

# Installation and Configuration Guide

*iPlanet Trustbase Transaction Manager*

**Version 3.0.1 Beta**

March 2002  
Rundate  
3:24, October 31, 2002

Copyright © 2002 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, Java, iPlanet, JDK, JVM, EJB, JavaBeans, HotJava, JavaScript, Java Naming and Directory Interface, Solaris, Trustbase and JDBC are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Federal Acquisitions: Commercial Software -- Government Users Subject to Standard License Terms and Conditions

This product is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun Microsystems, Inc. and its licensors, if any.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2002 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, the Sun logo, Java, iPlanet, JDK, JVM, EJB, JavaBeans, HotJava, JavaScript, Java Naming and Directory Interface, Solaris, Trustbase et JDBC logos sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autre pays.

Ce produit est soumise à des conditions de licence. Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable écrite de Sun, et de ses bailleurs de licence, s'il y en a.

DOCUMENTATION EST FOURNIE « EN L'ÉTAT », ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

# Contents

## List of Figures 9

## Introduction 13

Overall Layout .....	14
Related Documents .....	15
Software Requirements .....	16
Hardware requirements .....	17
Memory .....	17
Disk Space .....	17
Recommended Installation Template .....	18
iWS 6.0 SP2 .....	19
iAS 6.5 .....	20
iTMM 3.0.1 .....	21
iTMM Certificate Aliases .....	22
<b>Chapter 1 Installation Worksheet .....</b>	<b>23</b>
iTMM Installation Overview .....	24
iTMM Installation Procedure .....	25
SW pre-requisites .....	28
Pre-requisites .....	29
Packages included .....	29
Packages not included .....	30
Certificate Prerequisites .....	30
System Resources .....	30
iTMM prerequisite installations .....	32
iPlanet Web Server v6.0 Installation .....	32
Starting iPlanet Web Server 6.0 .....	35
Verifying iPlanet Web Server 6.0 .....	36
iWS Checklist .....	38
iPlanet Application Server v6.5 Installation .....	39

Post iPlanet Web Server and iPlanet Application Server Installation Steps .....	41
iAS Checklist .....	43
iTTM Installation Process .....	44
iTTM setup script .....	45
Silent iTTM setup script .....	46
OCSP Responder .....	57
JMS Proxy Setup .....	59
iTTM Configuration issues .....	61
Configuring the Trustbase OCSP Responder with the Certificate Authority .....	61
Configuring the iWS for URL rewriting and SSL Decryption .....	64
Installation Structure .....	71
iTTM Checklist .....	76
Post Installation Steps .....	77
Oracle Database Configuration .....	78
Running the iPlanet Trustbase Transaction Manager SQL Scripts .....	79
Oracle Example Checklist .....	81
Cryptographic Services Configuration .....	82
Cryptographic Installation Scenarios .....	82
Cryptographic Providers .....	83
PKCs11 provider with software token .....	84
PKCs11 provider with generic hardware token .....	85
PKCs11 provider with nCipher token .....	89
nCipher native provider upgrade: ittm 2.2.1 to 3.0.1 .....	92
LDAPS (Optional) .....	97
Identrus Configuration .....	98
iTTM Certificate Configuration .....	102
runCertWizard .....	103
runOCSPResponderCertWizard .....	111
Secure properties (Optional) .....	114
Installation Procedure .....	115
Start iPlanet Trustbase Transaction Manager .....	120
First Time Start Procedure .....	121
Restart procedure .....	121
Error checking .....	122
Normal Shutdown procedure .....	123
Uninstallation .....	124
Reinstallation .....	126
Identrus Authorisation .....	128
TokenKeyTool Checklist .....	133
<b>Chapter 2 Architectural Configuration .....</b>	<b>135</b>
iPlanet Application Server configuration .....	136
Using a DMZ .....	138

Machine Installations .....	140
HTTP/HTTPS Proxy Implementation .....	141
Configuring the SMTP Proxy Separately .....	145
iAS and iWS on separate machines .....	147
URL Rewrites with iAS and iWS on separate machines .....	148
LDAPS .....	149
Technical Note on LDAP .....	155
<b>Chapter 3 Logging on .....</b>	<b>157</b>
Configuration Management overview .....	158
Logon Screen .....	159
Configuration Options .....	161
<b>Chapter 4 Authorisation .....</b>	<b>163</b>
Introduction .....	164
Authorising users to access a service .....	165
Defining a role .....	165
Adding users to roles .....	167
Adding a Certificate .....	169
Mapping roles to Services .....	172
<b>Chapter 5 Logs .....</b>	<b>175</b>
Introduction .....	176
Audit log .....	177
Trustbase Audits: .....	177
Identrus Transaction Co-ordinator Audits: .....	178
Audit Configuration .....	178
Audit viewing .....	180
Raw Logging .....	182
Errors .....	183
Viewing .....	184
Configuring Error Event Types .....	186
Error Messages .....	187
Error checking .....	188
<b>Chapter 6 SMTP Proxy Configuration .....</b>	<b>191</b>
S/MIME Settings .....	192
<b>Chapter 7 Service Deployment .....</b>	<b>195</b>
Overview of Deploying a Service .....	196
Deploying .....	198

Authorisation Services .....	201
Services not requiring Authorisation .....	205
Registering your Service .....	206
<b>Chapter 8 Configuration Recovery .....</b>	<b>207</b>
Dynamic Configuration Export .....	208
Configuration Security .....	209
Dynamic Configuration Import .....	210
Data Model .....	211
Database Table Definitions .....	216
Auditdata .....	218
Auditparameters .....	219
audit_text .....	219
bill_data .....	220
cert_data .....	221
Error .....	222
error_codes .....	223
error_parameters .....	224
error_support .....	225
identrus_data .....	226
ocsp_data .....	227
ocsp_requests .....	228
ocsp_responses .....	229
raw_data .....	230
smime_transport .....	231
smtp_connection .....	232
smtp_message .....	233
What configuration data needs to be backed up? .....	234
What happens when certificates expire? .....	237
How to do Disaster Recovery? .....	238
<b>Glossary and References 239</b>	
Software Platform .....	240
Transport Protocols .....	241
Security Related Protocols .....	242
Trading Protocols .....	243
Message Protocols .....	244
Glossary .....	245
<b>Index 249</b>	

<b>Appendix A Using TokenKeyTool</b> .....	<b>253</b>
Starting TokenKeyTool .....	254
Commands for dealing with TokenKeyStoreManagers .....	255
Commands for dealing with TokenKeyStores .....	256
Utility commands .....	263
Miscellaneous commands .....	265





# List of Figures

Figure 1-1	Third Party Library jar files	28
Figure 1-2	Installing iPlanet Web Server	33
Figure 1-3	iPlanet Web Server Installed	34
Figure 1-4	iPlanet Web Server, Enterprise Edition	36
Figure 1-5	iPlanet Web Server Administration Server	37
Figure 1-6	Example iPlanet Application Server Script	40
Figure 1-7	Verifying iPlanet Application Server	41
Figure 1-8	Welcome Screen	49
Figure 1-9	License Agreement	50
Figure 1-10	User Group and account	50
Figure 1-11	Install Location	51
Figure 1-12	3rd Party Software Location	51
Figure 1-13	TokenKeyStore password	52
Figure 1-14	Database Settings	53
Figure 1-15	Database Username and Password	53
Figure 1-16	Oracle Driver Location	54
Figure 1-17	Identrus Settings	55
Figure 1-18	Installation Summary	56
Figure 1-19	OCSF Responder Wizard Welcome Page	57
Figure 1-20	OCSF Database Settings	58
Figure 1-21	Web Server Host	59
Figure 1-22	JMQ Driver location	60
Figure 1-23	Queue Server Name	60
Figure 1-24	OCSF Responders	63
Figure 1-25	Configuring URL rewrites within iWS	64
Figure 1-26	iPlanet Trustbase Transaction Manager Directory Overview	71
Figure 1-27	iPlanet Trustbase Transaction Manager Overview Directory	72
Figure 1-28	iPlanet Trustbase Transaction Manager Commonly Used Scripts	73

Figure 1-29	iPlanet Trustbase Transaction Manager Initialisation Files	74
Figure 1-30	identrus.properties file location	98
Figure 1-31	Example Identrus.properties	99
Figure 1-32	iTTM Certificate Wizard Welcome screen	103
Figure 1-33	Import Identrus Root Certificate	104
Figure 1-34	Import iTTM CA Certificate	104
Figure 1-35	Request End Entity Signing Certificate	105
Figure 1-36	End Entity Location	106
Figure 1-37	Request SSL Signing Certificate	106
Figure 1-38	SSL Location	107
Figure 1-39	Request Participant Signing Certificate	107
Figure 1-40	Participant Location	108
Figure 1-41	Import End Entity Signing Certificate	108
Figure 1-42	Import SSL Signing Certificate	109
Figure 1-43	Import Participant Signing Certificate	109
Figure 1-44	Certificate Summary	110
Figure 1-45	OCSP Certificate request	111
Figure 1-46	Save the Request as a file	112
Figure 1-47	OCSP Certificate Response	112
Figure 1-48	OCSP Install Complete	113
Figure 1-49	Securing property files (e.g. tbase.properties)	115
Figure 1-50	secure.properties	116
Figure 1-51	iAS kjs script	117
Figure 1-52	iPlanet Trustbase Transaction Manager Commonly Used Scripts	120
Figure 1-53	Installing a Certificate so as to send Identrus Messages within Trustbase	129
Figure 1-54	Installing a Certificate within Trustbase for sending Identrus Enabled Messages	130
Figure 1-55	Identrus Enabled Messages installed within Trustbase	132
Figure 2-1	DMZ Architecture	139
Figure 2-2	Standard iTTM installation	141
Figure 2-3	Proxy with single firewall	141
Figure 2-4	Proxy with two firewalls	142
Figure 2-5	Example Inbound Proxy	142
Figure 2-6	Example Outbound Proxy	143
Figure 2-7	DMZ Architecture for separating the SMTP Proxy	145
Figure 2-8	LDAPS settings	153
Figure 2-9	More LDAPS settings	154
Figure 3-1	Logon Screen	159
Figure 4-1	Authorisation Main Menu	164

Figure 4-2	List of default Roles .....	166
Figure 4-3	Add New User .....	168
Figure 4-4	Identrus PKI hierarchy .....	169
Figure 4-5	Adding a Certificate .....	171
Figure 4-6	Services to role mapping list .....	173
Figure 5-1	Logs Main Menu .....	176
Figure 5-2	Configure Audit .....	179
Figure 5-3	Audit View .....	180
Figure 5-4	Audit Results .....	181
Figure 5-5	Error Log Query .....	184
Figure 5-6	Error Log Query Results .....	185
Figure 5-7	Error Log Configuration .....	186
Figure 5-8	Selecting Error codes from your Oracle Database .....	187
Figure 7-1	Service Main Menu .....	196
Figure 7-2	Service Deployment .....	198
Figure 7-3	Example jar file of service being deployed .....	199
Figure 7-4	Service Deployment Results .....	200
Figure 7-5	Add Service .....	201
Figure 7-6	Add Role .....	202
Figure 7-7	Add User .....	203
Figure 7-8	Add Certificate .....	204
Figure 8-1	Oracle Data tables .....	211
Figure 8-2	Oracle OCSP tables .....	212
Figure 8-3	Comms .....	212
Figure 8-4	Audits .....	213
Figure 8-5	Users .....	213
Figure 8-6	Roles .....	214
Figure 8-7	Identrus .....	214
Figure 8-8	Errors .....	215
Figure 8-9	TTM table Auditdata .....	218
Figure 8-10	iTTM table Auditparameters .....	219
Figure 8-11	iTTM table audit_text .....	219
Figure 8-12	iTTM table bill_data .....	220
Figure 8-13	iTTM table cert_data .....	221
Figure 8-14	iTTM table error .....	222
Figure 8-15	iTTM table error_codes .....	223
Figure 8-16	iTTM table error_parameters .....	224

Figure 8-17	iTTM table error_support	225
Figure 8-18	iTTM table identrus_data	226
Figure 8-19	iTTM table ocsp_data	227
Figure 8-20	iTTM table ocsp_requests	228
Figure 8-21	iTTM table ocsp_responses	229
Figure 8-22	iTTM table raw_data	230
Figure 8-23	iTTM table smime_transport	231
Figure 8-24	iTTM table smtp_connection	232
Figure 8-25	iTTM table smtp_message	233

# Introduction

The following chapter discusses all related documents to this guide.

# Overall Layout

The complete documentation set comprises of:

- iTTM3.0.1-Install-Configuration-Guide.pdf (this Document) is designed for operators looking to produce applications that utilise the iPlanet Trustbase Transaction Manager framework. It is designed to provide information for operators looking to install the iPlanet Trustbase Transaction Manager platform. This guide identifies hardware and software required prior to installation, how to install iPlanet Trustbase Transaction Manager from CD-ROM
- iTTM3.0.1-Developer-Guide.pdf that indicates how to build your own services.

This manual Covers:

- Overview of the Installation
- Detailed Installation procedures
- Hardware Architectural Considerations
- Dynamic Configuration

At the end of the manual there is an appendix on how to operate your own PKI using the command line tool TokenKeyTool

## Related Documents

- **Solaris 8 and Java Development Kit**  
<http://docs.sun.com>  
<http://java.sun.com/products/jdk/1.1/docs/index.html>
- **iPlanet Application Server 6.5**  
<http://docs.sun.com/db/prod/s1.ipasee>  
<http://docs.iplanet.com/docs/manuals/ias.html>
- **iPlanet Web Server 6.0**  
<http://docs.sun.com/db/prod/s1.websrv60>  
<http://docs.iplanet.com/docs/manuals/enterprise.html>
- **iPlanet Certificate Management System**  
<http://docs.sun.com/db/prod/s1certsrv>  
<http://docs.iplanet.com/docs/manuals/cms.html>
- **Oracle 8.1.7 Installation and Configuration Guides**  
<http://www.oracle.com>
- **Hardware Security nCipher KeySafe 1.0 and CAFast**  
<http://www.ncipher.com>
- **Identrus Message Specifications**  
<http://www.identrus.com>  
**Transaction Coordinator requirements (IT-TCFUNC)**  
**Core messaging specification (IT-TCMPD)**  
**Certificate Status Check Messaging specification (IT-TCCSC)**

# Software Requirements

Solaris(TM) 8 for SPARC(TM)

JDK 1.3.1

iPlanet Web Server 6.0 SP2

iPlanet Application Server 6.5

iPlanet Trustbase(TM) Transaction Manager 3.0.1

Any JMS Provider (optional)

Oracle 8.1.7

Certificate Authority [e.g. iPlanet Certificate Management System 4.2]

Optional Hardware Security Module (HSM) on server [mandatory for Identrus participation - nCipher nShield 300 SCSI]



# Hardware requirements

## Memory

Recommended single machine setup 512 MB

## Disk Space

Recommended single machine setup 1 GB

# Recommended Installation Template

Before attempting an installation you should prepare a template outlining all your proposed settings. The following table provides the setting used throughout the documentation.

