might happen during installation is shown below. Bold type is used to indicate user typed instructions. In the cases where nothing appears to be typed by user it means the user just pressed return and accepted the proposed default.

```
ragnarok# cd <install_dir>/iwspts/build/scripts

ragnarok# ./install

Where is your iPlanet WebServer installation located?

/iplanet/webserver3/server4

What is the instance of your WebServer eg https-ragnarok-PKI ?

https-ragnarok.UK.Sun.COM

What is the full path to the documents directory you wish to use
? [ /iplanet/webserver3/server4/docs ]
/iplanet/webserver3/server4/docs

What is the full path to the servlet directory you wish to use ?
[ /iplanet/webserver3/server4/docs/servlet ]

What is your keystore password ? password

What is the domain name within which this webserver operates ?

uk.sun.com

What is the nick name server's signing cert ? [ Server-Cert ]

What is the nick name server's SSL signing cert ? [ Server-Cert ]

What is the nick name of the cert you wish to verify responses
with ? Identrus Root (Development)

What is the fully qualified hostname of your LDAP database server
? ragnarok.uk.sun.com

What is the port number of your LDAP database server ?

10000

What is the base DN of your LDAP database ? o=sun.com
```

```
These are the parameters that you input

[1] The server location is [ /iplanet/webserver3/server4 ]

[2] The server label is [ https-ragnarok.UK.Sun.COM ]

[3] The documents path is [ /iplanet/webserver3/server4/docs ]

[4] The servlet path is [
/iplanet/webserver3/server4/docs/servlet ]

[5] The keystore password is [ password ]

[6] The signing certificate nick name is [ Server-Cert ]

[7] The SSL signing certificate nick name is [ Server-Cert ]

[8] The domain name within which this webserver operates [
uk.sun.com ]

[9] The verification certificate nick name is [ Identrus Root
(Development) ]

[10] The hostname of your LDAP database server is [
ragnarok.uk.sun.com ]

[11] The port number of your LDAP database server is [ 10000 ]

[12] The base DN of your LDAP datbase is [ o=sun.com ]

if these are acceptable hit [0] otherwise hit the number of the
parameter you wish to change or hit [e] to leave the installation

0

domain name - niscafe.uk.sun.com host name - ragnarok when you
wish to install the cookiechecker shared object please add the
contents of the file
/iplanet/webserver3/server4/https-ragnarok.UK.Sun.COM/config/obj
.conf.cookie to the
/iplanet/webserver3/server4/https-ragnarok.UK.Sun.COM/config/obj
.conf file
```

Performing the iWS 6.0 plug-in for Trustbase Services installation

```
Where is your iPlanet WebServer installation located?

/iplanet/webserver6.FCS/server6

What is the name of the instance your WebServer instance ?

ragnarok.uk.sun.com

What is the instance's virtual server called ? [ default ]

What is the full path to the directory you wish to deploy the
application to ? [ /iplanet/webserver6.FCS/server6/deploy ]

What is your keystore password ?

password

What is the domain name within which this webserver operates ?

uk.sun.com

What is the nick name server's signing cert ? [ Server-Cert ]

What is the nick name server's SSL signing cert ? [ Server-Cert ]

What is the nick name of the cert you wish to verify responses
with ?

Identrus Root (Development)

What is the fully qualified hostname of your LDAP database server
? [ ragnarok ]

What is the port number of your LDAP database server ? [ 389 ]

10000

What is the base DN of your LDAP database ?

o=sun.com
```

These are the parameters that you input

```
[1] The server location is [ /iplanet/webserver6.FCS/server6 ]

[2] The server instance is [ ragnarok.uk.sun.com ]

[3] The virtual server id is [ https-ragnarok.uk.sun.com ]

[4] The deployment directory [
/iplanet/webserver6.FCS/server6/deploy ]

[5] The keystore password is [ password ]

[6] The signing certificate nick name is [ Server-Cert ]

[7] The SSL signing certificate nick name is [ Server-Cert ]

[8] The domain name within which this webserver operates [
uk.sun.com ]

[9] The verification certificate nick name is [ Identrus Root
(Development) ]

[10] The hostname of your LDAP database server is [ ragnarok ]

[11] The port number of your LDAP database server is [ 10000 ]

[12] The base DN of your LDAP datbase is [ o=sun.com ]

if these are acceptable hit [0] otherwise hit the number of the
parameter you wish to change or hit [e] to leave the installation

0

The directory /iplanet/webserver6.FCS/server6/deploy does not
exist

Do you want to create it ?

y

Creating directory

/iplanet/webserver6.FCS/server6/deploy

domain name - uk.sun.com

host name - ragnarok

Web application deploy successful

when you wish to install the cookiechecker shared object please
add the contents of the file
/iplanet/webserver6.FCS/server6/https-ragnarok.uk.sun.com/config
/obj.conf.cookie to the
/iplanet/webserver6.FCS/server6/https-ragnarok.uk.sun.com/config
/obj.conf file
```

Activating the Identrus Login Protection (iWS 4.1)

You cannot utilise the identrus login protection system without having a user community ldap database. In order to protect an area of the webserver from access through an identrus certificate status check you need to take a few final steps in configuration. First you need to add the lines that have been written to the file obj.conf.cookie to your obj.conf file. These files are located in the webserver instance's config directory which is determined by combining the install directory with the instance label. Which in the install example above would be */iplanet/webserver3/server4/https-ragnarok.UK.Sun.COM/config*. The lines with the first word as "Init" should be placed after the last Init line and the line with "PathCheck" should be placed before the first "PathCheck" line. An example altered file follows with the lines in question highlighted.

```
# Sun Netscape Alliance - obj.conf

# You can edit this file, but comments and formatting changes

# might be lost when the admin server makes changes.


Init fn="flex-init"
access="/iplanet/webserver3/server4/https-ragnarok.UK.Sun.COM/lo
gs/access" format.access="%Ses->client.ip% -
%Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"

Init fn="load-types" mime-types="mime.types"

Init fn="load-modules"
shlib="/iplanet/webserver3/server4/bin/https/lib/libNSServletPlu
gin.so"
funcs="NSServletEarlyInit,NSServletLateInit,NSServletNameTrans,N
SServletService" shlib_flags="(global|now)"

Init fn="NSServletEarlyInit" EarlyInit="yes"

Init fn="NSServletLateInit" LateInit="yes"
```

****----> Init fn="load-modules" funcs="identrus-cookie-init" shlib="/iplanet/webserver3/server4/bin/https/lib/cookiechecker.so"**

****-----> Init fn="acl-register-module" module="identrus" func="identrus-cookie-init"**

```
<Object name="default">
```

```
NameTrans fn="NSServletNameTrans" name="servlet"

NameTrans fn="pfx2dir" from="/servlet"
dir="/iplanet/webserver3/server4/docs/servlet"
name="ServletByExt"

NameTrans fn="pfx2dir" from="/ns-icons"
dir="/iplanet/webserver3/server4/ns-icons" name="es-internal"

NameTrans fn="pfx2dir" from="/mc-icons"
dir="/iplanet/webserver3/server4/ns-icons" name="es-internal"

NameTrans fn="pfx2dir" from="/help"
dir="/iplanet/webserver3/server4/manual/https/ug"
name="es-internal"

NameTrans fn="pfx2dir" from="/manual"
dir="/iplanet/webserver3/server4/manual/https"
name="es-internal"

NameTrans fn="document-root"
root="/iplanet/webserver3/server4/docs"
```

****---->PathCheck fn="get-client-cert" dorequest="1"**

```
PathCheck fn="unix-uri-clean"

PathCheck fn="check-acl" acl="default"

PathCheck fn="find-pathinfo"

PathCheck fn="find-index" index-names="index.html,home.html"

ObjectType fn="type-by-extension"

ObjectType fn="force-type" type="text/plain"

Service method="(GET|HEAD)" type="magnus-internal/imagemap"
fn="imagemap"

Service method="(GET|HEAD)" type="magnus-internal/directory"
fn="index-common"

Service fn="NSServletService" type="magnus-internal/jsp"

Service method="(GET|HEAD)" type="*~magnus-internal/*"
fn="send-file"

AddLog fn="flex-log" name="access"

</Object>


<Object name="cgi">

ObjectType fn="force-type" type="magnus-internal/cgi"
```

```
Service fn="send-cgi"
</Object>

<Object name="servlet">
ObjectType fn="force-type" type="text/html"
Service fn="NSServletService"
</Object>

<Object name="jsp092">
ObjectType fn="type-by-extension"
ObjectType fn="change-type" type="magnus-internal/jsp092"
if-type="magnus-internal/jsp"
Service fn="NSServletService" type="magnus-internal/jsp092"
</Object>

<Object name="ServletByExt">
ObjectType fn="force-type" type="magnus-internal/servlet"
Service type="magnus-internal/servlet" fn="NSServletService"
</Object>

<Object name="es-internal">
PathCheck fn="check-acl" acl="es-internal"
</Object>

<Object name="jsp">
Service fn="NSServletService"
</Object>
```

Activating the Identrus Login Protection (iWS 6.0)

Two files need to be edited

obj.conf

```
NameTrans fn="pfx2dir" from="/servlet"
dir="/iplanet/webserver6.CPI/server6/docs/servlet"
name="ServletByExt"

NameTrans fn="pfx2dir" from="/jsp.092"
dir="/iplanet/webserver6.CPI/server6/docs/jsp.092" name="jsp092"

NameTrans fn=pfx2dir from=/mc-icons
dir="/iplanet/webserver6.CPI/server6/ns-icons"
name="es-internal"

NameTrans fn="pfx2dir" from="/manual"
dir="/iplanet/webserver6.CPI/server6/manual/https"
name="es-internal"

NameTrans fn=document-root root="$docroot"
```

****---->PathCheck fn="get-client-cert" dorequest="1"**

```
PathCheck fn=unix-uri-clean

PathCheck fn="check-acl" acl="default"

PathCheck fn=find-pathinfo

PathCheck fn=find-index index-names="index.html,home.html"

ObjectType fn=type-by-extension

ObjectType fn=force-type type=text/plain

Service type="magnus-internal/jsp" fn="NSServletService"

Service method=(GET|HEAD) type=magnus-internal/imagemap
fn=imagemap

Service method=(GET|HEAD) type=magnus-internal/directory
fn=index-common

Service method=(GET|HEAD) type=*~magnus-internal/* fn=send-file

AddLog fn=flex-log name="access"

</Object>


<Object name=cgi>
```

```
ObjectType fn=force-type type=magnus-internal/cgi

Service fn=send-cgi user="$user" group="$group" chroot="$chroot"
dir="$dir" nice="$nice"

</Object>


<Object name="servlet">

ObjectType fn=force-type type=text/html

Service fn="NSServletService"

</Object>


<Object name="jsp092">

ObjectType fn="type-by-extension"

ObjectType fn="change-type" type="magnus-internal/jsp092"
if-type="magnus-internal/jsp"

Service fn="NSServletService" type="magnus-internal/jsp092"

</Object>


<Object name="ServletByExt">

ObjectType fn=force-type type=magnus-internal/servlet

Service type="magnus-internal/servlet" fn="NSServletService"

</Object>


<Object name="es-internal">

PathCheck fn="check-acl" acl="es-internal"

</Object>


magnus.conf

#ServerRoot
/iplanet/webserver6.CPI/server6/https-ragnarok.uk.sun.com

ServerID https-ragnarok.uk.sun.com

ServerName ragnarok.uk.sun.com
```

```
ErrorLog
/iplanet/webserver6.CPI/server6/https-ragnarok.uk.sun.com/logs/e
rrors

PidLog
/iplanet/webserver6.CPI/server6/https-ragnarok.uk.sun.com/logs/p
id

User nobody

MtaHost localhost

DNS off

Security off

ClientLanguage en

AdminLanguage en

DefaultLanguage en

RqThrottle 128

StackSize 131072

CGIWaitPid on

TempDir /tmp/https-ragnarok.uk.sun.com-8e0cef20

Init fn=flex-init access="$accesslog"
format.access="%Ses->client.ip% - %Req->vars.auth-user%
[%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"

Init fn=load-types mime-types=mime.types

Init fn="load-modules"
shlib="/iplanet/webserver6.CPI/server6/bin/https/lib/libNSServle
tPlugin.so"
funcs="NSServletEarlyInit,NSServletLateInit,NSServletNameTrans,N
SServletService" shlib_flags="(global|now)"

Init fn="NSServletEarlyInit" EarlyInit=yes

Init fn="NSServletLateInit"  LateInit=yes
```

****---->Init fn="load-modules" funcs="identrus-cookie-init" shlib="/iplanet/webserver4.1SP1.FCS/server4/bin/https/lib/cookiechecker.so"**

****---->Init fn="acl-register-module" module="identrus" func="identrus-cookie-init"**

Protecting An Area Using Identrus Method.

Now that you have completed the installation of the “Identrus” access control method you can use it just like any other access control method. Using access control methods is well detailed in the webserver administrators guide See for instance

- iWS 4.1

  http://docs.iplanet.com/docs/manuals/fasttrak/41/ag/esaccess.htm

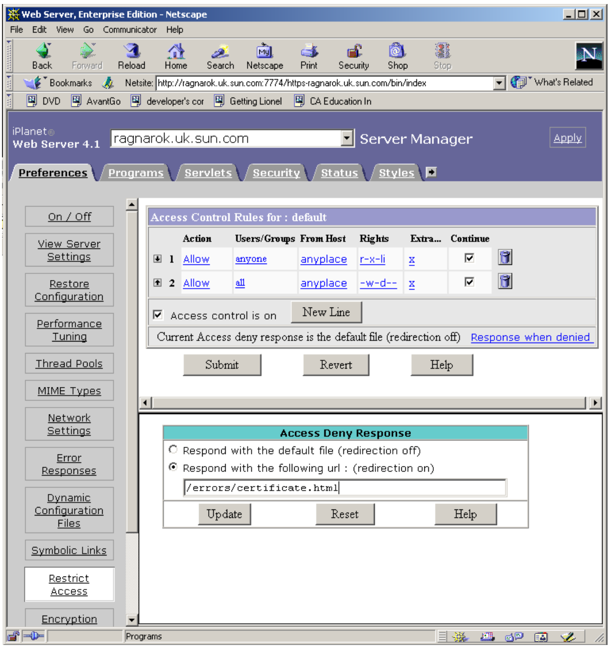
- iWS 6.0

  http://docs.iplanet.com/docs/manuals/enterprise/50/ag/esaccess.htm

Here is an overview of the main steps found in the iWS Administration Guides:

1. Access The administration server screens and <Manage> Your webserver.
2. Click on the <Restrict Access> button on the administration screen which in the lower left hand corner of the screen.