# iWS 6.0 SP2

Install directory /opt/iws6  
Would you like to continue with installation? [Yes]: Yes  
Do you agree to the license terms? [Yes]: Yes  
Choose an installation type [2]: 2  
Install location [/opt/iws6]: /opt/iws6  
Specify the components you wish to install [A] A  
Specify the components to install [1, 2, 3, 4.]: 1,2,3,4  
Computer name [myhost.mycompany.com]: myhost.mycompany.com  
System User [tbase]: tbase  
System Group [iplanet]: iplanet  
Run iWS Administration Server as [root]: root  
IWS Admin Server User Name [admin]: admin  
IWS Admin Server Password:  
IWS Admin Server Password (again):  
IWS Admin Server Port [8888]: 8888  
Web Server Port [80]: 80  
Web Server Content Root [/opt/iws6/docs]:  
Do you want to use your own JDK [No]: Yes  
JDK Directory [/usr/Java]: /usr/java1.3  
LibPath:  
JDK Classpath:

These last two questions can be answered by pressing return

## iAS 6.5

```
[0] CDROM directory /cdrom/cdrom0/ias6
[0] Install directory /opt/ias6
Would you like to continue with installation? [Yes]: Yes
Do you agree to the license terms? [No]: Yes
Select the component you want to install [1]: 1
Choose an installation type [2]: 2
Install location [/opt/ias6]: /opt/ias6
iPlanet Server Products components: Specify the components to install [All]: All
iPlanet Server Family Core: Specify the components to install [1, 2, 3]: 1,2,3
iPlanet Directory Suite components: Specify the components to install [1, 2]: 1,2
Administration Services components: Specify the components to install [1, 2]: 1,2
iPlanet Application Server Suite components: Specify the components you wish to install [1, 2, 3, 4,5]: 1,2,3,4,5
Computer name [myhost.mycompany.com]: myhost.mycompany.com
System User [tbase]:tbase
System Group [iplanet]: iplanet
Netscape configuration directory server? [No]: No
Do you want to use another directory to store your data? [No]: No
Directory server network port [389]: 389
Directory server identifier [myhost]: myhost
administrator ID [admin]: admin
Password:
Password (again):
Suffix [o=mycompany.com]: o=mycompany.com
Directory Manager DN [cn=Directory Manager]: cn=Directory Manager
Password:
Password (again):
Admin Domain [mycompany.com]: mycompany.com
Administration port [8889]: 8889
Run Administration Server as [root]: root
Product Key: XXXXXXXXXXXX-XXXXXXXXXXXX
Enter the location of your webservice: /opt/iws6/https-myhost.mycompany.com
Do you want to enable the user to access the registry and plugin libraries? [y] y
Do you want to continue with the iAS installation? [y] y
Username [admin]: admin
Password:
Password (again):
Do you want to enable I18N support for iAS? [No]: No
Username does not match [No]: Yes
```

Note patches may be required to install iAS6.5 on Solaris 8

## iTTM 3.0.1

Do you agree to the license terms? [No]: Yes

Install location [/opt/ittm]: /opt/ittm

The root location of iPlanet Application Server ? /opt/ias6

The root location of iPlanet Web Server documents directory? /opt/iws6/docs

The user to start iTTM is [ tbase ]

The group to which the Trustbase user belongs [ iplanet ]

What is the database user name that is used by iTTM? tbase

What is the database password that is used by iTTM? tbase

On what host is your database stored? mydatabase.mycompany.com

On what port is your database running? 1521

On what SID is your database? orcl

On what URL is your local OCSP responder?

http://myresponder.mycompany.com:8080/NASApp/OCSPResponder/OCSPResponderServlet

What is the AIA of this iTTM? https://myhost.mycompany.com

Enter the password to use for the tokenkeystore? password

## OCSPResponder

What is the Base DN of the LDAP Server that stores your certificate revocation list?

What is the Bind DN of the LDAP server that stores your certificate revocation list?

What is the Bind password?password

On what host is your LDAP? myhost.mycompany.com

On what port is your LDAP? 387

## JMSProxy

What Port is your ittm listening on? 80

On what host is your iTTM iPlanet Web Server ? [ myhost.uk.sun.com ]

On what HTTP port is your iTTM iPlanet Webserver running ? [ 80 ]

The JMS queue name for messages received from system backend ? [ backend\_to\_itps ]

The JMS queue server host ? [ myqueue.uk.sun.com ] { myqueue.mycompany.com }

The JMS queue server port ? [ 7676 ]

## iTTM Certificate Aliases

- (1) Location of alias = "IRCA" certificate e.g.  
"CN=Identrus Root,OU=Identrus Root,O=Identrus,C=US"
- (2) Location of alias = "L1CA" certificate e.g.  
"CN=L1 Bank CA,OU=L1 Bank,O=L1,C=GB"
- (3) Request/response for alias= "L1EESC" certificate e.g.  
"CN=L1 Bank End Entity Signing Certificate,OU=L1 Bank,O=L1"
- (4) Request/response for alias= "L1ESSL" certificate e.g.  
"CN=L1 Bank SSL CertificateCertificate,OU=L1 Bank,O=L1"
- (5) Request/response for alias= "L1IPSC" certificate e.g.  
"CN=L1 Bank Inter-Participant Signing Certificate,OU=L1 Bank,O=L1"

You should make a note of the defaults that you use as you may need them in later installs.

# Installation Worksheet

Installation Worksheet provides the user with a pre-requisite checklist of all the main features to a successful iPlanet Trustbase Transaction Manager installation. It lists software that is included, what isn't included, what must be downloaded and how it should be installed and configured.

# iTTM Installation Overview

The iTTM installation has been designed with a number of features in mind

1. iTTM itself is installed in a number of distinct modules:
  - a. The iTTM environmental packages:
    - ittm contains iTTM jar files and other binary files
    - ittm-tpri templates and routing rules for iTTM
    - ittm-sql SQL scripts for iTTM tables
    - ittm-conf iTTM Configuration files that are located in the directory /opt/ittm/myhost
    - ittm-stor contains the binaries relating to certificate store

- b. The OCSP responder packages

- ittm-ocsb contains OCSP Responder binaries
- ittm-ocsp contains the OCSP Responder Configuration options.

This is optional. iTTM supports any other RFC2560 compatible OCSP Responder.

- c. The JMS proxy packages

- ittm-jmsb binaries for jmsproxy
- ittm-jmsp configuration options for jmsproxy

Any other message Queue that supports JMS can be used here.

- d. The runCertificateOCSPResponderWizard and runCertWizard scripts that have no packages and can also be run using the command line interface tool runTokenKeyTool

2. Reinstallation is possible
3. Installing a similar installation on other machines is facilitated by editing a file that contains a list of settings.
4. Seamless upgrades and patches for future distributions are now also possible.



# iTTM Installation Procedure

The following steps need to be carried out:

1. Check/download/install/meet software pre-requisites
  - o Solaris 8
  - o JDK 1.3.1
  - o Oracle 8.1.7
  - o nCipher KeySafe 1.0
  - o Identrus Compliant Certificate Authority
  - o Optionally, a third party Identrus compliant Validation Authority can be used in place of the provided local OCSP Responder.
2. Create a <username>e.g. tbase <group> e.g. iplanet using the Unix command admintool.
3. Install from root iPlanet Web Server v6.0 SP2
4. Install from root iPlanet Application Server v6.5
5. Gather appropriate information in order to assist a silent iTTM 3.0.1 install
  - a. To ask for command line answers and place the answers to questions in the directory /outputdir that can then be edited and used to do installs on other machines.

```
./cdrom/cdrom0/ittm/setup -k /outputdir
```

- b. To gather settings from existing packages and place the answers to questions in the directory /outputdir that can then be edited and used to do installs on other machines.

```
./cdrom/cdrom0/ittm/setup -m /outputdir
```

6. Perform the actual iTTM 3.0.1 Installation using one of the following procedures
  - a. To perform a complete graphic install on the entire product
 

```
./cdrom/cdrom0/ittm/setup -g
```
  - b. To perform a complete command line install on the entire product entering answers to questions at the terminal
 

```
./cdrom/cdrom0/ittm/setup -c
```
  - c. To perform a complete command line install from a file created in /outputdir and read from /inputdir. The output will also be sent to a log file
 

```
./cdrom/cdrom0/ittm/setup -k /outputdir
./cdrom/cdrom0/ittm/setup -s /inputdir logfilename
```
  - d. To perform a complete command line install from existing package settings in order to perform a new install on a different machine created in /outputdir and read from /inputdir. The output will be sent to a log file.
 

```
./cdrom/cdrom0/ittm/setup -m /outputdir
./cdrom/cdrom0/ittm/setup -s /inputdir logfilename
```
7. Install Oracle SQL Script
 

```
/opt/ittm/current/Config/sql/tbaseNew.sql
```
8. Setup nFast Hardware and configure iPlanet Trustbase Transaction Manager to use it. There are four scripts to do this
  - o modutil PKCS11 generic hardware install
  - o ncipherP11install ncipher PKCS11 install
  - o ncipherupgrade ncipher iTTM2.2.1 to 3.0.1 upgrade
  - o installP11Library software PKCS11 install
9. Configure iWS LDAPS Security patch (Optional)
10. Configure iWS URL rewrites
11. Install and Configure certificates from the username you defined using the unix command line tool admintool
 

```
./runCertWizard or ./runtokenkeytool
./runOCSPResponderCertWizard or ./runtokenkeytool
```

12. Configure secure.properties
13. Setup site specific Identrus settings from identrus.properties
14. Start iPlanet Trustbase Transaction Manager
15. View, if necessary, Identrus Certificates and Keys using TokenKeyTool using the script

```
/opt/ittm/Scripts/runtokenkeytool
```

16. Uninstall/Backup/Restore/Patching
  - a. To Uninstall an installed package. From user: root

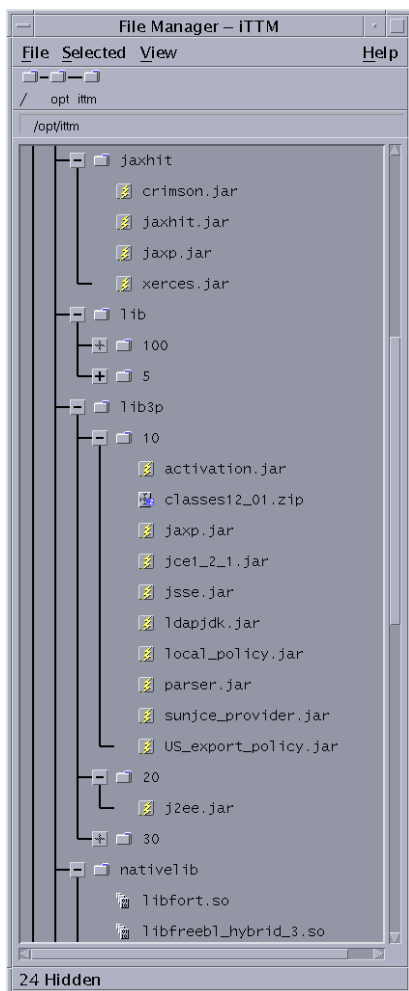
```
./cdrom/cdrom0/ittm/setup -u
```
  - b. To backup your install settings. From user: tbase
    - o ./cdrom/cdrom0/ittm/setup -m /dir
    - o cp /opt/ittm/myhost/\*.properties /temp
    - o From the iTTM configuration screen select <config><export>
    - o cp /opt/ittm/myhost/xmlconfig /temp
    - o Make sure your DBA backs up your Oracle database
  - c. To restore a similar installation. From user:tbase
    - o ./cdrom/cdrom0/ittm/setup -u
    - o ./cdrom/cdrom0/ittm/setup -s /dir
    - o cp /temp/\*.properties /opt/ittm/myhost
    - o cp -r /temp/store/\* /opt/ittm/store
    - o cp -r /temp/xmlconfig /opt/ittm/myhost
    - o From the iTTM configuration screen select <config><import>
  - d. To Add/Patch a package (For future use only). This will patch all packages it finds in the distribution directory. From user: root

```
./cdrom/cdrom0/ittm/setup -p
```

# SW pre-requisites

The following software should be installed prior to running iPlanet Trustbase Transaction Manager.

**Figure 1-1** Third Party Library jar files



## Pre-requisites

- Microsoft's Browser: Internet Explorer 4.0 or above for Web configuration or Netscape 4.0 and higher.
- Oracle 8.1.7.
- The iPlanet Trustbase Transaction Manager 3.0.1 requires the Solaris 8 Operating Environment system.
- The iPlanet Trustbase Transaction Manager 3.0.1 requires the use of JDK 1.3.1 environment. This can be obtained from the Sun Microsystems Web site <http://java.sun.com/j2se/1.3/>
- Java™ Servlet supporting WebServer
  - iPlanet Web Server 6.0 SP2  
[http://www.sun.com/software/products/web\\_srvr/home\\_web\\_srvr.html](http://www.sun.com/software/products/web_srvr/home_web_srvr.html)
- Application Server
  - iPlanet Application Server 6.5  
[http://www.sun.com/software/products/appsrvr\\_ee/home\\_appsrvr\\_ee6\\_5.html](http://www.sun.com/software/products/appsrvr_ee/home_appsrvr_ee6_5.html)
- Optionally, a third party Identrus compliant Validation Authority can be used in place of the provided local OCSP Responder.
- An Identrus Compliant Certificate Authority
- Hardware Security Module
  - nCipher KeySafe 1.0  
<http://www.ncipher.com>

## Packages included

The following packages are included with iPlanet Trustbase Transaction Manager:

- iPlanet Application Server 6.5  
[http://www.sun.com/software/products/appsrvr\\_ee/home\\_appsrvr\\_ee6\\_5.html](http://www.sun.com/software/products/appsrvr_ee/home_appsrvr_ee6_5.html)
- iPlanet Web Server 6.0 SP2  
[http://www.sun.com/software/products/web\\_srvr/home\\_web\\_srvr.html](http://www.sun.com/software/products/web_srvr/home_web_srvr.html)
- Java™ API for XML Processing  
<http://java.sun.com/xml>

## Packages not included

The following software must be downloaded before Installation:

- **JDBC™ -Thin / 100% Java API for JDK™ 1.1.x**  
<http://technet.oracle.com/software/content.html>
- **The Solaris 8 operating system environment incorporating the JDK™ 1.3.1 environment that can be downloaded from** <http://java.sun.com/j2se/1.3/>

## Certificate Prerequisites

Before you install iPlanet Trustbase Transaction Manager, you will need to know the location of your level 1 Certificate Authority PEM encoded CA certificate. This certificate is referred to from this point forward as the L1CA certificate.

## System Resources

- Determine hostname

```
hostname  
domainname
```

- Verify disk space to be greater than 1GB.

```
df -k
```

- Verify sufficient memory is not less than 256MB and preferably the recommended 1GB

```
prtconf | grep Mem
```

- Check Solaris version is 8

```
uname -a
```

- Check Java Version is 1.3.1

```
java -version
```

# iTTM prerequisite installations

Before iPlanet Trustbase Transaction Manager can be installed, a working web server and application server must be available.

## iPlanet Web Server v6.0 Installation

- Before starting the webserver installation you may need to use the Unix tool `Admintool` to create a new user. During the default installation we use  
`User: tbase`  
`Group ID: iplanet`
- Logon as root, locate the 'setup' script depending on distribution
- Start the installation using the 'setup' script.
- Select a directory `<install_directory>`. The default is `/opt/iws6`. Make a note of this directory as you will need this for the iPlanet Application Server installation later.

```
# cd /cdrom/cdrom0
# cd iws6
# ./setup
```

- Install all elements of the web server and select the option that changes their ownership/group to 'nobody'. Select the option that runs the webserver as 'root'. Do not use an existing Directory service. Select the web server document directory to be the 'docs' subdirectory of your chosen installation location. Make sure you select Yes when asked about which version of JDK you are using.



**Figure 1-2** Installing iPlanet Web Server

```
Would you like to continue with installation? [Yes]: Yes
Do you agree to the license terms? [Yes]: Yes
Choose an installation type [2]: 2
Install location [/usr/iplanet/servers]: /opt/iws6
Specify the components you wish to install [A] A
Specify the components to install [1, 2, 3, 4]: 1,2,3,4
Computer name [myhost.mycompany.com]: myhost.mycompany.com
System User [tbase]: tbase
System Group [iplanet]: iplanet
Run iWS Administration Server as [root]: root
IWS Admin Server User Name [admin]: admin
IWS Admin Server Password:
IWS Admin Server Password (again):
IWS Admin Server Port [8888]: 8888
Web Server Port [80]: 80
Web Server Content Root [/opt/iws6/docs]:
Do you want to use your own JDK [No]: Yes
JDK Directory [/usr/Java]: /usr/java1.3
```

---

**NOTE** You should make a note of these settings. Particularly port numbers since you may need them later.

If you want to set up an SSL within the Web Server you must apply for a certificate in the Web Server. For instance,

<http://docs.iplanet.com/docs/manuals/enterprise/50/ag/eseccerty.htm#1005553>

---

- The following output should appear:

**Figure 1-3** iPlanet Web Server Installed

```

                                Sun Netscape Alliance
                                iPlanet Web Server Installation/Uninstallation
-----

Extracting Server Core...
Extracting Java Runtime Environment...
Extracting Java Support...
Extracting SSJS Support...
Extracting SSJS Database Support...
Extracting Web Publishing Support...
Extracting SNMP Support...
Extracting Upgrade Files...

Server Core installed successfully.
Java Runtime Environment installed successfully.
Java Support installed successfully.
SSJS Support installed successfully.
SSJS Database Support installed successfully.
Web Publishing Support installed successfully.
SNMP Support installed successfully.

Press Return to continue...

Go to /opt/iws6 and type startconsole to begin
Managing your servers.
```

---

**NOTE** Please consult your iPlanet Web Server Installation Guide for more information on this. Selecting <return> takes the default.

---

## Starting iPlanet Web Server 6.0

Change into the directory that contains your Web Server instance and run the start script. This directory takes the form of `https-myhost.mycompany.com`.

For instance:

```
cd /opt/iws6/https-myhost.mycompany.com
./start
```

---

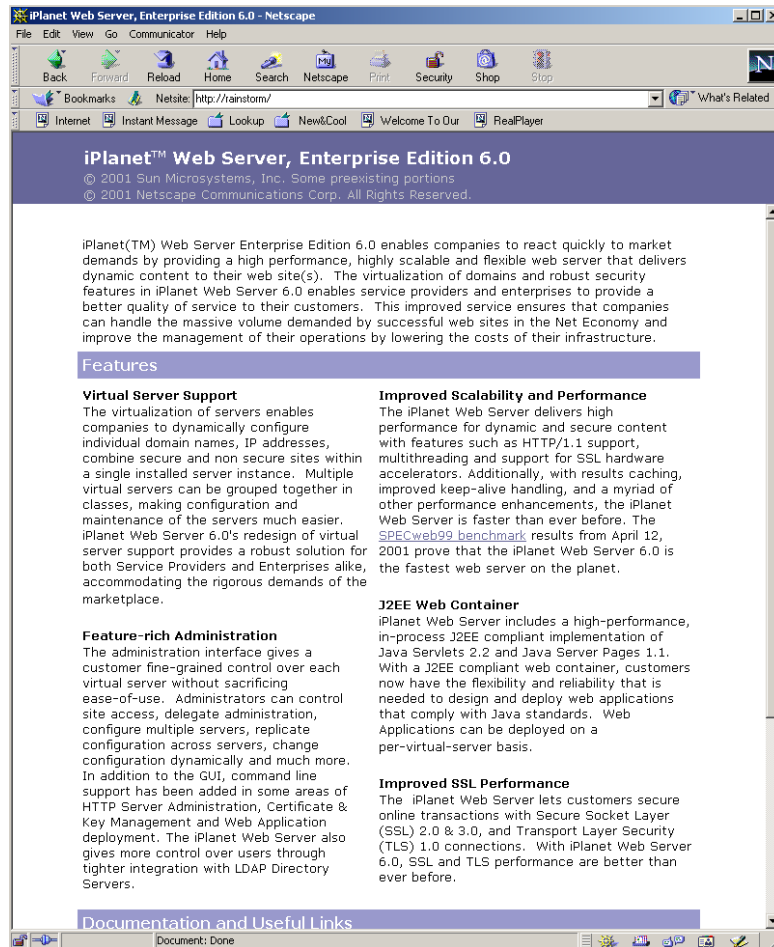
**NOTE** If in doubt you should consult the first chapter of the iPlanet Web Server configuration Guide.

---

## Verifying iPlanet Web Server 6.0

Check iPlanet Web Server is working by opening a browser window to <http://localhost>. If its working, you'll see the main page as illustrated below:

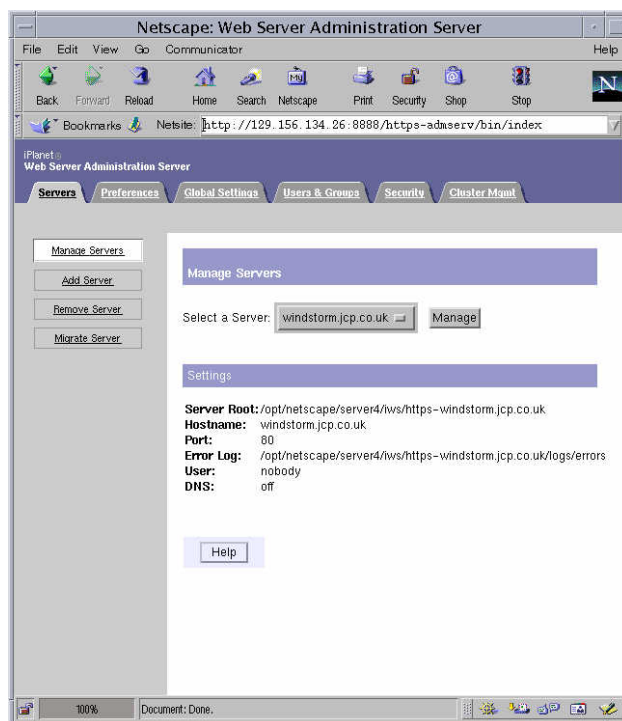
**Figure 1-4** iPlanet Web Server, Enterprise Edition



Or alternatively going straight to the Administration Server console as follows:

```
cd /opt/iws6  
./startconsole
```

**Figure 1-5** iPlanet Web Server Administration Server



## iWS Checklist

The following table summarises typical operational commands for the iPlanet Web Server.

Information Type	Example Set-up Value for iWS6.0
Install directory	/opt/iws6
Administration logon	Username: iwsadmin, Password: identrus
Operational ports	Server: 80, Admin: 8888
To start server	/opt/iws6/https-myhost.mycompany.com/start
To stop server	/opt/iws6/https-myhost.mycompany.com/stop
To start admin server	/opt/iws6/https-admin/start
To stop admin server	/opt/iws6/https-admin/stop
Processes grep	ps -ef   grep iws
Process list	nobody 9876 1 0 12:52:08 0:00 ./uxwdog -d /opt/iws6/https-myhost.mycompany.com/config nobody 9877 9876 0 12:52:08 0:01 ns-httpd -d /opt/iws6/https-myhost.mycompany.com/config also /opt/iws6/https-admin/config if the admin is running
Install logs	/opt/iws6/setup/WebServer/
Log directory	/opt/iws6/https-myhost.mycompany.com/logs
Document root	/opt/iws6/docs
Installation and Configuration Documents	http://docs.sun.com/db/prod/s1.websrv60 http://docs.iplanet.com/docs/manuals/enterprise/50/ig/contents.htm http://docs.iplanet.com/docs/manuals/enterprise/50/ag/contents.htm

## iPlanet Application Server v6.5 Installation

- Ensure that the iPlanet Web Server is running before you follow this procedure.
- Logon as root, locate the 'setup' script depending on distribution (e.g. unzip the tar file and the 'setup' file should be in the top directory)

```
# cd /cdrom/cdrom0
# cd ias6
# ./setup
```

- Warning the iAS 6.5 installation may require patches to your Solaris installation as such you should consult  
<http://docs.sun.com/source/816-6373-10/index.html>  
<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access>
- Start the installation using the 'setup' script. The default installation should be set to /opt/ias6. Answer the questions as follows:

**Figure 1-6** Example iPlanet Application Server Script

```

Would you like to continue with installation? [Yes]: Yes
Do you agree to the license terms? [No]: Yes
Select the component you want to install [1]: 1
Choose an installation type [2]: 2
Install location [/usr/iplanet/ias6]: /opt/ias6
iPlanet Server Products components: Specify the components to install [All]: All
iPlanet Server Family Core: Specify the components to install [1, 2, 3]: 1,2,3
iPlanet Directory Suite components: Specify the components to install [1, 2]: 1,2
Administration Services components: Specify the components to install [1, 2]: 1,2
iPlanet Application Server Suite components: Specify the components you wish to install [1, 2, 3,
4,5]: 1,2,3,4,5
Computer name [myhost.mycompany.com]: myhost.mycompany.com
System User [tbase]: tbase
System Group [iplanet]: iplanet
Netscape configuration directory server? [No]: No
Do you want to use another directory to store your data? [No]: No
Directory server network port [389]: 389
Directory server identifier [myhost]: myhost
administrator ID [admin]: admin
Password:
Password (again):
Suffix [dc=uk,dc=sun,dc=com]: dc=uk,dc=sun,dc=com
Directory Manager DN [cn=Directory Manager]: cn=Directory Manager
Password:
Password (again):
Admin Domain [iplanet.com]: iplanet.com
Administration port [12816]: 12816
Run Administration Server as [root]: root
Product Key: 1111111111-3333333333
Enter the location of your webserver: /opt/iws6/https myhost.mycompany.com
Do you want to enable the user to access the registry and plugin libraries? [y] y
Do you want to continue with the iAS installation? [y] y
Username [admin]: admin
Password:
Password (again):
Do you want to enable I18N support for iAS? [No]: No
Username does not match [No]: Yes
Do you want to change ownership pf iasfiles to tbase: [tbase]?

```

---

**NOTE** Please consult your iPlanet Application Server Installation Guide for more information on this. You should also make a note of these settings since you may need them later. Selecting <return> takes the default.

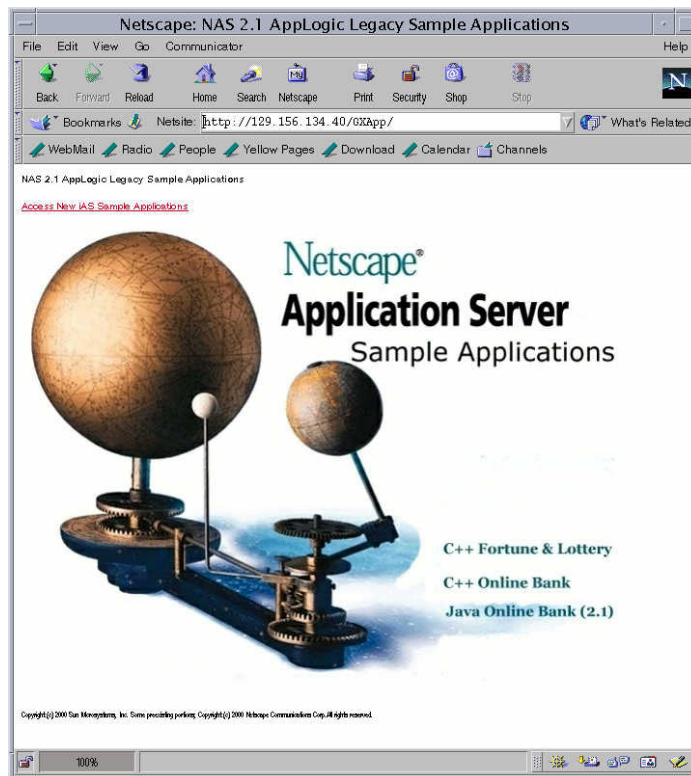
---



## Post iPlanet Web Server and iPlanet Application Server Installation Steps

- To check that the Application Server installation has been successful, use a browser to contact the following URL:  
<http://myhost.mycompany.com/GXApp>  
 Select the 'Java Fortune' application, if the reply indicates that the servlet 'Greets You' then the web server and application server are functioning correctly. You should also consult your iPlanet Application Server Installation Documentation. This is now illustrated below:

**Figure 1-7** Verifying iPlanet Application Server



- Having completed the installation, all processes must be shutdown:
- Logon as root

- **Application Server Shutdown** `<install_directory>/ias/bin/KIVaes.sh stop`

```
# cd /opt/ias6/ias/bin
# ./KIVaes.sh stop
# ps -ef | grep k.s
```

---

**NOTE** Immediately after installation, if this script fails to work, use 'ps -ef | grep k.s' to show all iPlanet Application Server processes and then 'kill -9 <pid>' to stop all of them.