Figure 2-9 Custom Error Page


Web Server, Enterprise Edition - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks
Netsite: http://ragnarok.uk.sun.com:7774/https-ragnarok.uk.sun.com/bin/index
What's Related
DVD AvantGo developer's cor Getting Lionel CA Education In
iPlanet
Web Server 4.1
ragnarok.uk.sun.com
Server Manager
Apply
Preferences Programs Servlets Security Status Styles
On / Off
View Server Settings
Restore Configuration
Performance Tuning
Thread Pools
MIME Types
Network Settings
Error Responses
Dynamic Configuration Files
Symbolic Links
Restrict Access
Encryption
Programs

Access Control Rules for : default

	Action	Users/Groups	From Host	Rights	Extra...	Continue
1	Allow	anyone	anyplace	r-x-li	x	✓
2	Allow	all	anyplace	-w-d--	x	✓

✓ Access control is on New Line

Current Access deny response is the default file (redirection off) Response when denied

Submit Revert Help

Access Deny Response

○ Respond with the default file (redirection off)

● Respond with the following url : (redirection on)

/errors/certificate.html

Update Reset Help

3. Click the browse button on the <Pick A Resource> section and select the area of the webserver you wish to protect.
4. Then click the <Edit Access Control Button>.
5. Create your Access Control List – in the user/group field if you click on the <anyone> field a requestor will appear – in that requestor you can choose which users and groups to protect from or allow, you can also specify the authentication which in this case is “identrus”.
6. Also when protecting an area you must use a custom error page installed by the plugin called /errors/certificate.html as illustrated on the previous page.

Configuring Identrus with LDAP

When using the “identrus” security method the LDAP user community database is consulted to check for the existence of the user that is trying to login. Furthermore the user’s entry is checked to see if it contains the identity certificate that was submitted during the login phase. If the identity certificate is not present in the LDAP record then the user is not allowed to login to the system. In a default installation the user’s identity certificate must be placed in the “usersmimecertificate” element in the record. However you can change this element by altering the “authservlet.properties” file. The section you need to alter is the DataProvider Section and the entry is:

```
LdapDatabase.Property=CertificiateElement=usersmimecertificate
```

If the entry is not already present then just add it with whatever value for “usersmimecertificate” you require.

The location of the authservlet.properties is different on iWS 4.1 and iWS 6.0 –

iWS 4.1 :

```
<iws_install_dir>/https-<instance_name>/config.
```

iWS 6.0 :

```
<iwstps_install_dir>/config.
```

What happens during Installation (iWS 4.1)

The install script copies files to the relevant directories within the Web Server. How this is done can be found in:

```
<install_directory>/build/scripts/install
```

Binaries.

The binaries are copied into the required locations. This means that the jar files are copied into to the servlet directory you selected during your installation. The shared libraries including the custom Access Control library – the "Cookie Authorisation" are copied in the *bin/https/lib* directory under your webservers install directory that you indicated during the install.

Text and resources

The html files , associated images and smart card applets are copied into the documents directory that you specified during your installation – they are all placed underneath an *errors* subdirectory.

Web Server Configuration Files

The webserver configuration files are created or altered as required they all reside in the webserver instance's config directory. The webserver instance directory is created by combining the webserver install directory that you specified on installation with webserver instance label that you also specified during installation. The files that are involved are :

authservlet.properties : This file controls the authentication servlet behaviour and the cookie authorisation shared object behaviour. The settings inside this file are detailed in the configuration section of this document.

jvm12.conf : This file controls the jvm configuration that runs the servlet engine. The classpath must be changed in order to place the servlet and other associated classes onto it. Do not try to change teh classpath from the Configuration Screen <Configure JVM Attributes>. Make these changes directlry to the file jvm12.conf.

servlet.properties This file controls the registration of servlet's on this instance of the webserver. This file is altered to register the adapter class for the authentication servlet. It is also altered to allow the authentication servlet to find the *authservlet.properties* file.

rules.properties This file controls registration of the virtual servlet paths. This file is altered to allow references to /AuthenticationServletAdapter. Because of the restrictions placed on the webserver JVM this servlet must operate through a virtual servlet path.

jsstokenkeystore.properties The JSS security system must also be configured. This means that the jarfile jssconfig is unpacked during installation – since this jarfile is located inside the servlet directory you specified during installation it will be placed on the classpath. The file *jsstokenkeystore.properties* which was part of the jar file is then altered to reflect the correct path to the webserver instances database. The full path to alter *jsstokenkeystore.properties* file is thus *<servlet_directory>/jssconfig/trustbase/security/jsstokenkeystore.properties*

What Happens During Installation (iWS 6.0)

The install script copies files to relevant directories , alters configuration files to suit the local configuration and deploy's the application using wdeploy. How this is done can be found at :

```
<install_directory>/build/scripts/install
```

- Binaries (Shared Objects).

  The shared objects that are required for the plugin's operation are copied into the webserver's library directory <webserver_install_dir> / bin / https / lib this includes the replacement of the libnss3.so with the plugins own version. The current version of libnss3.so is backed up with the name libnss3.original.so.

- Web Deployment.

  All text resources and java jar's are then placed in their appropriate places by using "wdeploy" – the plugin install script uses the webserver's own version.

- Application Configuration.

  The application is then configured to the local specification. The configuration files for the application are stored in the <deployment_dir> / config directory. The application files that are altered are :

  Authservlet.properties : This file controls all aspects of the Authentication Process.

  Dsmsdemo.properties : This is the installation verification utility.

  Web-apps.xml : The web deployment file is altered so that the applications can find their configuration files.

  iWS 6.0 files that are altered are :

  Jvm12.conf : This controls the JVM of the webserver and it is altered to include the classes directory of the deployment.

Post Installation procedure

After ensuring you have followed the installation and HSM configuration steps, the Web Server must be shut down and restarted.

Software Reinstallation

iWS 4.1 Reinstall

1. Consult

   http://docs.iplanet.com/docs/manuals/fasttrak/41/ig/unix.htm

2. Remove all Web Server instances.
3. Reinstall the Web Server and all its configured instances.
4. Reinstall the Plugin

iWS 6.0 Reinstall

1. Consult

   http://docs.iplanet.com/docs/manuals/enterprise/50/ig/unix.htm

   In this case it is possible to uninstall the iPlanet Web Server Plugin for Trustbase Services without removing all Web Server Instances. When the plugin is installed on iWS 6.0 it uses the answers to the questions you give at install time to configure the plugin but it actually uses "wdeploy" to deploy the application, this means that if you with to uninstall the plugin the simplest way is to use the wdeploy delete option. The uri that the application is deployed is /DSMS so an example of the command would be

```
wdeploy delete –u /DSMS –i ragnarok.uk.sun.com –v
https-ragnarok.uk.sun.com hard
```

During installation the plugin also installs a copy of libnss3.so into the webserver's library directory, this can be replaced if you wish with the original copy that is left there – it is given the name libnss3.original.so. The library directory location is <webserver_install_dir>/bin/https/lib.

Verifying the Installation

Verifying an installation or configuration can be performed by:

1. Placing a certificate in your database store so that the Demo can find the certificate to be checked
2. Setting the classpath and running the Java servlet demo DSMSDemo.java
3. Loading the DSMS Client Web page that instigates a DSMS Certificate Status check from the Identrus root.

Setting the Classpath (iWS 4.1)

The following shell script illustrates how to set the classpath

```
#!/bin/sh
# This script is provided for convenience as an example
# of how you might set up your environment for
# This script needs to be run from its directory location
# as the paths it sets depend on it being in a given place.
PATH_TO_LS=/bin/ls
PATH_TO_CAT=/bin/cat
PATH_TO_RM=/bin/rm
IWS_SERVER_HOME=/iplanet/webserver4.1SP1.FCS/server4
CURRENTDIR=`pwd`
LD_LIBRARY_PATH=$IWS_SERVER_HOME/bin/https/lib
$PATH_TO_LS $CURRENTDIR > /tmp/liblist
for x in `$PATH_TO_CAT /tmp/liblist`
do
CLASSPATH=$CLASSPATH:$CURRENTDIR/$x
done
$PATH_TO_RM /tmp/liblist
export CLASSPATH
export LD_LIBRARY_PATH
```

Setting the Classpath (iWS 6.0)

The following illustrate how to set the classpath

```
#!/bin/sh
# This script is provided for convenience as an example
# of how you might set up your environment for
# building and running tools dependant on the dsms
# This script needs to be run from its directory location
# as the paths it sets depend on it being in a given place.
PATH_TO_LS=/bin/ls
PATH_TO_CAT=/bin/cat
PATH_TO_RM=/bin/rm
IWS_SERVER_HOME=/iplanet/webserver6.CPI/server6
CURRENTDIR=`pwd`
LD_LIBRARY_PATH=$IWS_SERVER_HOME/bin/https/lib
$PATH_TO_LS $CURRENTDIR/WEB-INF/lib > /tmp/liblist
for x in `$PATH_TO_CAT /tmp/liblist`
do
CLASSPATH=$CLASSPATH:$CURRENTDIR/WEB-INF/lib/$x
done
CLASSPATH=$CURRENTDIR/WEB-INF/classes:$CLASSPATH
$PATH_TO_RM /tmp/liblist
export CLASSPATH
export LD_LIBRARY_PATH
```

Running the Demo Servlet

The following steps need to be followed to run the demo

1. Put RPCA certificate that you wish to have checked in the database
2. Create your own version of <install_directory>/build/example/DSMSDemo.sh and add the password the certificate purpose Id of the Identrus root, the SSL sigining certificate, the root RPCA and the Certificate name as illustrated below

```
#!/bin/sh
. ./cp.sh
java com.iplanet.trustbase.identrus.dsms.DSMSDemo password Server-Cert\
Server-Cert "Identrus Root (Development)" firestormcert
```

Running the Client Demo

1. Load the Web Server by typing

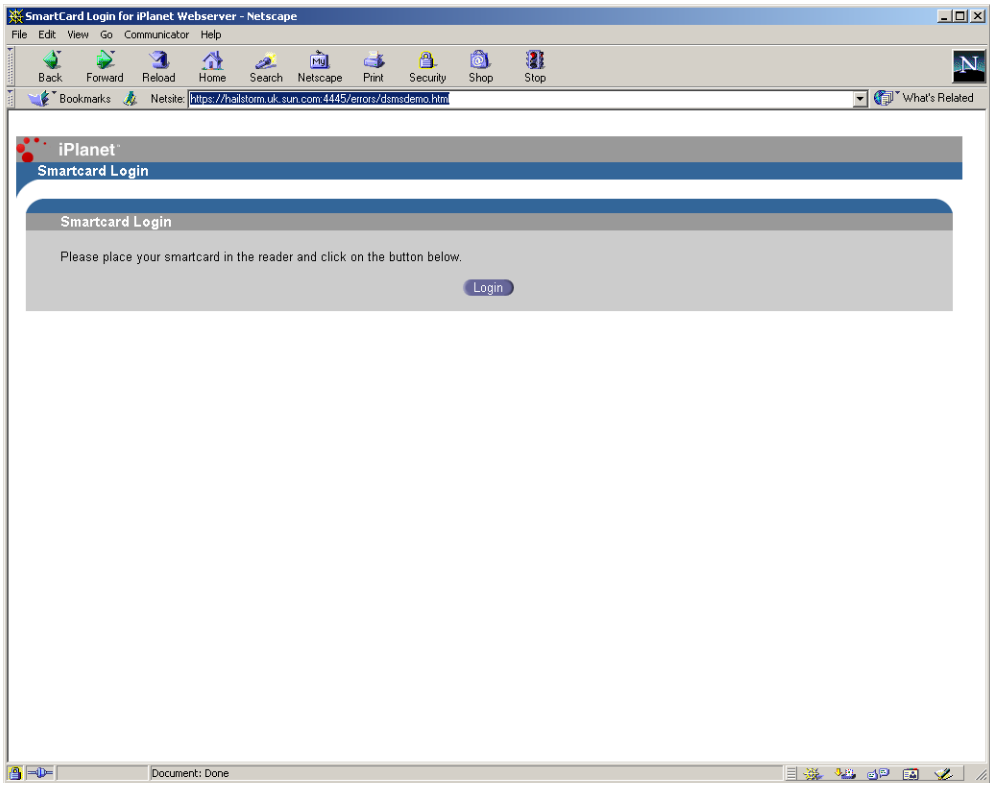
   a. iWS4.1

   http://<machine_name>:<Port>/errors/dsmsdemo.html

   b. iWS6.0

   http://machine_name>;<Port>/DSMS/errors/dsmsdemo.html

Figure 2-10 Logon to the WebServer DSMSDemo


SmartCard Login for iPlanet Webserver - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: What's Related
iPlanet
Smartcard Login
Login
Document: Done

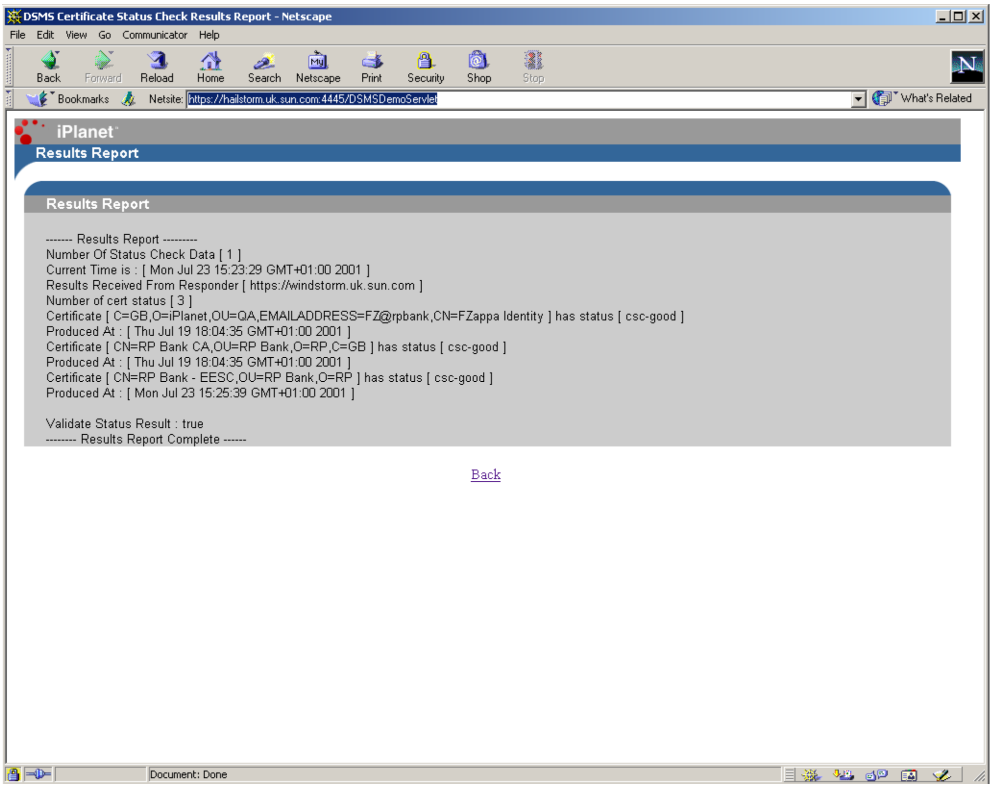
https://hailstorm.uk.sun.com:4445/errors/dsmsdemo.html

Smartcard Login

Please place your smartcard in the reader and click on the button below.