---

- **Directory Server Shutdown**

```
# cd /opt/ias6/slaped-myhost
# ./stop-slaped
# ps -ef | grep slap
```

- **Web Server Shutdown**

```
# ./opt/iws6/https-myhost.mycompany.com/stop
# ./opt/iws6/https-admserv/stop
```

- Once all the above scripts have been run check that all processes have been terminated using 'ps -ef | grep /opt/iws6'. Use 'kill -9 <pid>' on all remaining processes.

## iAS Checklist

The following table summarises typical operational commands for the iPlanet Web Server.

Information Type	Example Set-up Value for iAS6.5
Install directory	/opt/ias6
Administration logon	Username: admin, Password: password
Operational ports	Directory Admin: 20000, kas admin:10817, Directory server: 389
To start server	/opt/ittm/Scripts/startias
To stop server	/opt/ittm/Scripts/stopias
Installation logs	/opt/ias6/setup/setup.log
Processes grep	ps -ef   grep ias To get just the 'kiva' processes (the ones that do the jvm work) do a ps -ef   grep k.s
Process list	root 10066 10064 0 14:33:21 0:03 /opt/ias6/ias/bin/.kjs -cset CCS0 root 10059 9504 0 14:33:16 pts/6 0:00 /opt/ias6/ias/bin/.kas root 9504 1 0 12:47:38 pts/6 0:00 /bin/sh /opt/ias6/ias/bin/kas root 10070 1 0 14:33:25 0:00 /bin/sh /opt/ias6/ias/bin/kcs -cse t CCS0 -eng 2 root 10064 1 0 14:33:21 ? 0:00 /bin/sh /opt/ias6/ias/bin/kjs -cset CCS0 -eng 1 root 10061 1 0 14:33:19 ? 0:00 /bin/sh /opt/ias6/ias/bin/kxs -cset CCS0 -eng 0 root 10072 10070 0 14:33:25 ? 0:00 /opt/ias6/ias/bin/.kcs -cset CCS0 -eng 2 root 10062 10061 0 14:33:19 ? 0:01 /opt/ias6/ias/bin/.kxs -cset CCS0 -eng 0 nobody 8174 1 0 12:45:04 ? 0:04 ./ns-slapd -f /opt/ias6/slapd-unix d02/config/slapd.conf -i /opt/ias6/slapd-myhost.mycompany.com
Logged processes	kxs_0_CCS0: Contains information about the incoming message and the plugin start and stop kjs_0_CCS0: Contains the standard out from any running java process – can contain some debug information.
Installation Document	<a href="http://docs.sun.com/source/816-5788-10/index.html">http://docs.sun.com/source/816-5788-10/index.html</a>

## iTTM Installation Process

Now that iPlanet Web Server and iPlanet Application Server are installed the iPlanet Trustbase Transaction Manager installation can proceed.

The installation is provided on the CDROM. The installation program is designed for Unix Solaris 8. iPlanet Trustbase Transaction Manager currently requires JDK 1.3.1 to be installed on your machine to operate correctly. To ensure iPlanet Trustbase Transaction Manager is installed correctly it is advised that you install JDK 1.3.1 before iPlanet Trustbase Transaction Manager. Allow 100MB of Disc space for JDK 1.3.1 and iPlanet Trustbase Transaction Manager.

- Logon as root
- Make sure the directory server for iPlanet Application Server is running before you install Trustbase. For example,

```
cd /opt/ias6/slaped-myhost
./start-slaped
```

## iTTM setup script

The iTTM setup script has the following options:

- -g Performs a complete graphic install
- -s Operates on Command line answers silently
- -c Performs complete command line install interactively
- -k Asks for command line answers
- -m Gathers settings from installed packages
- -p Reinstalls, adds or patches a package
- -u Uninstalls an installed package

Before running the setup script in silent mode you will need to:

1. Make a change to silent installer default setting in

```
/var/sadm/install/admin/default
```

Make sure the following setting is set.

```
"action" ="nocheck"
```

2. Make sure any /outputdir is empty otherwise when you use it for an inputdir it will select the previous settings as well.
3. All Admin settings can be found in the same directory as your setup script by typing the following command:

```
ls -a /cdrom/cdrom0/ittm/*
```

4. Setting retrieval wont function if secure.properties settings have been made as such you should undo the settings you made with secure.properties.
5. When using silent install and editing your save settings (using the -m option), the following rules apply:

(a) When reinstalling from a previous installation on the same machine you cannot edit and change the settings you saved.

(b) The settings you use within the silent install option -s must also be persistently correct.

(c) You can however edit the settings you saved (with the -m option) when silent installing (with the -s option) on a new machine.

## Silent iTTM setup script

Silent installs are possible using combinations of these commands.

1. To gather appropriate information in order to assist a silent iTTM 3.0.1 install
  - a. To ask for command line answers and place the answers to questions in the directory `/outputdir` that can then be edited and used to do installs on other machines.

```
./cdrom/cdrom0/ittm/setup -k /outputdir
```

If you desire to use non-default values for the install directory and or username and groupname, you must alter the files that are generated from your answers. Files with your answers stored in them are located underneath the directory you nominated when using the `setup -k` option. There is one directory for each of the components you are installing. You must check the contents of all the files located within these directories, making sure that the values for `BASEDIR`, `Owner` and `Group` are what you wish them to be. Where "`BASEDIR`" is the value for the install directory, "`Owner`" is the value of the user you wish to run the component that you are installing, and "`Group`" is the value of that users group.

- b. To gather settings from existing packages and place the answers to questions in the directory `/outputdir` that can then be edited and used to do installs on other machines.

```
./cdrom/cdrom0/ittm/setup -m /outputdir
```

**2. Perform the actual iTTM 3.0.1 Installation using one of the following procedures**

- a. To perform a complete graphic install on the entire product**

```
./cdrom/cdrom0/ittm/setup -g
```

- b. To perform a complete command line install on the entire product entering answers to questions at the terminal**

```
./cdrom/cdrom0/ittm/setup -c
```

- c. To perform a complete command line install from a file created in /outputdir and read from /inputdir. The output will also be sent to a log file**

```
./cdrom/cdrom0/ittm/setup -k /outputdir
```

```
./cdrom/cdrom0/ittm/setup -s /inputdir logfilename
```

- d. To perform a complete command line install from existing package settings in order to perform a new install on a different machine created in /outputdir and read from /inputdir. The output will be sent to a log file.**

```
./cdrom/cdrom0/ittm/setup -m /outputdir
```

```
./cdrom/cdrom0/ittm/setup -s /inputdir logfilename
```

Before continuing you will need to create an Oracle user and as such you should follow the instructions on how to create a user described in “Oracle Database Configuration,” on page 78.

We now illustrate a complete graphic install type, logged in under root, the following command

```
# ./cdrom/cdrom0/ittm/setup -g
```

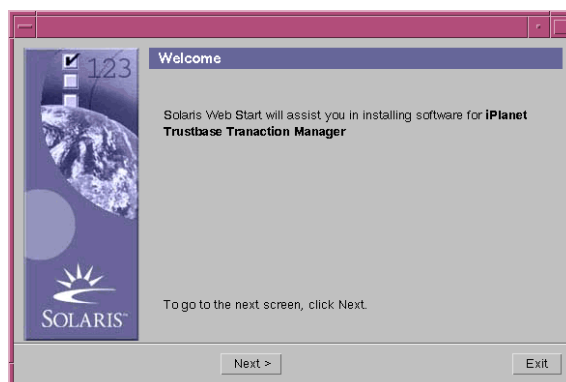


The installer will then provide you with a summary of your installation components. Select ittm.

- ittm
- JMS proxy settings
- OCSP Responder settings

First time this will ask you for the TokenKeyStore password. Enter any password to protect your TokenKeyStore.

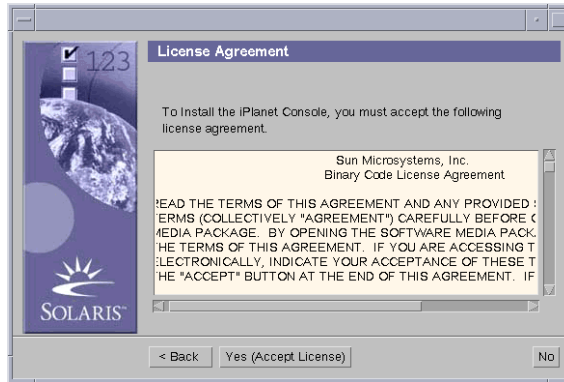
**Figure 1-8** Welcome Screen



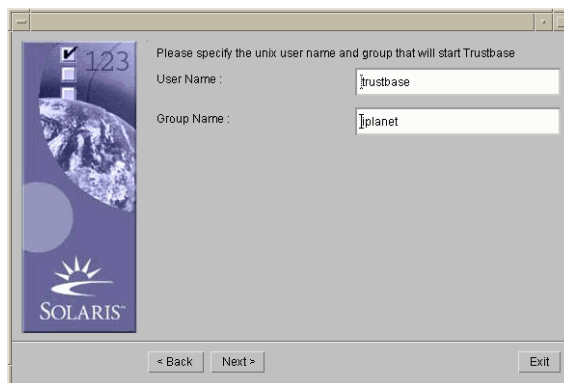
Enter data and select <next>

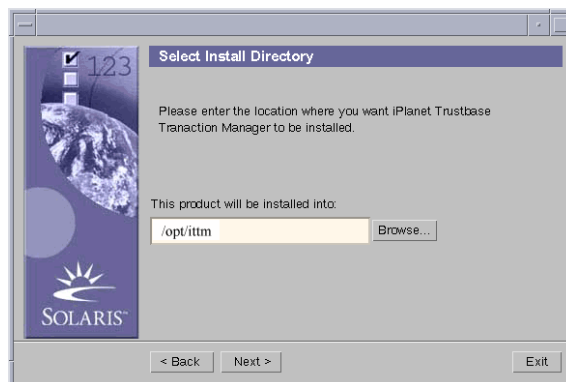
Define a user and group account to run iTTM from. This is the same as System User and System group you entered when installing iWS and iAS.

**Figure 1-9** License Agreement

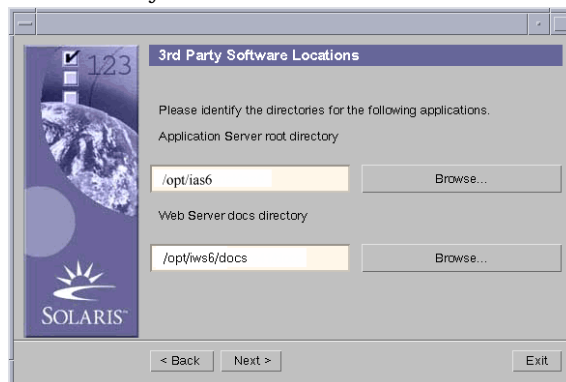


**Figure 1-10** User Group and account



**Figure 1-11** Install Location

By selecting <next> you may be asked whether the system can create the iTTM install directory. Enter data and select <next>

**Figure 1-12** 3rd Party Software Location

The Installer will then check to see whether your Application Server is running. Enter data and select <next>

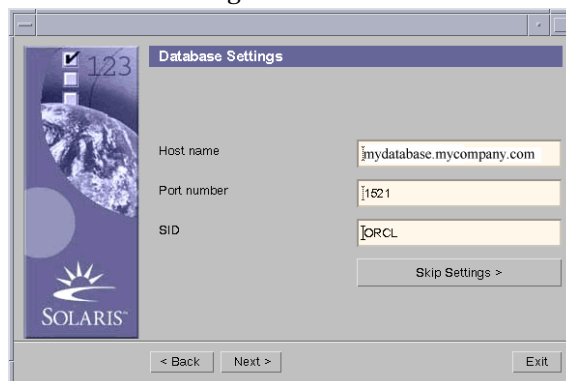
**Figure 1-13** TokenKeyStore password



The following parameters must be defined:

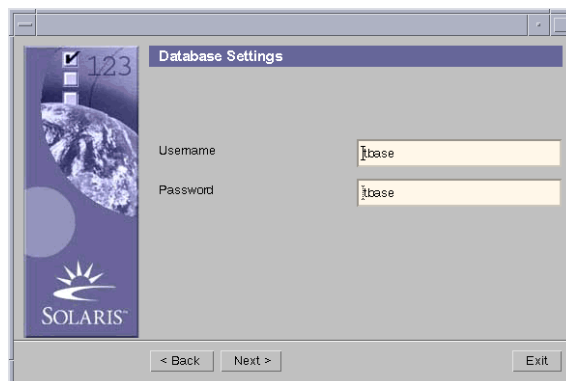
- Oracle login name (User defined)
- Oracle login password (User defined |)
- Oracle hostname The machine I.D> of where your Oracle installation is located
- Oracle port number This is normally set to 1521
- Oracle SID This is normally orcl but can be defined when you configure your own Oracle installation
- Oracle Driver Location

Before continuing you will need to create an Oracle user and as such you should follow the instructions on how to create a user described in “Oracle Database Configuration,” on page 78

**Figure 1-14** Database Settings

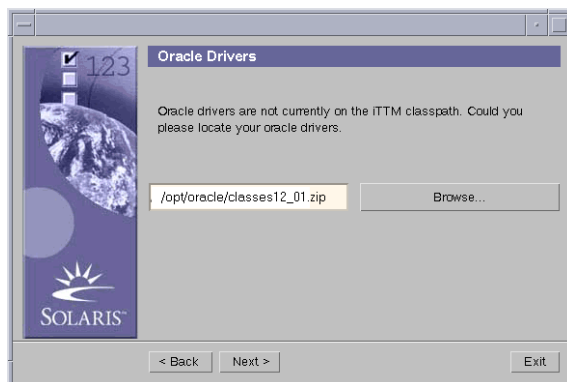
The <skip settings> option allows advanced users to skip these settings and return to them at a later date. Under normal circumstances this should not be used.

Enter data and Select <next>

**Figure 1-15** Database Username and Password

Enter data and select <next>

**Figure 1-16** Oracle Driver Location

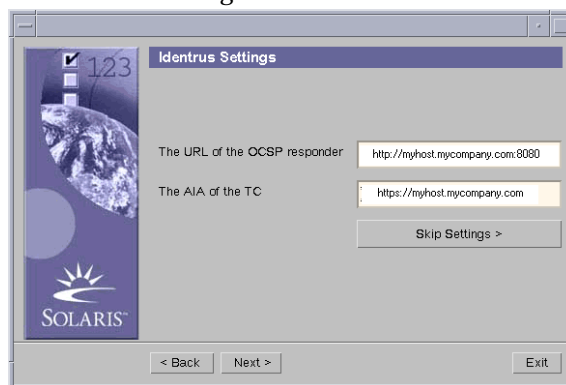


Enter data and select <next>

This completes your Oracle settings.

- Make sure AIAs are set correctly within your PKI software.

**Figure 1-17** Identrus Settings



- The URL of the OCSP Responder tells you where you can get authoritative responses to CSC checks. When running the environment wizard, the URL location of the OCSP responder is requested. In order for OCSP requests to be delivered to the iTTM responder, enter the following URL

`http://myresponder.mycompany.com/NASApp/OCSPResponder/OCSPResponderServlet`

Since the OCSP Responder is local to the iTTM application the protocol HTTP is used instead of HTTPS to maximise performance. If an external OCSP responder is used the following address is typically used:

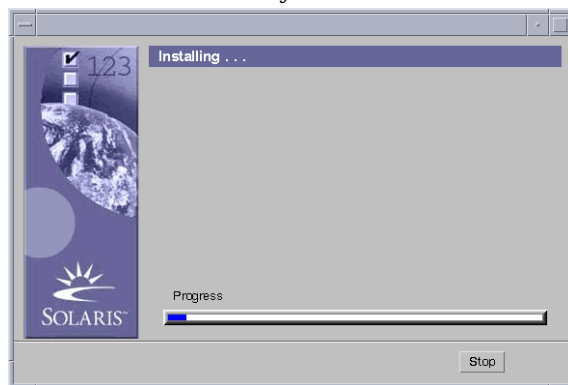
`https://myocspresponder.com`

- The AIA Access Information Authority is the field within a certificate that is used to determine whether or not the TC (or iTTM installation) has authority over that particular certificate. In the case where a particular certificate's AIA matches the TC's choice of AIA instance we can act on this certificate in an authoritative way. In the case where the certificate's AIA is not the same as your TC's (or iTTM installation) AIA no authoritative action can be performed on this certificate. The default value is the host name that a customer or other TC connects to for information. This is normally the same as the iTTM host name.

<https://myhost.mycompany.com>

- As before, the <skip settings> option allows advanced users to skip these settings and return to them at a later date. Under normal circumstances this should not be used.

**Figure 1-18** Installation Summary



The system then concludes this part of the iTTM installation.



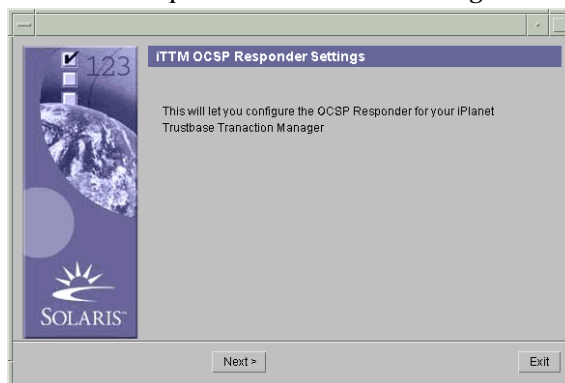
## OCSP Responder

TTM 3.0.1 provides its own OCSP responder which can be utilised as part of the certificate status check process or any other services that require certificate status checking. The OCSP Responder creates directories for the LDAP CRL list and as such its best to publish your CRL list within your CA after you have run the OCSP Responder installation procedure. If not you must make sure the location of the CRL list is the same for runOCSPResponderWizard as the one you use with CA.

Run the following script, logged in under root and Select Option (2) OCSP Responder

```
./cdrom/cdrom0/ittm/setup -g
```

**Figure 1-19** OCSP Responder Wizard Welcome Page



- Click next to process the wizard.

The database settings screen allows configuration of the LDAP directory database that will contain information about revoked certificates. A CA can publish CRLs (certificate revocation lists) to this directory which the OCSP responder will use to provide its responses.

The iAS provided with Trustbase, includes a directory server which it uses for its own configuration. This directory server can also be used for the OCSP responder. Ensure the directory server is running before continuing this wizard.

To use the iAS directory, enter the details of the directory server chosen when installing iAS. A default iAS install will normally place the directory server on port 389 and the Bind DN as “cn=Directory Manager”. The Base DN can be set to any valid distinguished name or left as the wizard defaults. The password is the LDAP password you used in the iAS 6.5 install.

**Figure 1-20** OCSP Database Settings

Field	Value
Host name	myhost.mycompany.com
Port number	389
Bind DN	cn=Directory Manager
Password	*****
Base DN	ou=Trustbase OCSP Responder, o=

Clicking next to configure the OCSP responder with the entered LDAP directory settings.

## JMS Proxy Setup

iTTM supports any JMS Message queue system and requires

1. A Host Name of the machine where the JMS broker is listening. The default is myhost.mycompany.com
2. A Port Number on which the JMS broker is running. The default is set to 7676
3. A Queue Name to indicate where iTTM receives messages. The default is set to backend\_to\_itps

These settings can be changed, only when you have installed iTTM, in the following properties file:

```
/opt/ittm/myhost/jmsproxy.properties
```

4. iTTM also requires a JMQ driver location. You will be asked for this location while you are installing iTTM.

Details of the JMS specification can be found in

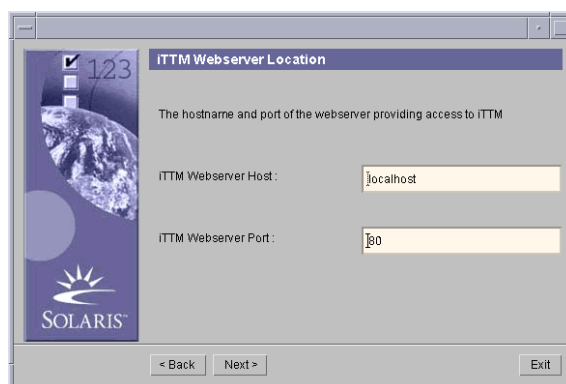
```
http://docs.sun.com/db/doc/816-5904-10
```

Select option (3) JMS Proxy, logged in under root, after typing

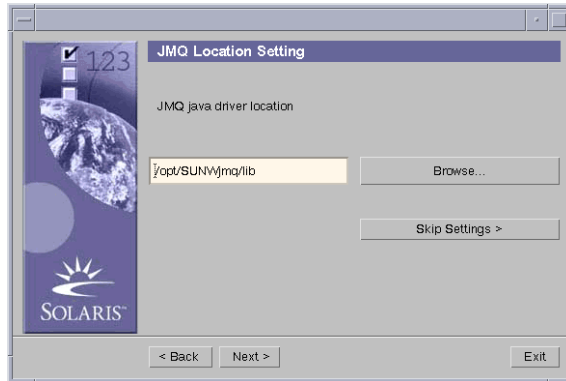
```
./cdrom/cdrom0/ittm/setup -g
```

The following screen should appear:

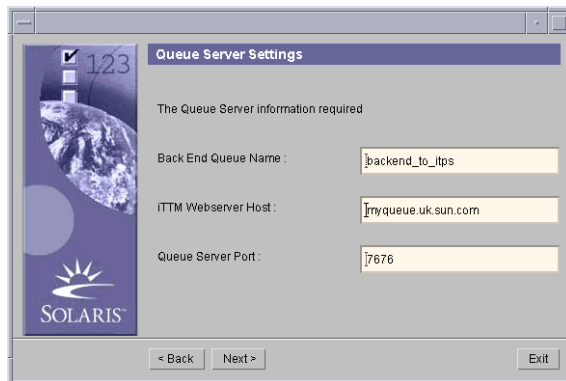
**Figure 1-21** Web Server Host



**Figure 1-22** JMQ Driver location



**Figure 1-23** Queue Server Name



# iTTM Configuration issues

The following needs to be configured before starting iTTM

- OCSF Responder with CA
- iTTM and OCSF Responses
- URL rewrites with iWS

## Configuring the Trustbase OCSF Responder with the Certificate Authority

In order for the OCSF responder to give the appropriate certificate status check results, it will be necessary to configure the CA to publish the revocation list to the same directory server configured for this responder.

This procedure will vary with different Certificate Authority software, but the following guidelines should be employed.

Publish the CRL to the same LDAP directory chosen in the OCSF installation wizard.

Publish the CRL under the same Base DN chosen in the installation wizard

The CA certificate that the OCSF responder provides responses for should be Published with the attribute “cacertificate;binary”

The CRL should be published to the OCSF responder with the attribute “certificaterevocationlist;binary”

## **Configuring Trustbase to validate the OCSP responses**

By default Trustbase does not check the signature on the OCSP responses. This can be enabled by editing the “identrus.properties” file as documented in “Identrus Configuration,” on page 98.

When a message is sent to any node a response comes back. That response needs to be verified in such a way that the sender of the response is who they say they are. This is an optional feature in that it can be assumed that responses are trusted. OCSP Responders that are used locally are unlikely to require this signing validation process as their communication can be considered secure. However OCSP Responders on non-local or insecure lines should have this feature configured.

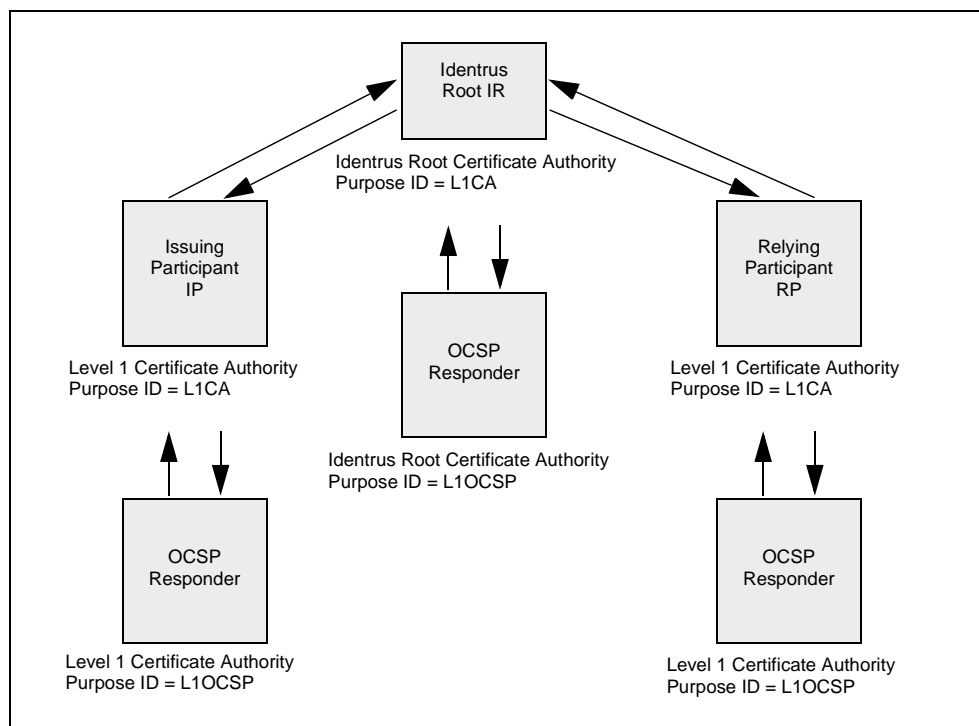
When an RP bank communicates with an IP that does not have an iPlanet Trustbase Transaction Manager Transaction Coordinator and fails it will need an OCSP Responder. This is sometimes referred to as OCSP fallback.

In order to verify an OCSP response we need to input the signing Certificate into the iTTM TokenKeyStore. Normally the OCSP responder certificate is directly descended from the Identrus Root. If this is the case, there is no need to perform any further configuration for response signature checking.

If a specific certificate is required to validate OCSP responses, the token key tool will need to be used to import the certificate. It then needs to have the alias of “L1OCSP” added to this certificate. For details on how to do this, see the appendix at the end of this guide.