2. CSC check should come back with the following message

Figure 2-11 CSC status check completed


DSMS Certificate Status Check Results Report - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Netsite: https://hailstorm.uk.sun.com:4445/DSMSDemoServlet
What's Related
iPlanet
Results Report
Results Report
Back
Document: Done

```
------- Results Report --------
Number Of Status Check Data [ 1 ]
Current Time is : [ Mon Jul 23 15:23:29 GMT+01:00 2001 ]
Results Received From Responder [ https://windstorm.uk.sun.com ]
Number of cert status [ 3 ]
Certificate [ C=GB,O=iPlanet,OU=QA,EMAILADDRESS=FZ@rpbank,CN=FZappa Identity ] has status [ csc-good ]
Produced At : [ Thu Jul 19 18:04:35 GMT+01:00 2001 ]
Certificate [ CN=RP Bank CA,OU=RP Bank,O=RP,C=GB ] has status [ csc-good ]
Produced At : [ Thu Jul 19 18:04:35 GMT+01:00 2001 ]
Certificate [ CN=RP Bank - EESC,OU=RP Bank,O=RP ] has status [ csc-good ]
Produced At : [ Mon Jul 23 15:25:39 GMT+01:00 2001 ]

Validate Status Result : true
-------- Results Report Complete ------
```

The configuration of the Web Server insatnce that you have installed has now been verifyed.

Chapter 3

Configuration

The following must be configured

Database logging

Authservlet.properties

configservlet.properties

dsmsservlet.properties

Enable Raw Logging

Logging can be done either to Oracle or directly to a file. You can specify where your logging should take place from

/build/config/authservlet.properties

if you are using Oracle, the generation of users and the tablespaces defined may differ for individual sites - contact the site DBA for advice. The following parameters must be defined: Oracle login name, Oracle login password, PBEPassword set the same as Oracle login password, Oracle hostname, Oracle port number and Oracle SID.

Installation Pre-requisite

You should download from

JDBC^{TM} -Thin / 100% Java API for JDK^{TM} 1.1.x
`http://technet.oracle.com/software/download.htm`

The filename used might be oracle-jdbc-815.zip or classes12_01.zip depending on the version of Oracle you are using. Copy this file into the <iws_servlet_directory>. Then make sure the location of this file is appended to the classpath within jvm12.conf

Running the iPlanet Web Server: Plug-in for Trustbase services SQL Scripts

- Switch to the Oracle user and run server manager:

```
myhost> su - oracle
Password:
myhost> cd <iwsplugin_install_directory>/Build/Config
myhost> svrmgrl

Oracle Server Manager Release 3.1.5.0.0 - Production

(c) Copyright 1997, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.5.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.5.0.0 - Production

SVRMGR> connect internal
Connected.
```

- The database must be enabled to support the UTF8 character set. The following script is an example of how to achieve this.

```
SVRMGR> SHUTDOWN;
SVRMGR> STARTUP MOUNT;
SVRMGR> ALTER SYSTEM ENABLE RESTRICED SESSION;
SVRMGR> ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0;
SVRMGR> ALTER DATABASE OPEN;
SVRMGR> ALTER DATABASE CHARACTER SET UTF8;
SVRMGR> SHUTDOWN;
SVRMGR> STARTUP;
```

- Create a iPlanet Web Server: Plug-in for Trustbase services user - you may need to change the username, password and default tablespaces depending on site policy:

```
SVRMGR> CREATE USER tbase IDENTIFIED BY tbase DEFAULT TABLESPACE USERS
TEMPORARY
TABLESPACE TEMP;
Statement processed.
SVRMGR> GRANT CONNECT TO tbase;
Statement processed.
SVRMGR> GRANT RESOURCE TO tbase;
Statement processed.
SVRMGR> ALTER USER tbase QUOTA UNLIMITED ON USERS;
Statement processed.
SVRMGR> quit
Server Manager complete.
```

- Connect as the iPlanet Web Server: Plug-in for Trustbase services user and run the scripts:

```
sunstorm% su - oracle
sunstorm% cd <install_directory>/build/config
sunstorm% sqlplus
SQL*Plus: Release 8.1.5.0.0 - Production on Fri Sep 22 12:07:11 2000
(c) Copyright 1999 Oracle Corporation. All rights reserved.
Enter user-name: tbase
Enter password:
Error accessing PRODUCT_USER_PROFILE
Warning: Product user profile information not loaded!
You may need to run PUPBLD.SQL as SYSTEM

Connected to:
Oracle8i Enterprise Edition Release 8.1.5.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.5.0.0 - Production

SQL> @iwsraw.sql
```

- The following changes need to be made to authservlet.properties

```
;log.dsms.store =
com.iplanet.trustbase.identrus.iws.log.raw.PluginFileRawLogStore

log.dsms.store =
com.iplanet.trustbase.identrus.iws.log.raw.PluginJDBCRawLogStore

connection =
jdbc:oracle:thin:<username>/<password>@<oracle_instance>:1521:orcl
```

iWSPTS Configuration

The properties files, with iWS 4.1, can be found in:

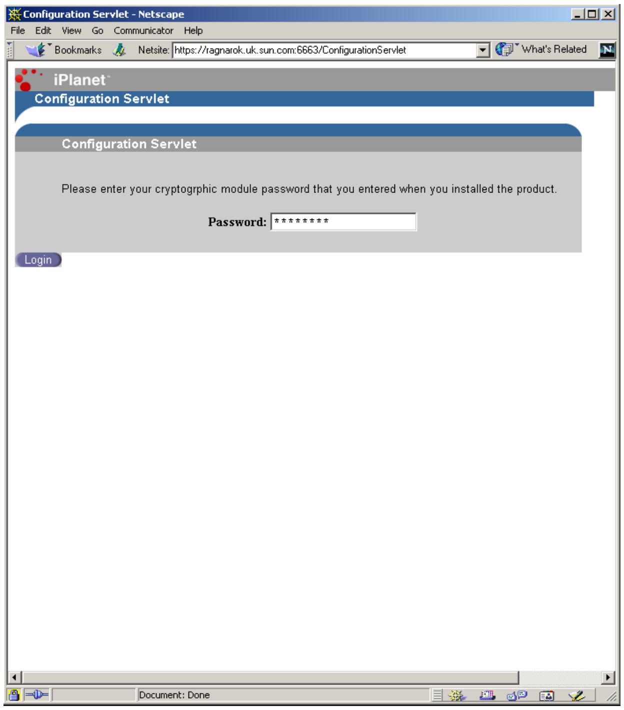
```
<iws4.1_Install_directory>/https-<instance_name>/config/Authservlet.properties
<iws4.1_Install_directory>/https-<instance_name>/config/configservlet.properties
<iws4.1_Install_directory>/https-<instance_name>/config/dsmsservlet.properties
```

The properties files, with iWS 6.0, can be found in:

```
<iwspts_Install_Directory>/config/Authservlet.properties
<iwspts_Install_Directory>/config/configservlet.properties
<iwspts_Install_Directory>/config/dsmsservlet.properties
```

and should be configured dynamically from the Web Server as follows:

Figure 3-1 Logon Main Menu



https://ragnarok.uk.sun.com:6663/ConfigurationServlet

Please enter your cryptogrphic module password that you entered when you installed the product.

Password: ********

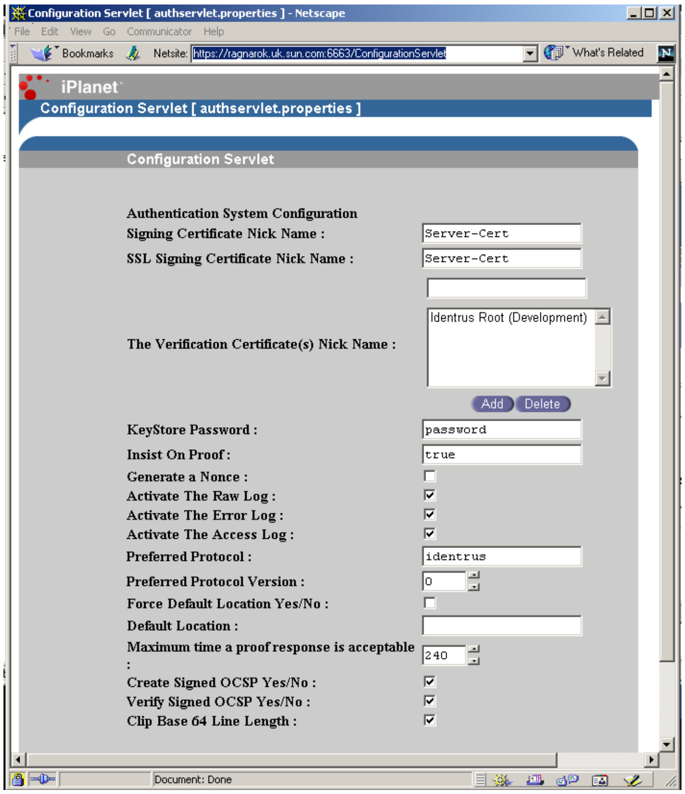
Document: Done

Configuring Authservlet.properties

Most of the configuration you can perform is centered around:

```
<iws_install_directory>/https-<iws-server-instance>/config/authserv
let.properties
```

Figure 3-2 Authentication Servlet


Configuration Servlet [authservlet.properties] - Netscape
File Edit View Go Communicator Help
Bookmarks
Netsite: https://ragnarok.uk.sun.com:6663/ConfigurationServlet
What's Related
iPlanet
Configuration Servlet [authservlet.properties]
Configuration Servlet
Add
Delete
Document: Done

Authentication System Configuration

Signing Certificate Nick Name :	Server-Cert
SSL Signing Certificate Nick Name :	Server-Cert
The Verification Certificate(s) Nick Name :	Identrus Root (Development)
KeyStore Password :	password
Insist On Proof :	true
Generate a Nonce :	☐
Activate The Raw Log :	☑
Activate The Error Log :	☑
Activate The Access Log :	☑
Preferred Protocol :	identrus
Preferred Protocol Version :	0
Force Default Location Yes/No :	☐
Default Location :	
Maximum time a proof response is acceptable :	240
Create Signed OCSP Yes/No :	☑
Verify Signed OCSP Yes/No :	☑
Clip Base 64 Line Length :	☑

This file controls the behaviour of both the authentication servlet and cookie authorisation modules. The file is divided into sections and each section controls a different element. The **Authservlet** section controls the authentication servlet as a whole entity – it controls what signing certificates and used and whether or not to log any data at all. The **cengine** section controls the cookie authorisation it controls what certificate to consider the server certificate. The **LogManager** section controls the authentication servlets logging behaviour and its configuration will be familiar to Trustbase transaction manager installers as it is exactly the same. It is used to configure the behaviour of the error , access and raw logs used by the servlet. The **CookieGenerator** section controls what cookie generators are available to the authentication servlet and how they are configured.

Database Provider

This is a module of the Authentication Servlet and it controls how a username which is entered by the user is associated with a LDAP database entry. Inside the LDAP database you must add the Base 64 encoded certificate for a user to that user's LDAP entry. The field you need to add it too by default is "usersmimecertificate" but this can be changed with "LdapDatabase.Property=CertificateElement=" property inside the DatabaseProvider section. You can control which Database Provider is chosen by using the setting in the AuthServlet section – see below.

CookieGenerator

This is a module of the Authentication Servlet an it controls what the contents of the cookie will be that is generated in response to a successful Certificate Status Check.

LogManager

Activate Logging : Controls whether or not these logs are written to

Authservlet

This section is also expected to contain any settings you wish to make to control the DSMS.

PropertyName	PropertyType	PropertyDefault	Remarks
ActivateErrorLogging	Boolean	True	Determines whether or not the authentication servlet will log any errors at all. Disabling error logging would be unwise.
ActivateAccessLogging	Boolean	True	Determines whether the authentication should log access's – an access would be a request to perform a certificate status check.
ActivateRawLogging	Boolean	True	Determines whether the authentication servlet logs the raw certificate status check data. This log can consume quite a lot of disk space quickly.
InsistOnProof	Boolean	True	Determines whether the authentication servlet trusts the End Entity it is asking to perform an Identrus CSC – Whether it will accept the absence of the end entity freshness proof.

KeyStoreType	String	Null	This determines what type of keystore is being used – for the moment this should always be set to JSS. This setting should be left alone after deploy-ment.
KeyStoreLocation	String	Null	This determines where the keystore can be found. This setting should be left alone after deployment.
KeyStorePassword	String	Null	This determines the password used to open the keystore. It should be left alone after deployment.
ReturnPage	String	errors/certificate_return.html	This setting controls what page is sent back after the initial chal-lenge. This setting should never be altered.
SubstString	String	REPLACETHISSTRING	This setting controls what strings is replaced in returned pages. This setting should never be aletered.
ErrorReturnPage	String	Errors/login-error.html	This setting controls what page is sent back when a login error occurs. This setting should never be changed.

DatabaseProvider	String	LdapDatabase	This setting controls which database provider is chosen from the available database providers.
CookieGenerator	String	CertificateCookie	This setting controls which cookie generator is chosen from the available cookie generators.
SigningCertificate	String	None	This setting controls which certificate will be used to sign requests. You should use the alias of your chosen cerrificate.
SSLSigningCertificate	String	None	This setting controls which certificate is used for SSL client communications.

Cengine

This section controls the cokie checking software the properties availbale are :

Property Name	Property Type	Default	Remarks
ServerCertNickName	String	None	The nick name of the server certificate that is being used to sign the cookies.

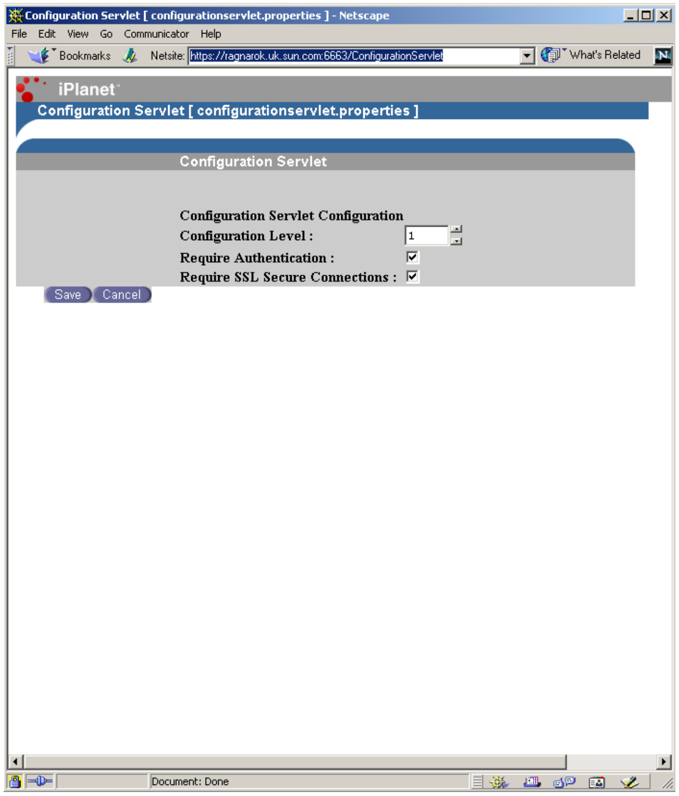
Configuring configurationservlet.properties

Configuration Level : Controls the amount of configuration options that are provided it effectively provides and beginner , medium , and advanced settings - valid values are 1 , 2 , 3 - default is 1.

Require Authentication : Controls whether the configuration servlet demands the password for the keystore when attempting to change the configuration. valid values : ticked / not ticked. default : ticked.

Require SSL Secure Connections : Controls whether the configuration servlet will accept connections through only secure connections or both secure and insecure. valid values : ticked / not ticked. Default : ticked.

Figure 3-3 Configuration Servlet

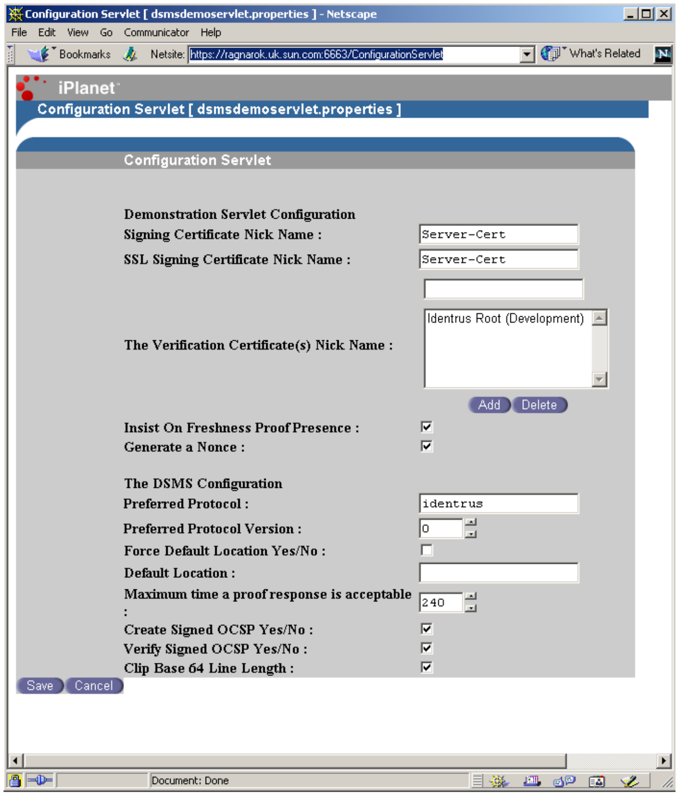

Configuration Servlet [configurationservlet.properties] - Netscape
File Edit View Go Communicator Help
Bookmarks
Netsite: https://ragnarok.uk.sun.com:6663/ConfigurationServlet
What's Related
iPlanet
Configuration Servlet [configurationservlet.properties]
Configuration Servlet
Save
Cancel
Document: Done

Configuration Servlet Configuration
Configuration Level : 1
Require Authentication : ☑
Require SSL Secure Connections : ☑

Configuring dsmsdemoservlet.properties

This file can be configured as follows:

Figure 3-4 Demonstration Servlet



Configuration Servlet [dsmsdemoservlet.properties] - Netscape

Netsite: https://ragnarok.uk.sun.com:6663/ConfigurationServlet

Configuration Servlet [dsmsdemoservlet.properties]

Configuration Servlet

Demonstration Servlet Configuration	
Signing Certificate Nick Name :	Server-Cert
SSL Signing Certificate Nick Name :	Server-Cert
The Verification Certificate(s) Nick Name :	Identrus Root (Development)
Insist On Freshness Proof Presence :	✓
Generate a Nonce :	✓
The DSMS Configuration	
Preferred Protocol :	identrus
Preferred Protocol Version :	0
Force Default Location Yes/No :	
Default Location :	
Maximum time a proof response is acceptable :	240
Create Signed OCSP Yes/No :	✓
Verify Signed OCSP Yes/No :	✓
Clip Base 64 Line Length :	✓

Document: Done

- Signing Certificate Nick Name : Is the nick name of the certificate you are using for signing your message between the authoritative entities and the webserver.
- SSL Signing certificate Nick Name : Is the nick name of the certificate you are using for SSL transactions between the authoritative entities and the webserver.

- Verification Certificate Nick Name : Are the nick names of the certificates you are using as trust anchors - to verify the signed messages you receive from authoritative entities.
- Insist On Freshness Proof Presence : Controls whether we insist on the presence of a freshness proof - and if one is not provided we will attempt to get one.
- Generate Nonce : controls whether nonce's are generated or not.
- Preferred Protocol : controls which protocol is used to perform certificate status checks - valid values : "identrus" or "ocsp". default value : "identrus".
- Protocol Version : Controls what version of the protocol we use. Default : 0. valid value : 0 , 1 , 2.
- Force Default Location : Controls whether we ignore the AIA of a certificate and send our check to the location specified in Default location.
- Maximum Time proof : Controls how long a freshness is considered valid since its generation. This value is specified in seconds.
- Create Signed OCSP : Controls whether we sign OCSP requests that we generate - this setting is overriden under the identrus protocol where they are never signed.
- Verify Signed OCSP : Controls whether we verify signed OCSP responses that we receive - this setting is overriden under the identrus protocol where they are not checked.
- clip base64 lines : controls whether the base64 generated is clipped to a line length.
- Default location: the location to send transactions to, if set to “Yes”.

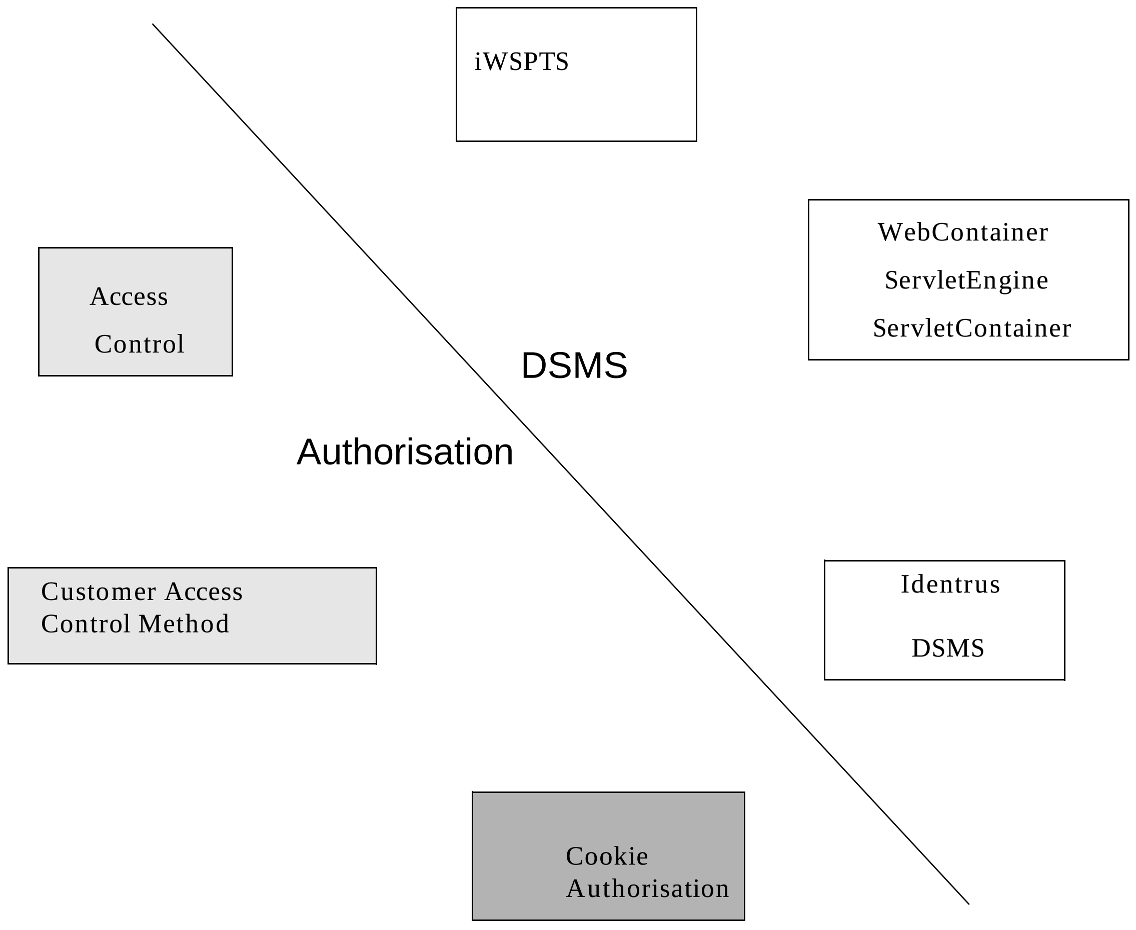
Chapter 4

Developing your own applications

Component Overview

There are number of components shipped with iWSPTS:

Figure 4-1 Component Overview



We now list the components within the Plug-in

- The cookie authorisation module passes from the results of a certificate status check to the browser in order to avoid having to repeat the same certificate status check twice.
- The Access Control API controls what data gets passed from one part of the system to another. CSCConfigAdapter and getProperty.
- The Identrus DSMS protocol allows the status check to take place

These Authorisation Components cannot change. However there are options to develop your applications using the DSMS API. Examples of how this might be used are now illustrated

- Example 1 Verification of the DSMS process
- Example 2 Modification of the Config Adapter
- Example 3 Modification of the Transport Adapter

The Source code is listed at the end of this chapter and can be found in the examples directory for iWSPTS. We now discuss how to deploy your own examples. You should also consult your JavaDocs and the API Guide that accompanies this manual.

Certificate Verification

You need to study the API interface package CSCConfigAdapter that utilises the method getProperty. DSMS behaviour can be modified by considering the following properties defined in the class:

```
com.iplanet.trustbase.initiator.PropertyCodes
```

Figure 4-2 DSMS Verification Options

Property Name	Property Type	Default	Purpose
CSC_LOCATION_FORCE_DEFAULT	Boolean	False	Determines if the DSMS will use the location specified in the AIA of its signing certificate for the OID specified in csc.prefered.oid or the default location provided.
CSC_LOCATION_DEFAULT	String	None	The hard coded location to use if csc.location.forceDefailt is set to *true*.
CSC_CREATE_SIGNED_OCSP	Boolean	False	Determines whether the DSMS will sign outgoing OCSP. This setting is not used when using XML wrapped OCSP.
CSC_VERIFY_SIGNED_OCSP	Boolean	False	Determines whether the DSMS should check OCSP that is received for correct signing. This setting is not used when using XML wrapped OCSP.
CSC_PREFERRED_OID	String	1.2.840.114021.4.1	Determines which type of transaction is carried out whether the DSMS carries out an identrus check or an OCSP check. The default setting indicates an Identrus check.
CSC_PREFERRED_VERSION	String	0	Determines which version of a protocol should be used. If “0” is given as the version then the latest version is used.
CSC_PREFERRED_PROTOCOL	String	Null	Determines the preferred protocol name to use – this takes priority over preferred.oid and is a more friendly way to specify it. Valid values are “identrus” and “ocsp”.
CSC_LOCATION_RPRESPONDER-PROXY	String	Null	Determines What the url for rpRootResponderProxy.
CSC_MAX_PROOF_RESPONSE_AGE	String	240	The length of time that a freshness proof is considered valid for. If acquire proof is true and this time limit is passed then the DSMS will make a companion request for another Signing Certificate proof.
CSC_CLIP_BASE64_LINES	Boolean	True	Determines whether the DSMS will clip base64 generated lines.