Once the OCSP signing Certificate has been successfully imported into the iPlanet Trustbase Transaction Manager token key store, a change needs to be made to the identrus.properties file. This file can be found in /opt/ittm/myhost. Change the entry OCSPResponderSigningCertificateRole as follows:-

```
OCSPResponderSigningCertificateRole=L1OCSP
```

**Figure 1-24** OCSP Responders

## Configuring the AIA for the OCSP responder

Identrus compliant certificates contain two AIAs. One refers to the address of the TC. The second refers to the address of the Responder. This should be set to the following address.

```
http://myhost.mycompany.com/NASApp/OCSPResponder/OCSPResponderServlet
```

Since the OCSP Responder is local to the iTTM application the protocol HTTP is used instead of HTTPS to maximise performance. If an external OCSP responder is used the following address is typically used:

```
https://myocspresponder.com
```

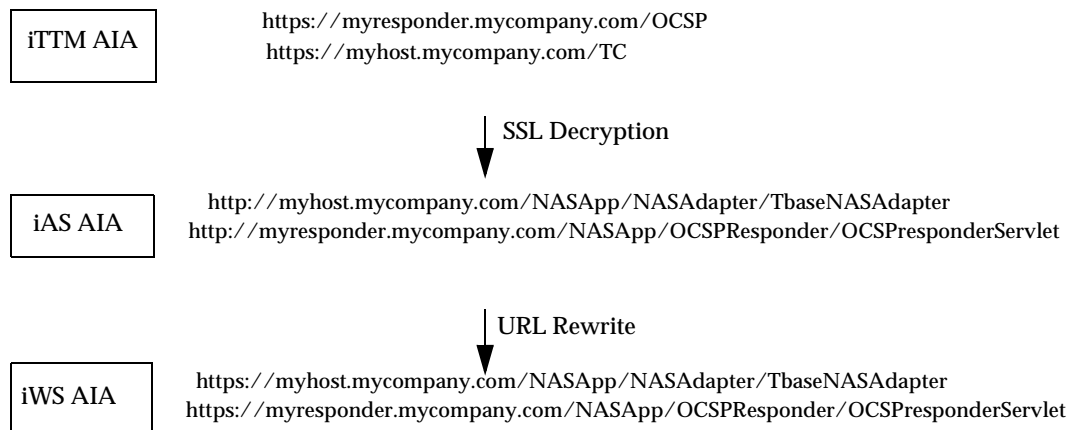
iPlanet Trustbase Transaction Manager is now installed. Some further configuration is now required to establish the local settings and Identrus setup - see the following sections.

## Configuring the iWS for URL rewriting and SSL Decryption

The URL's presented in the Certificate AIA's are typically a human readable form of the URL's required by the Trustbase application server. These must therefore be converted. The Web Server is configured to perform this task.

The following diagram illustrates how URL rewrites that take place between iTTM, iWS and iAS.

**Figure 1-25** Configuring URL rewrites within iWS





## **iTTM AIA's**

All deployed TC's will require the following translation to occur:

To map a URL such as `http://myhost.mycompany.com`,  
`https://myhost.mycompany.com` or `https://myhost.mycompany.com /TC` to  
`http://myhost.mycompany.com/NASApp/NASAdapter/TbaseNASAdapter`.

The short URL is typically issued in the Certificate AIA's of the L1 PKI.

Two steps are required:

1. If your TC responds to HTTPS then we need to decrypt the SSL session and forward the request to the web server HTTP port.
2. To rewrite the URL to `/NASApp/NASAdapter/TbaseNASAdapter`

## **OCSP AIA's**

If you are using the Trustbase OCSP Responder you will need to map the URL  
`http://myresponder.mycompany.com/ocsp` or

`https://myresponder.mycompany.com/ocsp` to  
`http://myresponder.mycompany.com/NASApp/OCSPResponder/OCSPResponderServlet`

The short URL is typically issued in the Certificate AIA's of the L1 PKI.

Two steps are required:

1. If your OCSP Responder responds to HTTPS then we need to decrypt the SSL session and forward the request to the web server HTTP port.
2. To rewrite the URL to `/NASApp/OCSPResponder/OCSPResponderServlet`.

## Configuring the Web Server for SSL

The web server performs SSL decryption

The web server is configured to respond to port 80 by default. In addition to this port we need to add a new port listener on 443 (SSL) or any other desired SSL port.

Port 80, HTTP, must be maintained as it will be used for internal non-SSL login to Trustbase and accessed via the JMSPROXY, which does not support SSL, used for asynchronous messaging.

Note: if required in your installation you will need to install the PKCS#11 hardware security module into this web server. This is documented in the ncipher Security Module's install guide.

## Installing the SSL Server Certificate

1. Go to the Web Server administration port, typically  
`http://myhost.mycompany.com:8888`
2. Select your <Manage Servers>. Your Server instance will be selected by default. Note: you won't be adding SSL to your administration server, only the instance configured for HTTP port 80.
3. Select <Security > <Create Database>
4. Request a Certificate. CA URL should be 'none'. Complete all other fields as appropriate but use a relevant name for the Common Name such as the fully qualified hostname of the iTTM host.
5. Sign this with your Root CA using the SSL Server profile.
6. Install Certificate. Certificate is for 'This Server' and Certificate Name should be left blank, which will then default to 'Server-Cert'.
7. Paste the certificate response and press OK. Don't restart the Server at this time.

## Configuring an Additional Listener on Port 443

1. Go to the Web Server administration port, typically  
`http://myhost.mycompany.com:8888`
2. Select your <Manage Servers>.
3. Select <Preferences> <Add Listen Socket>
4. The following values should be used:
  - a. ID: SSL\_Proxy
  - b. IP: 0.0.0.0
  - c. Port: 443
  - d. Servername: myhost.mycompany.com
  - e. Security: On
  - f. Select the default VS
5. Select <ok>
6. All other settings should be left untouched.
7. Just to be sure, select <Edit Listen Sockets>
  - a. Ensure Security is ON for SSL\_Proxy socket
  - b. Select Attributes for SSL\_Proxy socket:
  - c. Make sure that Server-Cert is selected
8. Press <ok>

You may now restart the server by selecting <Apply Changes>. Port 80 (HTTP) & 443 (HTTPS) should now be able to access the web server home page.

## Configuring NSAPI in the Web Server to rewrite URL's

Sometimes it is necessary to rewrite a URL, using Netscape SSL proxy, such that the URL seen by the outside world differs from that used internally by the application server. To perform such rewrites in iWS, the following steps must be taken.

1. From the root account, Stop the Web Server Instance that you are making this change to.

2. Locate the `tb_url_rewrite.so` in your distribution, typically

```
/opt/ittm/current/Config/WebserverSetup/JWS/plugins/tb_url_rewrite.so
```

If you are installing iAS and iWS on separate machines you should also consult "URL Rewrites with iAS and iWS on separate machines," on page 148

3. Edit adding two new lines

```
/opt/iws6/https-myhost.mycompany.com/config/magnus.conf
Init fn="load-modules" \
shlib="/opt/iws6/plugins/lib/tb_url_rewrite.so" \
funcs="basic_rewrite,basic_rewrite_init"
Init fn="basic_rewrite_init"
```

iWS maintains an archive of previous modifications to this file in case there is a need to revert back

4. Configuring the URL Rewrites. These translations must appear immediately after the line beginning `<Object name=default>` or they will not function. Edit

```
/opt/iws6/https-myhost.mycompany.com/config/obj.conf
```

adding the following line for each rewrite

```
NameTrans fn="basic_rewrite" from ="from_path" to="to_path"
```

For example to forward: `https://myhost.mycompany.com/TC` add the line:

```
NameTrans fn="basic_rewrite" from="/TC" \
to="/NASApp/NASAdapter/TbaseNASAdapter"
```

iWS maintains an archive of previous modifications to this file in case there is a need to revert back

5. Installing the NSAPI Library.

```
cp  
/opt/ittm/current/Config/WebserverSetup/JWS/plugins/tb_url_rewri  
te.so /opt/iws6/plugins/lib
```

- 6. Restart the Webserver instance. This asks for iWS 6.0 cert database password.**

## Other Suggested rewrites

The following rewrites might be required for Identrus compliance. When using rewrites, remember that all URL configuration entries should refer to the external URL. For example the Local OCSP Responder should be directed to

```
http://myhost.mycompany.com/OCSP
```

Edit

```
/opt/iws6/https-myhost.mycompany.com/config/obj.conf
<Object name="default">
```

adding the following line to each rewrite

```
NameTrans fn="basic_rewrite" from="/ocsp" \
to="/NASApp/OCSPResponder/OCSPResponderServlet"
```

This rewrite rule is optional but will allow access to the logon screen with a short URL: <https://myhost.mycompany.com/ocsp>

To achieve a rewrite of a base URL such as <http://mycompany.com>, it is necessary to utilise the browsers internal rewrite rules by placing “/index.html” in the from field. For example,

```
NameTrans fn="basic_rewrite" from="/logon" \
to="/NASAdapter/LogonFrame.html"
NameTrans fn=basic_rewrite from="/index.html" \
to="/NASApp/NASAdapter/TbaseNASAdapter"
```

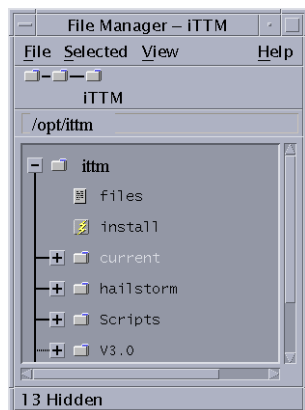
It must be noted that this will rewrite all URL's ending in “/”

## Installation Structure

Once the installation has completed the following directory structure and support files will exist:

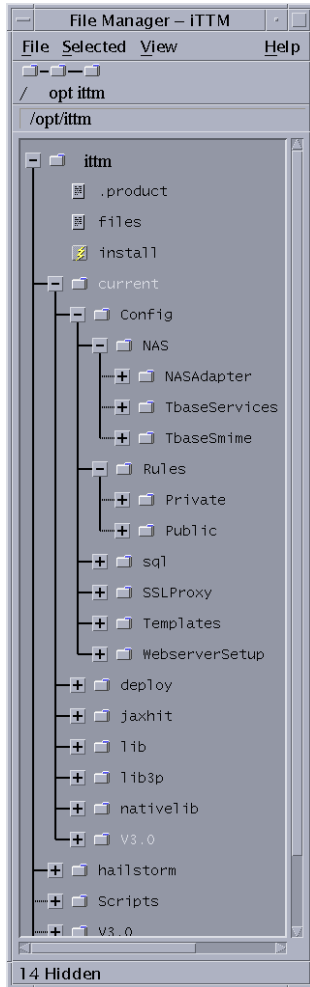
- iPlanet Trustbase Transaction Manager

**Figure 1-26** iPlanet Trustbase Transaction Manager Directory Overview



- **ittm**  
iPlanet Trustbase Transaction Manager contains all configuration and Library files, the SQL directory containing various database setup scripts.

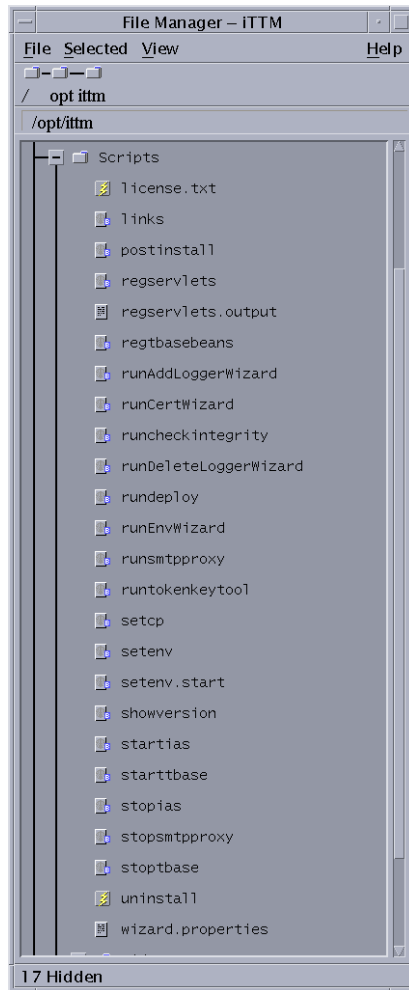
**Figure 1-27** iPlanet Trustbase Transaction Manager Overview Directory





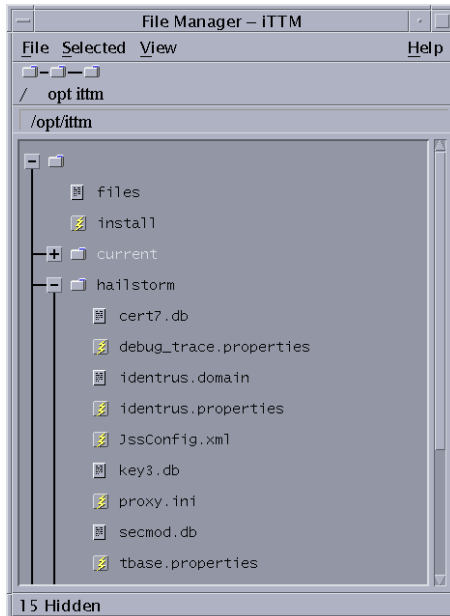
- **Scripts**  
This directory contains all of the scripts needed to start and stop iPlanet Trustbase Transaction Manager

**Figure 1-28** iPlanet Trustbase Transaction Manager Commonly Used Scripts



- **myhost**  
This directory contains all of the configuration files for this installation. i.e. /opt/ittm/myhost/tbase.properties, nFast.properties, identrus.properties  
These files should not be modified directly but can be accessed via the configuration screens in this manual.

**Figure 1-29** iPlanet Trustbase Transaction Manager Initialisation Files



- **current**  
This directory contains all of the TTM binaries and support files

- apidocs

Javadocs

- samples

Developer Examples

- store

Certificate store

## iTTM Checklist

The following operational commands are often used while running iTTM:

<b>Information Type</b>	<b>Example Set-up Value for iTTM 3.0.1</b>
Install directory	/opt/ittm
Administration logon via web	Username: administrator, Password: administrator
TokenKeyTool	/opt/ittm/Scripts/runtokenkeytool
Operational ports	Admin via web: 80 ( <a href="http://myhost.mycompany.com/NASAdapter/logon.html">http://myhost.mycompany.com/NASAdapter/logon.html</a> )
To start server	/opt/ittm/Scripts/startias
To stop server	/opt/ittm/Scripts/stopias
Property file location	/opt/ittm/myhost
Processes grep	ps -ef   grep java
Process list	root 9713 1 0 12:47:53 pts/6 0:08 /usr/bin/./java/bin/./jre/bin/./bin/sparc/native_threads/java uk.co.jcp.tbase
Installation Document	<a href="http://docs.sun.com/?p=prod/sl.iptbtranm">http://docs.sun.com/?p=prod/sl.iptbtranm</a>

## Post Installation Steps

Before starting iPlanet Trustbase Transaction Manager for the first time it is necessary to do some further configuration steps, in each of the following configuration steps refer to the “Installation Structure” section above to locate the necessary files:

- nCipher Security setup - see section “Oracle Database Configuration,” on page 78
- Oracle Database setup - see section “Oracle Database Configuration,” on page 78
- Identrus site specific setup - see section “Identrus Configuration,” on page 98

# Oracle Database Configuration

Both TTM and the Identrus Extensions require access to a database with a pre-configured schema. The installation comes with SQL scripts that must be executed in the database user's tablespace that was specified at installation time. The following sections explain how to create the user and generate the correct schema for that user.