Main Steps

The main steps to perform your own certificate Status Check are now listed:

1. Initialise the Cryptographic System
2. Initialise the TokenKeyStore
3. Initialise the SSL Subsystem
4. Initialise the Config Adapter
5. Initialise the CSCEngine with the Chosen ConfigAdapter
6. Once these steps have been done a CSC check may be performed using _cscEngine.getStatus
7. The check maybe controlled by settin g the appropriate Verification Constants mentioned in the previous section

Running the DSMS Examples.

Each of the examples can be run using the CLASSPATH that is set using the provided *cp.sh* script.

Example 1 DSMS Verification

This example is an extremely simple example showing the basic steps required to make the DSMS perform a certificate status check. It leaves all the configuration at the default values except for a few essential values such as the name of the certificate to sign the outgoing identrus message. To help with making it simple it can only check the status of certificates that have keys in the database, ie those certificates that the webserver describes as ‘own’ – not the trusted certificate authorities. It takes five command line parameters which are :

- Keystore password – the password of the database you created when you installed the webserver plugin.
- Signing Certificate name – The nick name of the certificate you want to use to sign the identrus message
- SSL Signing Certificate name – the nick name of the certificate you want to use in SSL Client transactions.
- The Trust Anchor – usually the identrus root.
- Check Certificate name – The nick name of the certificate you wish to use to check the status of.

The example should output something like this.

```
ragnarok# ./dsmsdemo.sh

In PKCS11SecureRandom constructor

In PKCS11SecureRandom constructor

*** Hostname: nescafe

Checking verification chain with leaf [ CN=Identrus Root
Certification Authority,OU=Identrus Root Certification
Authority,O=Identrus LLC ]

Checking verification chain with leaf [ CN=Identrus Root
Certification Authority,OU=Identrus Root Certification
Authority,O=Identrus LLC ]

DSMS result from validity check [ true ]
```

Example 2 Config Adapter Modification

This is a slightly more complex example. It illustrates how you can manipulate the behaviour of the DSMS by creating a new ConfigAdapter. In its extreme case this ability could be used to enable the DSMS to utilise a different kind of certificate store. However in this case the new ConfigAdapter simply returns an EngineLogger object when it is asked to provide one. This EngineLogger is then used to report the DSMS's progress through the transaction. It takes the same five command line parameters that the DSMSDemo from example 1 does. The example should output something like this.

```
ragnarok# ./dsmsdemo.sh

----Engine Logger-----

Engine Logger Message [ CSCEngine initialised ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Performing a getStatus with preferred.oid [
1.2.840.114021.4.1 ] preferredVersion [ 0 ] acquireProof is [ false
] generate nonce is [ false ]  ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ ProtocolAdapterFactory : getInstance of [
1.2.840.114021.4.1 ] with version [ 0 ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ GenericProtocolAdapter : Getting location
for Protocol [ 1.2.840.114021.4.1 ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Beginning
transaction with [ https://nescafe.jcp.co.uk:1234/TC ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ GenericProtocolAdapter : Getting message
factory with classname [
com.iplanet.trustbase.initiator.scheme.dsms.identrus.message.Identr
usv2MessageFactory] ]
```

```
----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ GenericProtocolAdapter : Getting Transport
adapter with classname [
com.iplanet.trustbase.initiator.transport.XURLTransportAdapter] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Building identrus
message ]

----Engine Logger Ends----

In PKCS11SecureRandom constructor

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Created
transaction id and constructed OCSP [
QTE3NDhBRkRFMkVFQjVBODU1QUMzOEMwOEIxNkQ4QTZGMDdEMDRFNQ== ] ]

----Engine Logger Ends----

In PKCS11SecureRandom constructor

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Completed
construction of message ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ XURLTransportAdapter: sending to location [
https://nescafe.jcp.co.uk:1234/TC ] ]

----Engine Logger Ends----

*** Hostname: nescafe

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Beginning parse
of message ]

----Engine Logger Ends----

Checking verification chain with leaf [ CN=Identrus Root
Certification Authority,OU=Identrus Root Certification
Authority,O=Identrus LLC ]
```

```
Checking verification chain with leaf [ CN=Identrus Root
Certification Authority,OU=Identrus Root Certification
Authority,O=Identrus LLC ]

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Completed parse
of message ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Completed
transaction with [ https://nescafe.jcp.co.uk:1234/TC ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ DataConverterFactory : getConverter of [
identrus ]  ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Performing a validateStatus with
preferred.oid [ 1.2.840.114021.4.1 ] preferred.version [ 0 ]
acquireProof is [ false ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ ProtocolAdapterFactory : getInstance of [
1.2.840.114021.4.1 ] with version [ 0 ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Beginning status
validation ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Completed status
validation status approved ]

----Engine Logger Ends----

DSMS result from validity check [ true ]
```

Example 3 Transport Adapter Modification

This again is a slightly more complex example. It illustrates how you can manipulate the behaviour of the DSMS by providing a different Transport Adapter. In its extreme case this ability could be used to allow the DSMS to perform certificate status checks over completely different transports to the ones it supports out of the box. However in this example the new Transport Adapter just used the EngineLogger to log the data that is being sent and received. This Transport Adapter will only work over HTTP. It takes the same five command line parameters that the DSMSDemo from example 2 does with the addition of a sixth parameter that specifies the http location of the TC to perform the checks. The example should output something like this :

```
ragnarok# ./dsmsdemo.sh

----Engine Logger-----

Engine Logger Message [ CSCEngine initialised ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Performing a getStatus with preferred.oid [
1.2.840.114021.4.1 ] preferredVersion [ 0 ] acquireProof is [ false
] generate nonce is [ false ]  ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ ProtocolAdapterFactory : getInstance of [
1.2.840.114021.4.1 ] with version [ 0 ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ GenericProtocolAdapter : Getting location
for Protocol [ 1.2.840.114021.4.1 ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Beginning
transaction with [
http://nescafe.uk.sun.com/NASApp/NASAdapter/TbaseNASAdapter ] ]

----Engine Logger Ends----
```

```
----Engine Logger-----

Engine Logger Message [ GenericProtocolAdapter : Getting message
factory with classname [
com.iplanet.trustbase.initiator.scheme.dsms.identrus.message.Identr
usv2MessageFactory] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ GenericProtocolAdapter : Getting Transport
adapter with classname [
com.example.example3.ExampleTransportAdapter] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Building identrus
message ]

----Engine Logger Ends----

In PKCS11SecureRandom constructor

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Created
transaction id and constructed OCSP [
RUIwNUUxQzZDMUI5NjhFQTFCNkI3MkIyNUFDNTA2RTRDREM2QzA5Rg== ] ]

----Engine Logger Ends----

In PKCS11SecureRandom constructor

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Completed
construction of message ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ outgoing HTTP Data [ <!DOCTYPE CSCRequest
PUBLIC "-//IDENTRUS//CERTIFICATE STATUS CHECK DTD//EN"
"http://www.identrus.com/TC/2.0/CertificateStatusCheck.dtd">
<CSCRequest><NIB
id="NIB_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"
version="2.0"><ContextInfo
msggrpid="11F410055BD68EC8B4F1CC53F708764EC31D427D"
msgid="RUIwNUUxQzZDMUI5NjhFQTFCNkI3MkIyNUFDNTA2RTRDREM2QzA5Rg=="></
ContextInfo><StartTime><LocalTime
id="LocalTime_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"
```

```
time="20010921164029Z"/></StartTime><MsgTime><LocalTime
id="LocalTime_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_2"
time="20010921164029Z"/></MsgTime></NIB><Signature
xmlns="http://www.w3.org/2000/02/xmldsig#"><SignedInfo><Canonicaliz
ationMethod
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></CanonicalizationMethod><SignatureMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#rsa-sha1"></SignatureM
ethod><Reference
URI="#NIB_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"><Transforms><
Transform
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#sha1"></DigestMethod><
DigestValue>pa5Gpq6WHuykS+/nPYKxJSfnPqM=</DigestValue></Reference><
Reference
URI="#Request_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"><Transfor
ms><Transform
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#sha1"></DigestMethod><
DigestValue>86TQw4GMkpMzrETaAIIGC/0jOXY=</DigestValue></Reference><
/SignedInfo><SignatureValue>Zm7E5pZ0eCmssbq0ZN0L6sjHKm04RV0WwFSDadM
YLA4rBL1nFVMAg4JhTB7F2rLJ

It80HBGblXbUO+4sCUeqdi731+bmmMyWRxUoYjc+zDGfFZ0X/BfLFE31cNBzvYe+

IN8u+RRSvvuIR2qGfRUVzaP3LBYU9v2iW9MN8lwtxfo=</SignatureValue><KeyIn
fo><X509Data><X509IssuerSerial><X509IssuerName>C=GB,O=Identrus
LLC,OU=Development,CN=Nescafe CA
Cert</X509IssuerName><X509SerialNumber>8294</X509SerialNumber></X50
9IssuerSerial></X509Data></KeyInfo></Signature><CertBundle><X509Dat
a><X509IssuerSerial><X509IssuerName>C=GB,O=Identrus
LLC,OU=Development,CN=Nescafe CA
Cert</X509IssuerName><X509SerialNumber>8294</X509SerialNumber></X50
9IssuerSerial><X509Certificate>MIIDljCCAn6gAwIBAgICIGYwDQYJKoZIhvcN
AQEEBQAwVDELMAkGA1UEBhMCR0Ix

FTATBgNVBAoTDElkZW50cnVzIExMQzEUMBIGA1UECxMLRGV2ZWxvcG1lbnQxGDAW

BgNVBAMTD05lc2NhZmUgQ0EgQ2VydDAeFw0wMTA4MjAwOTU1NTNaFw0wMjAxMTUx

NTQyNDdaMGAxCzAJBgNVBAYTAlVLMRAwDgYDVQQKEwdJcGxhbmV0MRIwEAYDVQQL

EwlUcnVzdGJhc2UxKzApBgNVBAMTIlJhZ25hcm9rLnVrLnN1bi5jb20gdjYgU2Vy

dmVyIENlcnQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOdvJbuKFyIq2QzE

yz5EfCC1S5im5GBFC4BdfHzckH5E34jbJjX/BMEKCjMjA6Q2OqUpeZqnHxXYjwVN

jq7OOCZHzhKIFnhY8RLnk1b7PZ99BRNfVPYg5AI2KoHD1Ni1B55cJ1ALzthYnCaY
```

```
0xla0V5kQ7a9sCGNGWVeDvdGu3t1AgMBAAGjgekwgeYwEQYJYIZIAYb4QgEBBAQD
AgXgMB0GA1UdDgQWBBSvXtrTQ9nhIcEw7HEiA7fCydmWajAfBgNVHSMEGDAWgBQN
C3QrQbClHv/qzvZ9v7URsWgjQjAOBgNVHQ8BAf8EBAMCBPAwFgYDVR0RBA8wDYEL
ZXpyYUBnaGVsbG8waQYIKwYBBQUHAQEEXTBbMCoGCCsGAQUFBzABhh5odHRwOi8v
bmVzY2FmZS5qY3AuY28udWs6MjM4OS8wLQYIKoZIhvplBAGGIWh0dHBzOi8vbmVz
Y2FmZS5qY3AuY28udWs6MTIzNC9UQzANBgkqhkiG9w0BAQQFAAOCAQEAdE6duzdw
Rfs0U2n9PGyEVbypJFxPsGmSjAK/9YSQHzaURYfHrR966NExxeTXjLClTrYpd+r8
ygqHNTzduIGEvkoCpfvxLtY9ilhDKsReJsE1NQlrnjCFjheR7AXzZVRix50ixl6E
LUXbfLiASucxtqLYGSQgMIquS8ZEaos+uJOlP9oaFbMGihoVwxBMnPEoDU7iM+PZ
pTobl0nMn6QD/hSIVrG3T50VH9A2jT8+6huCeYziJqP73YtTRvEmmX97L5gTbZUr
ddUdFsKWaVH9BmDKgNE1m/r1TpjuNg+Syg8jBK48nqrSZ64DU2Zr19t09F2RNIzO
3Z3ydZTXFNpZrw==</X509Certificate></X509Data><X509Data><X509IssuerS
erial><X509IssuerName>O=Identrus LLC,OU=Identrus Root Certification
Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4101</X509SerialNumber>
</X509IssuerSerial><X509Certificate>MIIEJjCCAw6gAwIBAgICEAUwDQYJKoZ
IhvcNAQEEBQAwdzEVMBMGA1UEChMMSWRl
bnRydXMgTExDMS4wLAYDVQQLEyVJZGVudHJ1cyBSb290IENlcnRpZmljYXRpb24g
QXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1cyBSb290IENlcnRpZmljYXRpb24g
QXV0aG9yaXR5MB4XDTAxMDExNTE1NDI0N1oXDTAyMDExNTE1NDI0N1owVDELMAkG
A1UEBhMCR0IxFTATBgNVBAoTDElkZW50cnVzIExMQzEUMBIGA1UECxMLRGV2ZWxv
cG1lbnQxGDAWBgNVBAMTD05lc2NhZmUgQ0EgQ2VydDCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBANV02B1G3Xk2yhpINqGr5tiCqAA0c+NaNq5PWRj/rSZ+
OcnXy1OpZL0tcGSFvOfQT5XE+PW79xSAidk8ye/okp8pUNOys9uVp0R1O5dkB198
HR0AKb6YQIcowi8mmUITGsBnN6mwZsIbq1zXCWZ8riPCMnJciwya5C2sWYUWRj5C
KYpr6bpJFKKPPG2jxiht2zG+eDQiAEHdcrquWQ4Shu07wqa0JzbHWv9gGRNoK6VF
1IzCD5P+E/MIxMDaCNLgex4LZQBNG1+xhpnFLJ3pLY3eLjECwjbq+HzzGHp1ylyF
rh9C0JXminVxxlMbgRJ+Fxd48cZJpIcpr0tZExaCyUECAwEAAaOB3jCB2zARBglg
hkgBhvhCAQEEBAMCAAcwHQYDVR0OBBYEFA0LdCtBsKUe/+rO9n2/tRGxaCNCMB8G
A1UdIwQYMBaAFH5yUfrWfaLG383YvkYU2E8UWLKmMA8GA1UdEwEB/wQFMAMBAf8w
DgYDVR0PAQH/BAQDAgHGMGUGCCsGAQUFBwEBBFkwVzAoBggrBgEFBQcwAYYcaHR0
cDovL2tlbmNvLmpjC5jby51azoyMzg5LzArBggqhkiG+mUEAYYfaHR0cHM6Ly9r
```

```
ZW5jby5qY3AuY28udWs6MTIzNC9UQzANBgkqhkiG9w0BAQQFAAOCAQEAQ38IiCcp

VkvdiIe7Jc3rQtq/Nd3VHnwn9w2XIkofT63Lg6EXbpIOyoJ2P/9Sav5iPOqY0vKk

FlLTvxBJUyDQ8PmKWzZrVW1URUazKAOgjVMS/sTedSL8KZf8+u9UpbkBKNTkIFAs

jup9NRac+ad6vQ6Yhv0Xt/zP4OvU2P9ax5eqvYozvHKWZLJ2moppKHAEH5S39sWD

bsYmqngPxmTozyCf7BZfRcmJi+zD49x8YyvM1IU1fIjsbfyXGpjRsbN16h9ifZyP

doSwG+HkY88UegP6FvmaMYrf7xgbp9AiDviuPLOuhk2p54nULTjCVRN153H/1aD5

rcLtw3O2WLSazA==</X509Certificate></X509Data><X509Data><X509IssuerS
erial><X509IssuerName>O=Identrus LLC,OU=Identrus Root Certification
Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4096</X509SerialNumber>
</X509IssuerSerial><X509Certificate>MIID0DCCArigAwIBAgICEAAwDQYJKoZ
IhvcNAQEFBQAwdzEVMBMGA1UEChMMSWRl

bnRydXMgTExDMS4wLAYDVQQLEyVJZGVudHJ1cyBSb290IENlcnRpZmljYXRpb24g

QXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1cyBSb290IENlcnRpZmljYXRpb24g

QXV0aG9yaXR5MB4XDTAxMDEwMTAwMDAwMFoXDTEwMDExNTAwMDAwMFowdzEVMBMG

A1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGVudHJ1cyBSb290IENlcnRp

ZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1cyBSb290IENlcnRp

ZmljYXRpb24gQXV0aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC

AQEAylTeEYsHamiJt1BFXoRVcUVRNX27nYikLo0w/Hp0Ad3kSIXM+bM24cNRpVs2

TvazA+vwiG/uFr9nj7yMTXM0gUUdZMmcumufmj49+gPOiDVHlYY6y8L+WkxrXCfp

LteFunmycMd28v9DuX/I0ZZl6y0l7VapgbjpeOCTRVDWs8t20mMgdzT5aHY7C+Xo

g6wIW+i0M+kUJXb0+Wibj5gwT3ltosS8xE0O+gD/sw7muiqwy2AfyL+86S0U7p2M

TtTFPnBX/UAvsA6xpP8Zg7txfIkTQAPnP5wjD/eYNOXaR1tDs0rEY3KVrQ28kK0Q

GVg/QaeD3LArWtq0/tVs4KrMQwIDAQABo2YwZDARBglghkgBhvhCAQEEBAMCAAcw

DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUfnJR+tZ9osbfzdi+RhTYTxRYsqYw

HwYDVR0jBBgwFoAUfnJR+tZ9osbfzdi+RhTYTxRYsqYwDQYJKoZIhvcNAQEFBQAD

ggEBACio1AsoYGDskG40Fzd/BEnLzvZSSq8CUpBYJXg4U+aqI0T3cq5N8Dx0fPqk

UvhVyoPYw6igHEmV+oGgsl7HFCTP3FSOD6kptfnUkiEhWsuoquAD1kM663ukedWY

c4pgh7lRNmJeX7JHuQVoxk6q/sePIfKX1gTXWNDIDkFJAmZYsQyY1YGH5H6g2m8e

vmrjak547lB4NeAhA0cZQI5/2084jsd5Uicatqp/1auOP8E8iZtBskHqOwH1ea60

hrqTjlcKckUzHKZugfPr8kK0tDg//xB6O9ZHlEu0mZiCzuD1ehMSwfcc9SiFYoXW
```

```
Qcpo2ejb7y9DzV0QSEm9XpCFB3s=</X509Certificate></X509Data></CertBund
le><Request
id="Request_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"><RequestDat
a>MIICqjCCAqagAwIBAaFkpGIwYDELMAkGA1UEBhMCVUsxEDAOBgNVBAoTB0lwbGFu

ZXQxEjAQBgNVBAsTCVRydXN0YmFzZTErMCkGA1UEAxMiUmFnbmFyb2sudWsuc3Vu

LmNvbSB2NiBTZXJ2ZXIgQ2VydDCCAjcwggEIMDkwBwYFKw4DAhoEFPo/jtp9Jijt

Zksi7LC6frCbW6YnBBQNC3QrQbClHv/qzvZ9v7URsWgjQgICIGaggcowgccwgcQG

CSsGAQUFBzABBwSBtjCBszBUMQswCQYDVQQGEwJHQjEVMBMGA1UEChMMSWRlbnRy

dXMgTExDMRQwEgYDVQQLEwtEZXZlbG9wbWVudDEYMBYGA1UEAxMPTmVzY2FmZSBD

QSBDZXJ0MFswKgYIKwYBBQUHMAGGHmh0dHA6Ly9uZXNjYWZlLmpjcC5jby51azoy

Mzg5LzAtBggqhkiG+mUEAYYhaHR0cHM6Ly9uZXNjYWZlLmpjcC5jby51azoxMjM0

L1RDMIIBJzA5MAcGBSsOAwIaBBQMln2wYrp7My2VSzTMFvuuwz4aHQQUdYx5Pcs8

rm8H3Voo4rbhlz/Rj/0CAhAFoIHpMIHmMIHjBgkrBgEFBQcwAQcEgdUwgdIwdzEV

MBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGVudHJ1cyBSb290IENl

cnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1cyBSb290IENl

cnRpZmljYXRpb24gQXV0aG9yaXR5MFcwKAYIKwYBBQUHMAGGHGh0dHA6Ly9rZW5j

by5qY3AuY28udWs6MjM4OS8wKwYIKoZIhvplBAGGH2h0dHBzOi8va2VuY28uamNw

LmNvLnVrOjEyMzQvVEM=</RequestData></Request></CSCRequest> ]  ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ incoming HTTP Data [ <?xml version="1.0"
encoding="UTF-8"?><!DOCTYPE CSCResponse PUBLIC
"-//IDENTRUS//CERTIFICATE STATUS CHECK DTD//EN"
"http://www.identrus.com/TC/2.0/CertificateStatusCheck.dtd"><CSCRes
ponse><NIB id="NIB_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"
version="2.0"><ContextInfo
msggrpid="11F410055BD68EC8B4F1CC53F708764EC31D427D"
msgid="1001090422879"></ContextInfo><StartTime><LocalTime
id="LocalTime_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"
time="20010921164029Z"/></StartTime><MsgTime><LocalTime
id="LocalTime_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_2"
time="20010921164022Z"/></MsgTime></NIB><Signature
xmlns="http://www.w3.org/2000/02/xmldsig#"><SignedInfo><Canonicaliz
ationMethod
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></CanonicalizationMethod><SignatureMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#rsa-sha1"></SignatureM
ethod><Reference
```

```
URI="#NIB_340ED3D166D00325EFCE5CAAD31C9EF9254B8656_1"><Transforms><
Transform
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#sha1"></DigestMethod><
DigestValue>KX0xNDQYebVDgCXqwvVX/NZw9eA=</DigestValue></Reference><
Reference
URI="#Response_752D24F5C2992DEBB4C339ED026145BB33F71DD3_1"><Transfo
rms><Transform
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#sha1"></DigestMethod><
DigestValue>cfYf+FXmmsKmdbdnIlDazFZRua4=</DigestValue></Reference><
/SignedInfo><SignatureValue>GittXG8ydDSc0fGAhjOpg9+d0QUbbugcdrZfZQO
X03hwlK1O3ImwDO3o9rzddokp9jKHEm7ujBDvIUD/hWUk8/BiTjbxOLFvPO3aiUUgCw
t11Aq5f//ncWKZCx9Hk2VlLKOHr6dZ90r3Mhz/nFqGKnKCHzlvYzEJvkRaZaSDnSE=<
/SignatureValue><KeyInfo><X509Data><X509IssuerSerial><X509IssuerNam
e>O=Identrus LLC,OU=Identrus Root Certification
Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4111</X509SerialNumber>
</X509IssuerSerial></X509Data></KeyInfo></Signature><CertBundle><X5
09Data><X509IssuerSerial><X509IssuerName>O=Identrus LLC,OU=Identrus
Root Certification Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4111</X509SerialNumber>
</X509IssuerSerial><X509Certificate>MIIDqzCCApOgAwIBAgICEA8wDQYJKoZ
IhvcNAQEEBQAwdzEVMBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGVudH
J1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1c
yBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTAxMDExOTEwMTMxMloXDTAy
MDExOTEwMTMxMlowYTEaMBgGA1UEChMRaVBsYW5ldCBUcnVzdGJhc2UxFDASBgNVBAs
TC0RldmVsb3BtZW50MSAwHgYDVQQDExdOZXNjYWZlIEVFIFNpZ25pbmcgQ2VydDELMA
kGA1UEBhMCR0IwgZ4wDQYJKoZIhvcNAQEBBQADgYwAMIGIAoGAPtIp/WrdbKdKIgfk2
3w9JLt+yM2/F9sEv5oXMAIvWUB25xMuxviniE2t5TgZFk66wxmgAqlGQppENK8ygNjP
A4m+mlgxEUJK1A6AibaucxoL+23X7+QuZ7b0awaAAeFvmC462+4H1nzcI/J5y3xO+N8
2kvZdJh0IjNvgoSAeNwMCAwEAAaOB2zCB2DARBglghkgBhvhCAQEEBAMCBaAwHQYDVR
0OBBYEFHWpg/ujONid6t78XHEW/WPeUGoEMB8GA1UdIwQYMBaAFH5yUfrWfaLG383Yv
kYU2E8UWLKmMA4GA1UdDwEB/wQEAwIF4DAMBgNVHREEBTADgQFhMGUGCCsGAQUFBwEB
BFkwVzAoBggrBgEFBQcwAYYcaHR0cDovL2tlbmNvLmpjC5jby51azoyMzg5LzArBgg
qhkiG+mUEAYYfaHR0cHM6Ly9rZW5jby5qY3AuY28udWs6MTIzNC9UQzANBgkqhkiG9w
0BAQQFAAOCAQEAWuH6QwBQDyEs83DBL16im+Eu3ot2UI/1TsXl7mi9uuBR+3/4xpI53
IEikyhB4ICMRX9HySfaP2g1JNSJJyj8LMCWZAleltm3UNPojL23iiQCDO+09Zvn+M9g
mkob6wlkf1/xTMAKr/eze19zNMIvRqypzgybPQt1JIwU3KjI6SE93EzP6MqqLuFz2hO
vk+Uz7qBIFvnYKmG/x8x23/t+fC7+72/Q2ifsOftH08Thz6EL/eWXad9VYHJLdEhkTY
4r3XBS3dIZHVVPsWG4gHcfbbxfFagGRrUcBqjBVjEQmRjq2t+4LRu3nln7Sj+EIKpk5
sY8e8DHwUuaNcUPmtVzEA==</X509Certificate></X509Data><X509Data><X509
IssuerSerial><X509IssuerName>O=Identrus LLC,OU=Identrus Root
Certification Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4096</X509SerialNumber>
```

```
</X509IssuerSerial><X509Certificate>MIID0DCCArigAwIBAgICEAAwDQYJKoZ
IhvcNAQEFBQAwdzEVMBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGVudH
J1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1c
yBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTAxMDEwMTAwMDAwMFoXDTEw
MDExNTAwMDAwMFowdzEVMBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGV
udHJ1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudH
J1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFA
AOCAQ8AMIIBCgKCAQEAylTeEYsHamiJt1BFXoRVcUVRNX27nYikLo0w/Hp0Ad3kSIXM
+bM24cNRpVs2TvazA+vwiG/uFr9nj7yMTXM0gUUdZMmcumufmj49+gPOiDVHlYY6y8L
+WkxrXCfpLteFunmycMd28v9DuX/I0ZZl6y0l7VapgbjpeOCTRVDWs8t20mMgdzT5aH
Y7C+Xog6wIW+i0M+kUJXb0+Wibj5gwT3ltosS8xE0O+gD/sw7muiqwy2AfyL+86S0U7
p2MTtTFPnBX/UAvsA6xpP8Zg7txfIkTQAPnP5wjD/eYNOXaR1tDs0rEY3KVrQ28kK0Q
GVg/QaeD3LArWtq0/tVs4KrMQwIDAQABo2YwZDARBglghkgBhvhCAQEEBAMCAAcwDwY
DVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUfnJR+tZ9osbfzdi+RhTYTxRYsqYwHwYDVR
0jBBgwFoAUfnJR+tZ9osbfzdi+RhTYTxRYsqYwDQYJKoZIhvcNAQEFBQADggEBACio1
AsoYGDskG40Fzd/BEnLzvZSSq8CUpBYJXg4U+aqI0T3cq5N8Dx0fPqkUvhVyoPYw6ig
HEmV+oGgsl7HFCTP3FSOD6kptfnUkiEhWsuoquAD1kM663ukedWYc4pgh7lRNmJeX7J
HuQVoxk6q/sePIfKX1gTXWNDIDkFJAmZYsQyY1YGH5H6g2m8evmrjak547lB4NeAhA0
cZQI5/2084jsd5Uicatqp/1auOP8E8iZtBskHqOwH1ea60hrqTjlcKckUzHKZugfPr8
kK0tDg//xB6O9ZHlEu0mZiCzuD1ehMSwfcc9SiFYoXWQcpo2ejb7y9DzV0QSEm9XpCF
B3s=</X509Certificate></X509Data></CertBundle><Response
id="Response_752D24F5C2992DEBB4C339ED026145BB33F71DD3_1"><ResponseD
ata>MIIFfQoBAKCCBXYwggVyBgkrBgEFBQcwAQEEggVjMIIFXzCCARGgAwIBAaFWMFQ
xCzAJBgNVBAYTAkdCMRUwEwYDVQQKEwxJZGVudHJ1cyBMTEMxFDASBgNVBAsTC0Rldm
Vsb3BtZW50MRgwFgYDVQQDEw9OZXNjYWZlIENBIENlcnQYDzIwMDEwOTIxMTY0MDIyW
jCBoDBOMDkwBwYFKw4DAhoEFPo/jtp9JijtZksi7LC6frCbW6YnBBQNC3QrQbClHv/q
zvZ9v7URsWgjQgICIGaAABgPMjAwMTA5MjExNjMyMTlaME4wOTAHBgUrDgMCGgQUDJZ
9sGK6ezMtlUs0zBb7rsM+Gh0EFHWMeT3LPK5vB91aKOK24Zc/0Y/9AgIQBYAAGA8yMD
AxMDkyMTE2MjMxN1owDQYJKoZIhvcNAQEEBQADgYEADE9QQTcq5N22oEnR1hiafahGG
mwApCkq2gLZZf39XD+82VMd3Erc38MOCPR5o+V36Qi8wzwOj/MxtTwTzZ2uvuYdR3wz
Wd15C+PAg/zoQGqg5jTUeZwHmz5WK8V0u2CV8aigoK9I7fCXKA8FVh6f4IPPKmWjGRR
671thadarbdqgggOzMIIDrzCCA6swggKToAMCAQICAhAOMA0GCSqGSIb3DQEBBAUAMH
cxFTATBgNVBAoTDElkZW50cnVzIExMQzEuMCwGA1UECxMlSWRlbnRydXMgUm9vdCBDZ
XJ0aWZpY2F0aW9uIEF1dGhvcml0eTEuMCwGA1UEAxMlSWRlbnRydXMgUm9vdCBDZXJ0
aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0wMTAxMTkxMDEyMzhaFw0wMjAxMTkxMDEyMzh
aMGExGjAYBgNVBAoTEWlQbGFuZXQgVHJ1c3RiYXNlMRQwEgYDVQQLEwtEZXZlbG9wbW
VudDEgMB4GA1UEAxMXTmVzY2FmZSBJUCBTaWduaW5nIENlcnQxCzAJBgNVBAYTAkdCM
IGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgGUvCMy2kHckWjvuSFbyTiHI68N9QzLA
kentHGyAh6UncC4s9yhyrFodScuU7E2+P0zWnINc2YN/9Jx9pieEa9rj5yqvPuWniJi
WR1ekJHaPajAAqXlrRtanSqSaB1D2wJhnPQx9Whpp4K7Qk0GrFxVWGgrYMVVcC11uU3
o1CbFzAgMBAAGjgdswgdgwEQYJYIZIAYb4QgEBBAQDAgWgMB0GA1UdDgQWBBRNu+61f
QPXswwl0sZ6LuwVaSqrUjAfBgNVHSMEGDAWgBR+clH61n2ixt/N2L5GFNhPFFiypjAO
BgNVHQ8BAf8EBAMCBeAwDAYDVR0RBAUwA4EBYTBlBggrBgEFBQcBAQRZMFcwKAYIKwY
BBQUHMAGGHGh0dHA6Ly9rZW5jby5qY3AuY28udWs6MjM4OS8wKwYIKoZIhvplBAGGH2
h0dHBzOi8va2VuY28uamNwLmNvLnVrOjEyMzQvVEMwDQYJKoZIhvcNAQEEBQADggEBA
FsGDx1V3gNi8XxY0YgmHFt+vtEiehkVDF0+lMGaP+GKKDZkLyglEIuL14XaYege2Xt3
N3TNRK57UgI9U8Nrgtej2mepnevzhkdnaYGsjWbWPTjHDra5LTjKelO/tzh2Kyq8IuE
```

```
hU6Uq6cA3HhB3TZR1IdoLxHEQHwejuYMYlDTY6Pd0edB8b/dSrSOMHL1Gg8SJjupBOZ
f6JsBXeK2moT5mqIGn2+ljcmt6DkocE75vu9Uw9hQSr/iqgWMQjF0stUqQQ6qEhucLy
La/eHk89+Mt8PEzoqF3ZYUAVIbW7N3lfXtS13sF/10X2JjO0WtexzZuBYjxcGtAdgB2
TaMPY3c=</ResponseData><CSCResponse><NIB
id="NIB_2A02E495C2249C74E1F8750233C511D0615A2263_1"
version="2.0"><ContextInfo
msggrpid="182DFBAE53803E5B7A281BBD499804CBC7674C58"
msgid="1001089853924"></ContextInfo><StartTime><LocalTime
id="LocalTime_2A02E495C2249C74E1F8750233C511D0615A2263_1"
time="20010921163142Z"/></StartTime><MsgTime><LocalTime
id="LocalTime_2A02E495C2249C74E1F8750233C511D0615A2263_2"
time="20010921163053Z"/></MsgTime></NIB><Signature
xmlns="http://www.w3.org/2000/02/xmldsig#"><SignedInfo><Canonicaliz
ationMethod
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></CanonicalizationMethod><SignatureMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#rsa-sha1"></SignatureM
ethod><Reference
URI="#NIB_2A02E495C2249C74E1F8750233C511D0615A2263_1"><Transforms><
Transform
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#sha1"></DigestMethod><
DigestValue>TZDC1lo4dkk3mZBDu47rADVuHcg=</DigestValue></Reference><
Reference
URI="#Response_DCF2930D3B4E9046741A9F52AD89EEA5C39B20DA_1"><Transfo
rms><Transform
Algorithm="http://search.ietf.org/internet-drafts/draft-ietf-trade-
hiroshi-dom-hash-03.txt"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/02/xmldsig#sha1"></DigestMethod><
DigestValue>UvKzRQ67wgbl9ExfAkWvEaNA9fQ=</DigestValue></Reference><
/SignedInfo><SignatureValue>EqsZGlPlUmqW9nlKkcSLVxsvCzS700fyuKzXFwA
znSi3TUOev3H8Uan4TnmuS5KSqdxd0o6KUxQqVxOMe3G1MVvlV/jNBpiECj/D+nv1Id
2YsL5pjtuE40O+vyNngm+6RS8yetHipb5K/4lyvXmcQss49TXZv+Y5QDVXC924V5U=<
/SignatureValue><KeyInfo><X509Data><X509IssuerSerial><X509IssuerNam
e>O=Identrus LLC,OU=Identrus Root Certification
Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4103</X509SerialNumber>
</X509IssuerSerial></X509Data></KeyInfo></Signature><CertBundle><X5
09Data><X509IssuerSerial><X509IssuerName>O=Identrus LLC,OU=Identrus
Root Certification Authority,CN=Identrus Root Certification
Authority</X509IssuerName><X509SerialNumber>4103</X509SerialNumber>
</X509IssuerSerial><X509Certificate>MIIDvzCCAqegAwIBAgICEAcwDQYJKoZ
IhvcNAQEEBQAwdzEVMBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGVudH
J1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1c
yBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTAxMDExNjEyNTQzMloXDTAy
MDExNjEyNTQzMlowXjEdMBsGA1UEAxMUUm9vdCBJUCBTaWduaW5nIENlcnQxGjAYBgN
```

```
VBAoTEWlQbGFuZXQgVHJ1c3RiYXNlMRQwEgYDVQQLEwtEZXZlbG9wbWVudDELMAkGA1
UEBhMCR0IwgZ4wDQYJKoZIhvcNAQEBBQADgYwAMIGIAoGAZnTLq+nxMLSHkSlVw/zY7
91XiLYe9lkishAK/E6XmEjoei1gjk++FIUvZhSclF1VpURo7G7RYYzUTDv1HULEydRD
DF3I0Ack8tkZpxtHivRd9RreUIxp7ubJ90uHU4UbSFCIKIvMltfNwG6u5nEYsrfxDvv
kStCqydYZu1YjQeUCAwEAAaOB8jCB7zARBglghkgBhvhCAQEEBAMCBaAwHQYDVR0OBB
YEFKES1GXwy4zv5gykMI/VJPWVtXaFMB8GA1UdIwQYMBaAFH5yUfrWfaLG383YvkYU2
E8UWLKmMA8GCSsGAQUFBzABBQQCBQAwEwYDVR0lBAwwCgYIKwYBBQUHAwkwDQYDVR0R
BAYwBIECYXMwZQYIKwYBBQUHAQEEWTBXMCgGCCsGAQUFBzABhhxodHRwOi8va2VuY28
uamNwLmNvLnVrOjIzODkvMCsGCCqGSIb6ZQQBhh9odHRwczovL2tlbmNvLmpjcC5jby
51azoxMjM0L1RDMA0GCSqGSIb3DQEBBAUAA4IBAQA+TEf2oX3ovBXepCBAbnBViPe5a
VxweBToBiSdvlpkFF9UnS+nFuqv/Zzi66/dMN4ZxRHKChzRAshJm41cnVK0sA6XZA7g
wjghuWeMJ0M09bGqkhnRhPCC+QFnV4OrNhtBU9kv34Pdhsc6TqbO3I+SZe5MOskcn2w
D8WdpRF8HQCTEci1dw+IeYhp8C5fk1EF2R+KZaKdi6EB2fKzLc61RSOJEBpnXpyJwij
eI/cLWssZz64pGLEPo0Qac+I+XzQhc0w4IZBU+tQcOs/wwLwHQn8709Pcx2aoIgBrai
4nwCaCuky4NO7n5YFxt4hr7VO36Ont3gnQGK9uFUc7BtbXa</X509Certificate></
X509Data><X509Data><X509IssuerSerial><X509IssuerName>O=Identrus
LLC,OU=Identrus Root Certification Authority,CN=Identrus Root
Certification
Authority</X509IssuerName><X509SerialNumber>4096</X509SerialNumber>
</X509IssuerSerial><X509Certificate>MIID0DCCArigAwIBAgICEAAwDQYJKoZ
IhvcNAQEFBQAwdzEVMBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGVudH
J1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudHJ1c
yBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTAxMDEwMTAwMDAwMFoXDTEw
MDExNTAwMDAwMFowdzEVMBMGA1UEChMMSWRlbnRydXMgTExDMS4wLAYDVQQLEyVJZGV
udHJ1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MS4wLAYDVQQDEyVJZGVudH
J1cyBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFA
AOCAQ8AMIIBCgKCAQEAylTeEYsHamiJt1BFXoRVcUVRNX27nYikLo0w/Hp0Ad3kSIXM
+bM24cNRpVs2TvazA+vwiG/uFr9nj7yMTXM0gUUdZMmcumufmj49+gPOiDVHlYY6y8L
+WkxrXCfpLteFunmycMd28v9DuX/I0ZZl6y0l7VapgbjpeOCTRVDWs8t20mMgdzT5aH
Y7C+Xog6wIW+i0M+kUJXb0+Wibj5gwT3ltosS8xE0O+gD/sw7muiqwy2AfyL+86S0U7
p2MTtTFPnBX/UAvsA6xpP8Zg7txfIkTQAPnP5wjD/eYNOXaR1tDs0rEY3KVrQ28kK0Q
GVg/QaeD3LArWtq0/tVs4KrMQwIDAQABo2YwZDARBglghkgBhvhCAQEEBAMCAAcwDwY
DVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUfnJR+tZ9osbfzdi+RhTYTxRYsqYwHwYDVR
0jBBgwFoAUfnJR+tZ9osbfzdi+RhTYTxRYsqYwDQYJKoZIhvcNAQEFBQADggEBACio1
AsoYGDskG40Fzd/BEnLzvZSSq8CUpBYJXg4U+aqI0T3cq5N8Dx0fPqkUvhVyoPYw6ig
HEmV+oGgsl7HFCTP3FSOD6kptfnUkiEhWsuoquAD1kM663ukedWYc4pgh7lRNmJeX7J
HuQVoxk6q/sePIfKX1gTXWNDIDkFJAmZYsQyY1YGH5H6g2m8evmrjak547lB4NeAhA0
cZQI5/2084jsd5Uicatqp/1auOP8E8iZtBskHqOwH1ea60hrqTjlcKckUzHKZugfPr8
kK0tDg//xB6O9ZHlEu0mZiCzuD1ehMSwfcc9SiFYoXWQcpo2ejb7y9DzV0QSEm9XpCF
B3s=</X509Certificate></X509Data></CertBundle><Response
id="Response_DCF2930D3B4E9046741A9F52AD89EEA5C39B20DA_1"><ResponseD
ata>MIIBiAoBAKCCAYEwggF9BgkrBgEFBQcwAQEEggFuMIIBajCB1KFtMGsxCzAJBgN
VBAYTAnV6MQswCQYDVQQIEwJ4eDELMAkGA1UEBxMCeHgxCzAJBgNVBAoTAnh4MQswCQ
YDVQQLEwJ4eDELMAkGA1UEAxMCeHgxGzAZBgkqhkiG9w0BCQEWDGV6cmFAc3VuLmNvb
RgPMjAwMTA5MjExNjMwNTNaMFIwUDA7MAkGBSsOAwIaBQAEFAyWfbBiunszLZVLNMwW
+67DPhodBBR1jHk9yzyubwfdWijituGXP9GP/QICEA+AABgPMjAwMTA5MjExNjIzMTd
```

```
aMA0GCSqGSIb3DQEBBQUAA4GBAGz0OtGDRHIdSpP1S95DQ9zF6FhqYO+3wmhdqM0lk7
Ennk+teu+nCfmqdhSscioQYAblsAx/VVZX2xewAdaJZC0sVD23tt/4t1bLHO1f0h/oV
GZTllE2It/dK/IEfwiPV8XRnD27DwoSo02VHJ3/XhOMaK4w0mprTG1YUlslPglz</Re
sponseData></Response></CSCResponse></Response></CSCResponse> ]  ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Beginning parse
of message ]

----Engine Logger Ends----

Checking verification chain with leaf [ CN=Identrus Root
Certification Authority,OU=Identrus Root Certification
Authority,O=Identrus LLC ]

Checking verification chain with leaf [ CN=Identrus Root
Certification Authority,OU=Identrus Root Certification
Authority,O=Identrus LLC ]

----Engine Logger-----

Engine Logger Message [ Identrusv2MessageFactory : Completed parse
of message ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Completed
transaction with [
http://nescafe.uk.sun.com/NASApp/NASAdapter/TbaseNASAdapter ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ DataConverterFactory : getConverter of [
identrus ]  ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ Performing a validateStatus with
preferred.oid [ 1.2.840.114021.4.1 ] preferred.version [ 0 ]
acquireProof is [ false ] ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ ProtocolAdapterFactory : getInstance of [
1.2.840.114021.4.1 ] with version [ 0 ] ]
```

```
----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Beginning status
validation ]

----Engine Logger Ends----

----Engine Logger-----

Engine Logger Message [ IdentrusProtocolAdapter : Completed status
validation status approved ]

----Engine Logger Ends----

DSMS result from validity check [ true ]
```

Sample Source Code

The source code for each of these three examples can also be found in your examples iwstps <install_directory>. We now list some of the source code for each example. Note that the main java program DSMSDemo.java has to be modified slightly for each example.

Sample Code DSMSDemo.java

```
package com.example.example1;


import java.util.Properties;
import java.util.HashMap;
import java.util.Collection;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Date;

import java.security.cert.Certificate;
import java.security.cert.X509Certificate;

import com.iplanet.trustbase.initiator.ConfigAdapter;
import com.iplanet.trustbase.initiator.ConfigAdapterException;
import com.iplanet.trustbase.initiator.PropertyCodes;

import com.iplanet.trustbase.initiator.config.ConfigAdapterImpl;

import com.iplanet.trustbase.initiator.dsms.CSCEngine;
import com.iplanet.trustbase.initiator.dsms.StatusCheckData;
import com.iplanet.trustbase.initiator.dsms.CertStatus;

/*
```

```
* DSMSDemo :
* Is a simple demonstration application which
* takes in some command line parameters and performs
* a Certificate Status Check using the DSMS API based
* on these parameters.
* The command line parameters that are expected are :
* 1> The keystore password for opening the certificate database.
* 2> The siging certificate nick name - the nick name of the
*       certificate to use when signing DSMS transactions.
* 3> The SSL Client transaction certificate nick name. The
*       nick name of the certificate to use when performing
*       SSL Transactions.
* 4> The Trust Anchor certificate. The Identrus Root in normal
*       cases.
* 5> The nick name of the certificate to perform the status check on.
*       The certificate you check must be a KeyEntry in that it
*       must have a private key as well as a certificate this
*       guarentee's that the store will have the entire certificate
*       chain available to it and simplifies the example.
*/
public class DSMSDemo
{
        protected ConfigAdapterImpl _configAdapter;
        protected CSCEngine _cscEngine;

        public static void main ( String [] args )
                throws Exception
        {

                // check arguments before proceeding
                if ( args.length < 5 )
                {
                        System.out.println ( "usage : DSMSDemo
keystorepassword , signingCertName , SSL Certificate ,
verificationCertName , chainToCheck" );
                        System.exit ( 1 );
                }


                // set up for the check.
                DSMSDemo dsmsdemo = new DSMSDemo ( args );
               System.out.println ( "DSMS result from validity check
[ " + dsmsdemo.makeChecks ( args[4] ) + " ]" );
        }


        public DSMSDemo ( String [] args )
                throws Exception
        {
                // initiatialise the properties object
                // that will then initialise the ConfigAdapter
                Properties props = new Properties ( );
              props.put ( PropertyCodes.INITIATOR_KEYSTORE_PASSWORD
, args[0] );


                props.put (
PropertyCodes.INITIATOR_KEYSTORE_SIGNING_CERTIFICATE , args[1] );
                props.put (
PropertyCodes.INITIATOR_KEYSTORE_SSLSIGNING_CERTIFICATE , args[2] );
                props.put (
PropertyCodes.INITIATOR_KEYSTORE_VERIFICATION_CERTIFICATE + ".1" ,
args[3] );


                // initialise the config Adapter.
                _configAdapter = new ConfigAdapterImpl ( props );
                // initialise the engine.
                _cscEngine = new CSCEngine ( _configAdapter );
        }


        public boolean makeChecks ( String certificateToCheck )
                throws Exception
        {
              // perform the check and return the validation status.
                X509Certificate[] checkedCertificateChain =
getCertificateChain ( certificateToCheck );
                StatusCheckData [] sd = _cscEngine.getStatus (
checkedCertificateChain , false , (byte[]) null );
                return _cscEngine.validateStatus ( sd ,
checkedCertificateChain , false , null );
        }


        protected X509Certificate [] getCertificateChain ( String
certificateNickName )
                throws Exception
        {
                // get the certificate chain from store.
                return _configAdapter.getStore().getKeyEntry (
certificateNickName ).getCertificateChain ( );
        }


}
```

Sample Code ExampleConfigAdapter.java

```
package com.example.example2;
import java.util.*;
import com.iplanet.trustbase.initiator.ConfigAdapter;
import com.iplanet.trustbase.initiator.ConfigAdapterException;
import com.iplanet.trustbase.initiator.EngineLogger;
import com.iplanet.trustbase.initiator.PropertyCodes;
import com.iplanet.trustbase.initiator.logger.DefaultEngineLogger;
import com.iplanet.trustbase.initiator.config.ConfigAdapterImpl;
/*
* A simple subclass of the provided config adapter that
* supplies a EngineLogger class.
*/
public class ExampleConfigAdapterImpl extends ConfigAdapterImpl
{
   private EngineLogger logger;
   public ExampleConfigAdapterImpl(Properties props)
      throws ConfigAdapterException
   {
      super ( props );
      logger = new DefaultEngineLogger ( );
   }
   public Object getObjectProperty ( String property )
      throws ConfigAdapterException
   {
   if ( property.equals ( PropertyCodes.INITIATOR_ENGINE_LOGGER ) )
         return logger;
      return super.getObjectProperty ( property );
   }
   }
```

Sample Code ExampleTransportAdapter.java

```
package com.example.example3;

import java.io.IOException;
import java.io.OutputStream;
import java.io.InputStream;
import java.io.ByteArrayOutputStream;

import com.iplanet.trustbase.initiator.ConfigAdapter;
import com.iplanet.trustbase.initiator.PropertyCodes;
import com.iplanet.trustbase.initiator.TransportAdapterException;
import com.iplanet.trustbase.initiator.ConfigAdapterException;

import com.iplanet.trustbase.initiator.transport.HTTPTransportAdapter;

public class ExampleTransportAdapter extends HTTPTransportAdapter
{
      private ConfigAdapter ourconfigadapter;
      public ExampleTransportAdapter(ConfigAdapter configAdapter)
      {
            super ( configAdapter );
            ourconfigadapter = configAdapter;

      }

      public byte[] sendReceive(byte[] data, String location, String mes-
sageType) throws TransportAdapterException, ConfigAdapterException
      {
            byte[] outData = null;

            try
            {
                  EngineLogger logger = ( EngineLogger ) ourconfigada-
pter.getObjectProperty ( PropertyCodes.INITIATOR_ENGINE_LOGGER );
```

```
                    if ( logger != null )
                    {
                          logger.log ( new String ( "outgoing HTTP Data [ "
+ new String ( data ) + " ] " ) , null , null );
                    }
              }
              catch ( ConfigAdapterException cae )
              {

              }


              outData = super.sendReceive ( data , location , messageType
);
              try
              {
                    EngineLogger logger = ( EngineLogger ) ourconfigada-
pter.getObjectProperty ( PropertyCodes.INITIATOR_ENGINE_LOGGER );
                    if ( logger != null )
                    {
                          logger.log ( new String ( "incoming HTTP Data [ "
+ new String ( outData ) + " ] " ) , null , null );
                    }
              }
              catch ( ConfigAdapterException cae )
              {

              }


        }
              }
```

Glossary and References

The objectives of this chapter are to cover

- Software Platform
- Protocol
- Glossary

Software Platform

Java Development Kit 1.2.1

`http://java.sun.com/products/jdk/`

Java

`http://www.javasoft.com`

Java interface

`http://java.sun.com/products/jndi/index.html`

iPlanet Web Server

http://www.iplanet.com/products/iplanet_web_enterprise/home_2_1_1m.html

Hardware Security nCipher KeySafe 1.0 and CAFast

`http://www.ncipher.com`

Oracle 8I

http://www.oracle.com

GemSAFE IS SmartCard tool

http://www.gemplus.com

Protocol

OCSP

http://www.imc.org/ietf-pkix/

http://www.ietf.org/rfc/rfc2560.txt

SmartCard Standard

http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-11.html

Certificate requests and responses

PKCS10 requests RFC2314 can be found in http://www.ietf.org/rfc.html

PKCS7 responses RFC2315 can be found in http://www.ietf.org/rfc.html

HTML

HTML 3.2 as specified in http://www.w3.org/TR/REC-html32.html

HTTP

HTTP/1.0 or 1.1 protocol

http://www.w3.org/Protocols/rfc1945/rfc1945.txt

Identrus

http://www.identrus.com

LDAP

LDAP client protocol specified in RFC 1777 and RFC 1778, as supported by the JNDI API http://www.cis.ohio-state.edu/htbin/rfc/rfc1777.html

Glossary

3DES	Similar to DES.
AIA	Authority Information Access
Application protocol	An application protocol is a protocol that normally layers directly on top of the transport layer (e.g., TCP/IP). Examples include HTTP, TELNET, FTP, and SMTP.
ASN.1	Abstract Syntax Notation One.
Authentication	Authentication is the ability of one entity to determine the identity of another entity. i.e. in the case of NetMail Lite, you know who your email message came from.
base64	A representation of characters in digital format using a 65 character subset of U.S. ASCII.
BBS	A random number generating algorithm.
BER	Basic encoding Rules used with X509.
Block cipher	A block cipher is an algorithm that operates on plaintext in groups of bits, called blocks. 64 bits is a typical block size.
Bulk cipher	A symmetric encryption algorithm used to encrypt large quantities of data.
CA	Certificate Authority
Cipher Block Chaining Mode (CBC)	CBC is a mode in which every plaintext block encrypted with the block cipher is first eXclusive-OR-ed with the previous ciphertext block (or, in the case of the first block, with the initialisation vector).
Certificate	As part of the X.509 protocol (a.k.a. ISO Authentication framework), certificates are assigned by a trusted Certificate Authority and provide verification of a party's identity and may also supply its public key.
Certificate Authority	An organisation authorised to issue certificates (as in CA).
Client	The application entity that initiates a connection to a server.
CN	Common Name See for instance `http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html` for definition.
Connection	A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer to peer relationships. The connections are transient. Every connection is associated with one session.

CRL	Certificate Revocation List. A list of certificates that have been declared invalid by their issuing CA before their expiry dates
CSC	Certificate Status Check
Data Encryption Standard (DES)	DES is a very widely used symmetric encryption algorithm. DES is a block cipher.
DER	Distinguished Encoding rules used in X509.
DH	A public-key cryptographic algorithm for encrypting and decrypting data.
Digital Signature Standard (DSS)	A standard for digital signing, including the Digital Signing Algorithm, approved by the National Institute of Standards and Technology, defined in NIST FIPS PUB 186, "Digital Signature Standard," published May, 1994 by the U.S. Dept. of Commerce.
DSMS	Digital Signature Message System
Digital signatures	Digital signatures utilise public key cryptography and one-way hash functions to produce a signature of the data that can be authenticated, and is difficult to forge or repudiate.
DN	Distinguished Name. See for instance `http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html` or `http://search.ietf.org/internet-drafts/draft-zeilenga-ldapv3bis-rfc2253-00.txt` for definition.
DSA	Digital Signature Algorithm.
EE	End Entities are customers. i.e. the last person in the certificate chain.
Handshake	An initial negotiation between client and server that establishes the parameters of their transactions.
HSM	Hardware Security Module.
HTML	HyperText Markup Language.
IDEA	A 64-bit block cipher designed by Xuejia Lai and James Massey.
Integrity i.e.	You know your email message has not changed.
IP	Issuing Participant Bank (or other financial institution) issuing smart cards containing private keys and certificates to Subscribing Customers.
IR	Identrus Root

iWS	iPlanet Web Server
iAS	iPlanet Application Server
iWSPTS	iPlanet Web Server Plugin for Trustbase Services
key	The key used to encrypt data written by the client.
LDAP	Lightweight Directory Access Protocol
Message Authentication Code (MAC)	A Message Authentication Code is a one-way hash computed from a message and some secret data. Its purpose is to detect if the message has been altered.
MD5	MD5 is a secure hashing function that converts an arbitrarily long data stream into a digest of fixed size.
MIME	MultiPURPOSE Internet Mail Extension
Non-repudiation	A process set up to ensure that the sender cannot disavow a message
OCSP	Online Certificate Status Protocol
OU	Organisation Unit See for instance `http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html` or `http://search.ietf.org/internet-drafts/draft-zeilenga-ldapv3bis-rfc2253-00.txt` for definition.
PBE	Password based encryption
PEM	Privacy enhanced mail
Public Key Infrastructure (PKI)	Defines protocols to support online interaction.
Public key cryptography	A class of cryptographic techniques employing two-key ciphers. Messages encrypted with the public key can only be decrypted with the associated private key. Conversely, messages signed with the private key can be verified with the public key.
OSI	Open Systems Inter-Connection.
RC2, RC4	Proprietary ciphers from RSA Data Security, Inc. RC2 is block cipher and RC4 is a stream cipher.
RC	Relying Customer Party with whom the Subscribing Customer initiates a signed transaction.
RC Host	Server software that performs the role of the RC in the identrus certificate status check scheme. In the case of this document this is the Web server.

RC NetMail Lite or RC Mail	The client software interface that a customer uses to send and receive messages. In the case of this document this is NetMail Lite.
RP	Relying Participant Bank with which the Relying Customer communicates to obtain some level of trust in the signed data received from the Subscribing Customer.
RSA	A very widely used public-key algorithm that can be used for either encryption or digital signing.
Server	The server is the application entity that responds to requests for connections from clients. The server is passive, waiting for requests from clients.
SC	Subscribing Customer. Member of the Issuing Participant bank authorised to participate in Identrus activities.
Session	A SSL session is an association between a client and a server. Sessions are created by the handshake protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.
SmartCard	A hardware token that incorporates one or more integrated circuit (IC) chips to implement cryptographic functions and that possesses some inherent resistance to tampering.
SHA	The Secure Hash Algorithm is defined in FIPS PUB 180-1. It produces a 20-byte output.
SSL	Secure sockets layer
Stub	The Java interface to support communication with the CAFast hard server
TC	Transaction Co-ordinator
X509	An authentication framework based on ASN.1 BER and DER and base64

Index