## Oracle Database Setup pre-requisites

iTTM requires that the Oracle instance have the following minimum configuration values, these are defined in the init file for the Oracle instance:

- The generation of users and the tablespaces defined may differ for individual sites - contact the site DBA for advice. There needs to be a minimum of 100MB of free space in the 'default' tablespace for each iTTM database user, this comfortably allows for a basic installation and testing volumes.
- Advise DBA that the database block size (db\_block\_size) for the instance must be at least 8192 to accommodate the indexing requirements of the iTTM schema. It should also be noted that this parameter has no effect until the underlying database files are recreated with the new block size.
- Advise DBA that processes should be at least 200, this is sufficient to support 2 concurrent iTTM instances, increase this figure to accommodate additional instances.
- Advise DBA that open\_cursors should be at least 300

From a running SQL session, logged on as system user the following query will show the current values of the settings:

```
select name, value from v$parameter where name in  
( 'processes', 'open_cursors', 'db_block_size' );
```

## Running the iPlanet Trustbase Transaction Manager SQL Scripts

- You may need to ask your DBA to create your username and password. The following procedure should be followed.
- Switch to the Oracle user and run server manager:

```
myhost> su - oracle
Password:
myhost> cd ../opt/ittm/current/Config/sql
myhost> svrmgrl

Oracle Server Manager Release 3.1.5.0.0 - Production

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

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.7.0.0 - 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 RESTRICTED 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 Trustbase Transaction Manager 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;
SVRMGR> GRANT CONNECT TO tbase;
SVRMGR> GRANT RESOURCE TO tbase;
SVRMGR> ALTER USER tbase QUOTA UNLIMITED ON USERS;
SVRMGR> quit
Server Manager complete.
```

- **Connect as the iPlanet Trustbase Transaction Manager user and run the scripts:**

```
mydatabase% su - oracle
mydatabase% cd /opt/ittm/current/Config/sql
mydatabase% sqlplus
SQL*Plus: Release 8.1.7.0.0 - Production on Fri Feb 15 12:07:11 2002
(c) Copyright 1999 Oracle Corporation. All rights reserved.
Enter user-name: tbase
Enter password:
Connected to:
Oracle 8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.7.0.0 - Production

SQL>spool myoutput.txt
SQL>set echo on
SQL>@tbaseNew.sql
```

- **Existing users may upgrade from iTTM.2.2.1 to iTTM3.0.1 using the script**  
`/opt/ittm/current/Config/sql/tbaseUpgrade3_0.sql`
- **Uninstalling may be achieved by deleting all user scheme objects using**  
`/opt/ittm/current/Config/sql/Drop_tbaseAll.sql`
- **You should check the output file to ensure that all SQL tables were created successfully.**  
`/opt/ittm/current/Config/sql/myoutput.txt`



# Oracle Example Checklist

The following provides typical operational commands available within Oracle.

Information Type	Example Set-up Value for Oracle 8.1.7
Install directory	Oracle program files: /opt/oracle/app/product/8.1.7/bin    Oracle data files: /identrusdb/orcl
Oracle user login	Username: oracle, Password: oracle
Sqlplus - dba admin	Username: sys, Password: change_on_install
Sqlplus - tbase user	Username: tbase, Password: tbase
Operational ports	Oracle ports: 1521
SID	orcl
To start server	As oracle user: svrmgrl; Connect internal; startup; exit lsnrctl; start; exit
To stop server	As oracle user: lsnrctl; stop; exit svrmgrl; connect internal; shutdown; exit
Processes grep	ps -ef   grep oracle
Logs of interest	Auditdata: Contains internal audit information & indicates what the TC processed. Error: Shows unexpected errors e.g. cannot communicate with Certificate Authority Error_support: Shows any java stack trace associated with the error table.

# Cryptographic Services Configuration

Cryptographic operations are an integral part of iTTM function. iTTM can make use of several different cryptographic service providers. A cryptographic service provider offers key and certificate management, digital signature, encryption and ssl functions. The choice of which service provider to use is dependent on the nature of the iTTM installation. Two service providers are supplied in an iTTM installation, the *PKCS#11 cryptographic service provider* and the *nCipher native cryptographic service provider*.

## Cryptographic Installation Scenarios

You must select one of the following scenarios that come packaged with a provider:

1. Software PKCS11 for iTTM 3.0.1
  - PKCS11 provider with software token
2. Generic Hardware PKCS11 for iTTM 3.0.1
  - PKCS11 provider with hardware token
3. nCipher Hardware PKCS11 for iTTM 3.0.1
  - PKCS11 provider with hardware token
4. nCipher Upgrade from iTTM 2.2.1 to iTTM 3.0.1
  - nCipher native provider

We now describe each of the providers that are associated with each installation Scenario

# Cryptographic Providers

1. PKCS#11 cryptographic service provider [ also known as the “JSS” service provider ]. *PKCS#11* is an industry standard interface to cryptographic services devices, known as Tokens. iTTM supports several different PKCS#11 Tokens

- a. Software PKCS#11 Token

The software Token is not suitable for a production environment: it is only appropriate for test and development installations. While private keys are protected by password based encryption, the password is freely available, so private key material is not secure.

- b. Hardware PKCS#11 Token

This is the normal service provider to use in a production environment. Private key material is stored securely, either on a hardware token, or wrapped with a symmetric key stored on a hardware token. The iTTM PKCS#11 interface is tested with nCipher hardware Token.

2. nCipher native cryptographic service provider [ also known as the “TTM-NCIPHER” service provider ]. The nCipher native provider communicates directly with nCipher hardware security modules, using an nCipher proprietary protocol. In previous versions of iTTM, the only support for hardware security modules was provided in this manner.

If keys created under an iTTM 2.2.1 installation are to be used with an iTTM 3.0.1 installation, this service provider must be used. Private key material is stored on disk or in a directory, securely wrapped with a secret key persistently stored on an nCipher HSM. This provider is suitable for use in a production environment

In order to run the scripts from iTTM, log in under your username e.g. tbase. We now discuss the four Installation Scenarios possible.

## PKCs11 provider with software token

The PKCS#11 cryptographic service provider always has an active software Token, which is selected as the default Token in a standard iTTM installation. No action is required beyond the normal iTTM installation if the soft PKCS#11 Token is the only Token to be used

## PKCS#11 provider with generic hardware token

The vendor of the hardware Token supplies a PKCS#11 library, through which iTTM interacts with the module. Before the PKCS#11 service provider can use a hardware Token, the hardware must be correctly configured and initialised [ consult vendor documentation. For nCipher hardware see also the section entitled “nCipher Module Installation” below ]. If you are configuring an nCipher hardware security module for use with iTTM, then scripts are available to install the vendor PKCS#11 library for you. See the section “nCipher PKCS#11 Token configuration” which follows on from nCipher Module Installation. Otherwise, follow the instructions presented here.

There are two steps to be taken in configuring the PKCS#11 cryptographic service provider for use with a hardware Token.

- a. Identifying the HSM vendor’s PKCS#11 library to the Plug-in PKCS#11 cryptographic services
  - I. You will need
    - o the location of the hardware security module vendor’s PKCS#11 library
    - o a module name [ can be anything, so long as it’s not already in use ]
  - II. Scripts are included in the iTTM distribution to assist the installation of PKCS#11 libraries
  - III. Change directory to the iTTM Scripts directory

```
cd /opt/ittm/Scripts
```

#### IV. run the installP11Library script

```
./installP11Library -module <moduleName> -libfile <vendorLibrary>
```

For example

```
./installP11Library -module ncipher -libfile
```

#### V. Check the installation using the modutil script

```
./modutil -nocertdb -list
```

For example

```
./modutil -nocertdb -list
```

## VI. 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
-----
```

### b. Configuring the default Token

- I. Once a vendor PKCS#11 library has been installed, iTTM can use keys and certificates stored on Tokens provided by that library. New cryptographic Objects will, however, still be created on the default Token, which is the software Token in a standard install
- II. Change directory to the iTTM host specific configuration directory

```
cd /opt/ittm/myhost
```

### III. edit the JssConfig.xml file

```
vi JssConfig.xml
```

**IV. Change the defaultToken attribute of the JssConfig element to be the desired default token name [ as reported by modutil in the previous section ]. An empty string defaultToken attribute value refers to the software PKCS#11 Token**

```

<JssConfig
  defaultToken="tokenName">
  <!-- pathnames are processed in the following ways
      pattern ~ at the start of the path is replaced with the System
property "user.home"
      pattern $JSSCONFIGDIR at the start of the path is replaced
with the directory this
          file is loaded from, if it was loaded from a file,
rather than a jarred resource
          or directory
      pattern $HOSTNAME : the first occurrence is replaced with the
hostname -->
<JssDbConfig
  configDir="$JSSCONFIGDIR"
  moduleDb="secmod.db"
  keyDbPrefix=""
  certDbPrefix=""
  readOnly="false"/>
<JssSslConfig
  cacheMaxEntries="10000"
  cacheSsl2Timeout="100"
  cacheSsl3Timeout="100"
  cacheDirectory=".">
SSL3_RSA_WITH_RC4_128_SHA
  SSL3_RSA_WITH_RC4_128_MD5
  SSL3_RSA_WITH_3DES_EDE_CBC_SHA
</JssSslConfig>
</JssConfig>

```

**V. Save the file, leaving other lines untouched**

## Example Generic PKCS11 Checklist

<b>Information Type</b>	<b>Example Set-up Value generic PKCS11</b>
Adding PKCS11 Library	<pre>cd /opt/ittm/Scripts ./installP11Library -module ncipher -libfile /opt/nfast/toolkits/pkcs11/lib-cknfast.so</pre>
Verifying addition of PKCS11 Library	<pre>cd /opt/ittm/Scripts ./modutil -nocertdb -list</pre>
Configuring default token	<pre>cd /opt/ittm/myhost vi JssConfig.xml</pre>



## PKCs11 provider with nCipher token

There are two steps

### 1. nCipher module installation

For each group of nCipher modules that will be sharing a common key database, the following setup procedure must be adopted.

- a. Install the nCipher hardware as described in the supplied documentation, making sure that each unit is set to be in pre-initialisation mode. The nCipher documentation describes the process of installing and setting up the "Security World" on a module (See nCiphers: KeySafe user guide).
- b. The nFast module will only accept connections from the local host. Therefore the nFast module must be located on the same host computer used to run the TC.
- c. Create a "Security World" on one module, using the nCipher KeySafe tool.
- d. Copy this "Security World" to the other modules, and install this world using the "restore" function in KeySafe.
- e. The nfast startup script must be modified to ensure the nFast server can communicate with the iPlanet™ Trustbase Transaction Manager nFast stub via TCP. To do this, the environment variable NFAST\_SERVER\_PORT must be defined and exported in /etc/init.d/nfast. The iPlanet™ Trustbase Transaction Manager stub has a default value for this port of 9000, but it can be set to any available port.
- f. Once these settings have been made, the server should be restarted. If the nCipher software has been installed in the default location the commands to do this are:

```
/opt/nfast/sbin/init.d-nfast stop
/opt/nfast/sbin/init.d-nfast start
```

**Note** If the NFAST\_SERVER\_PORT value is anything other than 9000, the nFast.properties file in the iPlanet™ Trustbase Transaction Manager install directory (/opt/ittm/myhost) must be modified such that the StandardPort property is consistent.

Once the nCipher modules have been initialised into a security world follow one of the two procedures outlined below "Upgrading an iTTM 2.2.1 nCipher key store to iTTM 3.0.1" or "nCipher PKCS#11 Token configuration"

### 2. nCipher PKCS#11 Token configuration

If the nCipher hardware security module is to be used with the PKCS#11 cryptographic services provider, rather than the nCipher native cryptographic services provider follow this procedure, which installs the nCipher PKCS#11 library and marks the nCipher Token as the default for iTTM.

Once the nCipher modules have been initialised into a security world, some operator cards must be created for use with the nCipher PKCS#11 interface. An operator card must be inserted into the nCipher module while it is being used through the PKCS#11 interface.

- a. Change directory to the nFast binary's directory

```
cd /opt/nfast/bin
```

- b. Use the createocs or createoc-simple commands to create an operator card or cardset [ see nCipher documentation ] . The name of the card or cardset is required but the installP11Library script used below. The simplest configuration creates a single operator card

```
./createoc-simple [ --fips <AUTHSLOT> ] [ --overwrite ]
                  <MODULE> <SLOT> <NAME> 1 0
```

For example

```
./createoc-simple 1 0 operator 1 0
```

- c. Change directory to the iTTM Scripts directory

```
cd /opt/ittm/Scripts
```

- d. run the installP11Library script

```
./ncipherP11Install [ -libfile <ncipherLibrary> ]
                   <NAME>
```

For example

```
./ncipherP11Install operator
```

The nCipher PKCS#11 Token is now installed as the default provider for iTTM cryptographic services

## nCipher Example PKCS11 Checklist

Information Type	Example Set-up Value nCipher
Install directory	/opt/nfast
Operational ports	9000
To start server	/etc/init.d/nfast start
To stop server	/etc/init.d/nfast stop
Processes grep	ps -ef   grep hard
Process list	nfast 4241 1 0 Mar 05 ? 0:22 ../sbin/hardserver -llogfile nfast 4246 4241 0 Mar 05 ? 0:10 ../sbin/hardserver -llogfile
Creating nCipher operator card	cd /opt/nfast/bin ./createoc-simple 1 0 operator 1 0
Installing nCipher PKCS11 module	cd /opt/ittm/Scripts ./ncipherP11Install operator
Documentation	<b>nCipher KeySafe 1.0</b> <a href="http://www.ncipher.com">http://www.ncipher.com</a>

## nCipher native provider upgrade: ittm 2.2.1 to 3.0.1

There are two steps

### 1. nCipher module installation

For each group of nCipher modules that will be sharing a common key database, the following setup procedure must be adopted.

- a. Install the nCipher hardware as described in the supplied documentation, making sure that each unit is set to be in pre-initialisation mode. The nCipher documentation describes the process of installing and setting up the "Security World" on a module (See nCiphers: KeySafe user guide).
- b. The nFast module will only accept connections from the local host. Therefore the nFast module must be located on the same host computer used to run the TC.
- c. Create a "Security World" on one module, using the nCipher KeySafe tool.
- d. Copy this "Security World" to the other modules, and install this world using the "restore" function in KeySafe.
- e. The nfast startup script must be modified to ensure the nFast server can communicate with the iPlanet™ Trustbase Transaction Manager nFast stub via TCP. To do this, the environment variable NFAST\_SERVER\_PORT must be defined and exported in /etc/init.d/nfast. The iPlanet™ Trustbase Transaction Manager stub has a default value for this port of 9000, but it can be set to any available port.
- f. Once these settings have been made, the server should be restarted. If the nCipher software has been installed in the default location the commands to do this are:

```
/opt/nfast/sbin/init.d-nfast stop  
/opt/nfast/sbin/init.d-nfast start
```

**Note** If the NFAST\_SERVER\_PORT value is anything other than 9000, the nFast.properties file in the iPlanet™ Trustbase Transaction Manager install directory (/opt/ittm/myhost) must be modified such that the StandardPort property is consistent.

Once the nCipher modules have been initialised into a security world follow one of the two procedures outlined below “Upgrading an iTTM 2.2.1 nCipher key store to iTTM 3.0.1” or “nCipher PKCS#11 Token configuration”

## 2. Upgrading an iTTM 2.2.1 nCipher key store to iTTM 3.0.1

If you have cryptographic keys that were created using an iTTM 2.2.1 installation, these keys can be used by iTTM 3.0.1 through the nCipher native cryptographic services provider. This upgrade procedure must be followed to migrate the keys from the Oracle database [ where they were stored for iTTM 2.2.1 ] to the XML based storage used by iTTM 3.0.1

The upgrade procedure changes the cryptographic service provider used by TTM, from the JSS provider [ supporting PKCS#11 interaction with HSMs ] to the native nCipher provider. It is not possible to use both providers simultaneously, so if private keys generated with a TTM 2.2.1 installation are to be used with a iTTM 3.0.1 installation, the iTTM 3.0.1 installation must use the nCipher cryptographic services provider

There are two steps to this upgrade step

### a. nCipher module setup for the nCipher native provider.

Once the Security world is installed, the following points should be noted:

- The nFast cryptographic services provider read certain settings from a properties file "nFast.properties". This file is read from the iTTM installation directory, and is only required if any of the nCipher configuration parameters differ from the default values described below. An installation which uses only the default values requires no nFast.properties file
- If created, nFast.properties should be located in the iTTM host specific installation directory

`/opt/ittm/myhost/nFast.properties`

- nFast.properties should contain the following values:

ServerAddress. The ServerAddress property should be set to the local host. If set to other values the server may refuse connection attempts on unix systems. On NT systems, the server will accept connections from clients elsewhere than "localhost". The default value is 127.0.0.1

**StandardPort.** The StandardPort property contains the number of the port on which the server is listening for standard connections. By default this value is 9000.

- **StandardConnections.** The StandardConnections property contains the number of standard connections that will be maintained with the nFast module. Low values for this property will not allow the module to make best use of its parallel processing capabilities. By default, this value is obtained from the module itself.
- **module.key.** The module.key property contains the identifying number of the module key to be used for encrypting keys for storage outside the box. If the standard install procedure has been followed, and the Security World has been correctly set up, the module keys installed on the box are:

**KM0** - the randomly generated module key that is never exported in any form. If this key is used as the default, archived keys may only ever be used on that specific module. If the module is replaced for any reason, all the keys must be regenerated.

**KM1** - a well known module key used for bootstrapping of the recovery keys. This module key should NOT be used.

**KM2** - the security world key. This is the module key that should be used, and is the default value assumed if no nFast.properties file is present

```
ServerAddress = localhost
ServerPort = 9000
StandardConnections = 10
PrivilegedConnections = 0
module.key = 2
```

**b. Importing iTTM 2.2.1 keys and certificates to iTTM 3.0.1**

- i.** During this procedure you will need: Details of the oracle database containing the cryptographic keys used with TTM 2.2.1 at a minimum, the hostname, username and password are required. additional information will be required in the following cases
  - -the Oracle listener port, if it is not 1521
  - -the database SID, if it is not ORCL
  - -the PBE password for the KeyStore, if it is not the same as the database password

- II. You will also need the name of the TrustDomain into which the nCipher keys are to be imported [ if it is not identrus ]. If this TrustDomain already exists, it must be associated with Token type TTM-NCIPHER [ if not, then the import will fail ]

- III. Change directory to the iTTM Scripts directory

```
cd /opt/ittm/Scripts
```

- IV. Run the ncipherupgrade script

```
./ncipherupgrade [ -pbe <pbdepassword:<password>> ]
                  [ -port <oracle_port:1521> ]
                  [ -sid <oracle_sid:ORCL> ]
                  [ -domain <trustdomain:identrus> ]
                  <oracle_host> <oracle_user>
                  <oracle_password>
```

- V. This script extracts keys and certificates from the TTM 2.2.1 key store in the oracle database, and imports them into the given TrustDomain. Purpose identifiers in the TTM 2.2.1 key store are converted into aliases in the TrustDomain. A TTM 2.2.1 Identrus key store imported in this manner will function as a iTTM 3.0.1 Identrus key store

The import procedure is now complete, and iTTM 3.0.1 is configured to use the iTTM 2.2.1 keys and certificates.

*Note:* The import procedure works only for nCipher keys in a TTM 2.2.1 KeyStore. The import of soft keys is not supported [ since soft keys are not to be used in a production environment ]

## nCipher Example Upgrade Checklist

<b>Information Type</b>	<b>Example Set-up Value nCipher</b>
Install directory	/opt/nfast
Operational ports	9000
To start server	/etc/init.d/nfast start
To stop server	/etc/init.d/nfast stop
Processes grep	ps -ef   grep hard
Process list	<pre>nfast 4241  1 0  Mar 05 ?    0:22 ../sbin/hardserver -llogfile nfast 4246 4241 0  Mar 05 ?    0:10 ../sbin/hardserver -llogfile</pre>
Importing iTTM2.2.1 keys	<pre>cd /opt/ittm/Scripts ./ncipherupgrade k9 tbase tbase36</pre>
Documentation	<p><b>nCipher KeySafe 1.0</b>  <a href="http://www.ncipher.com">http://www.ncipher.com</a></p>



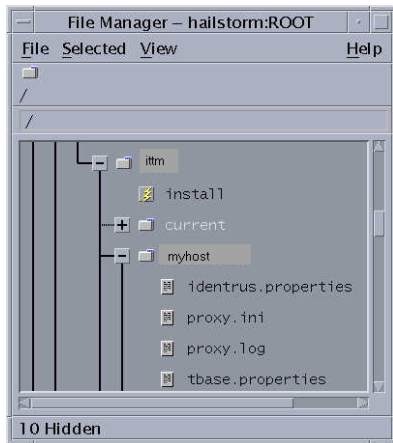
## LDAPS (Optional)

Please consult the next chapter “iAS and iWS on separate machines,” on page 147 and “LDAPS,” on page 149

# Identrus Configuration

A Transaction Coordinator (TC) comprises of all Idenstrus Services that have been deployed within the iPlanet Trustbase Transaction Manager. The file `identrus.properties` needs to be edited to change settings appropriately. Log in under your username e.g. `tbase`. It can be found in `/opt/ittm/myhost` as illustrated below:

**Figure 1-30** `identrus.properties` file location



- Normally, no changes to `Identrus.properties` need to be made. However the locations of `OurAIA` and `LocalOCSPResponderLocation` can change. For instance:

```
OurAIA = https://rp
LocalOCSPResponderLocation = http://rp:8080
```

- The following illustrates an example `identrus.properties` file.

**Figure 1-31** Example `Identrus.properties`

```
[CSC]
; OID's for OCSP and TC
TCOID=1.2.840.114021.4.1
OCSPResponderOID=1.3.6.1.5.5.7.48.1

;CaCertificateRole=
;EndEntityCertificateRole=
;SigningCertificateRole=
;RootCertificateRole=

; Local OCSP Responder Communication
SignLocalOCSPRequests=false
OCSPResponderSigningCertificateRole=L1OCSP
LocalOCSPResponderLocation = identrus_local_resp_url

OurAIA = identrus_our_aia
DNOofAuthority=

; Response Cache Parameters
ApprovedResponseMaxAge=60
ApprovedResponseMaxKeep=60
DebugMode=true
;OCSPRequestorName=CN=KENCO
```

Other settings can be modified:

- **OID's**
  - **TCOID:** This is ASN object ID for the "authority info access" attribute for Identrus Transaction Coordinator 's (default acceptable). All Identrus Certificates will have this and are supplied by Identrus with the ID of 1.2.840.114021.4.1.
  - **OCSPOID:** This is ASN object ID for the "authority info access" attribute for OCSP responder's (default acceptable).

- **Certificate Roles**
  - **CaCertificateRole:** This is the certificate attribute used for the Transaction Coordinator CA certificate. (Default is acceptable).
  - **EndEntityCertificateRole:** This is the certificate attribute used for the Transaction Coordinator end entity signing certificate (default acceptable).
  - **SigningCertificateRole:** This is the certificate attribute used for the Transaction Coordinator inter participant signing certificate (default acceptable).
  - **RootCertificateRole:** This is the certificate attribute used as by the Identrus Transaction Coordinator to identify the Identrus root (default acceptable).
- **Local OCSP Responder Communication**
  - **SignLocalOCSPRequests:** boolean - unsigned Local OCSP requests have a parameter set to false and signed Local OCSP requests have this parameter set to true.
  - **OCSPResponderSigningCertificateRole:** This is the certificate used to verify OCSP responses (only used if signLocalOCSPRequests is true).
  - **LocalOCSPResponderLocation:** The URL for the local OCSP responder.
- **Misc settings**
  - **OurAIA:** The Access Information Authority (AIA) of this Identrus Transaction Coordinator.
  - **DNOofAuthority:** This is the Distinguished Name (DN) of the Identrus root signing certificate. This is for internal use only.
- **Response Cache Parameters**

To avoid repeated calls to the Identrus root, an institution can contact the root about the status of its own certificates and store them for a cached period. Whenever a message is sent to another party, it will include this cached response. It is then up to the other institution to decide whether this cached response is acceptable or if it wants to contact the root itself. If a relying customer wants to ensure that cached responses are not used anywhere in an Identrus transaction then that customer can supply a nonce in the outgoing OCSP requests. This nonce is then used as a signal to the institutions not to use the cached response.

Caching prevents the need for repeated access to the Identrus Root. Each institution holds a value for how long its own cached certificate statuses are kept for. It also holds a maximum acceptable age of received certificate statuses. The following parameters must be defined:

- **ApprovedResponseMaxAge**: number of seconds to allow a cache entry to persist for.
- **ApprovedResponseMaxKeep**: number of seconds that a response provided to use can be acceptable.
- **DebugMode**: internal debug on/off
- **OCSPRequestorName**: name to use when talking to an OCSP responder. It should be noted that this must be set to the Distinguished name (DN) of the certificate used to sign OCSP if signing is set to true (i.e. `SignLocalOCSPRequests=true`). In the example provided, the common name (CN) is used as the distinguished name since the distinguished name is made up of the common name and the organisation unit (OU).

# iTTM Certificate Configuration

Log in under your username e.g. tbase

The following certificate scripts must be run

```
/opt/ittm/Scripts/runCertWizard
```

```
/opt/ittm/Scripts/runOCSPResponderCertWizard
```

## runCertWizard

Before continuing with runCertWizard, you will need to install your ncipher and as such you should follow the instructions described in “Cryptographic Installation Scenarios,” on page 82 This will ensure your keys are generated using hardware Cryptography instead of software Cryptography.

To run Certificate Wizard type

```
cd /opt/ittm/Scripts
```

```
./runCertWizard
```

If you make a mistake while importing certificates you can start again by deleting the following files

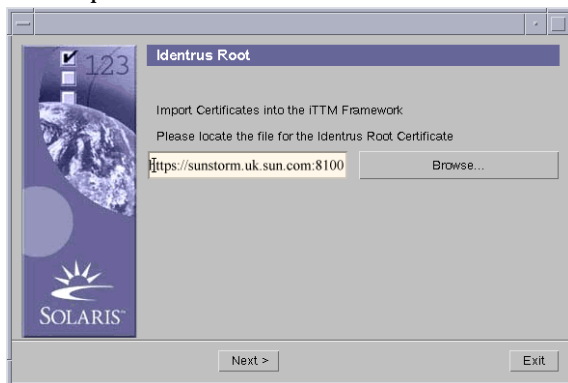
```
/opt/ittm/store/*
```

**Figure 1-32** iTTM Certificate Wizard Welcome screen



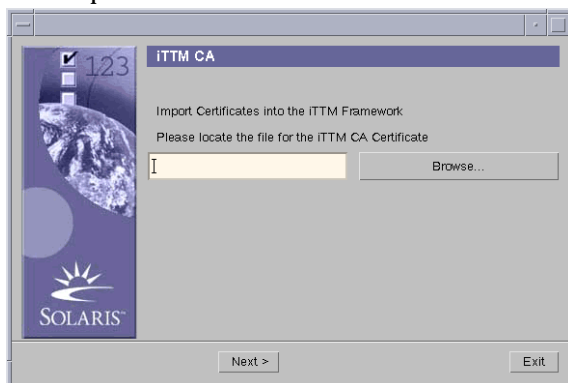
You will need to collect your Identrus root certificate by specifying its location.

**Figure 1-33** Import Identrus Root Certificate



Next you will need to supply the certificate of your L1 CA by specifying its location.

**Figure 1-34** Import iTTM CA Certificate



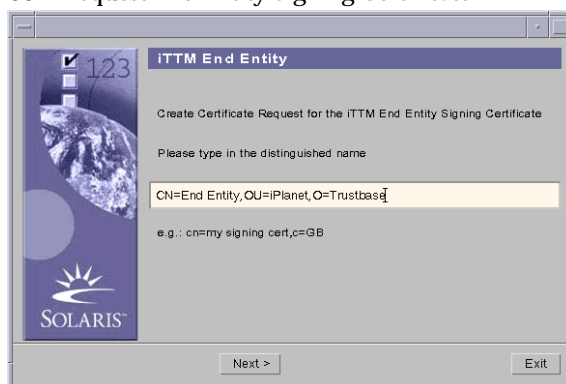
Once the certificate is in the database and you have exited from CertWizard you can delete this filename

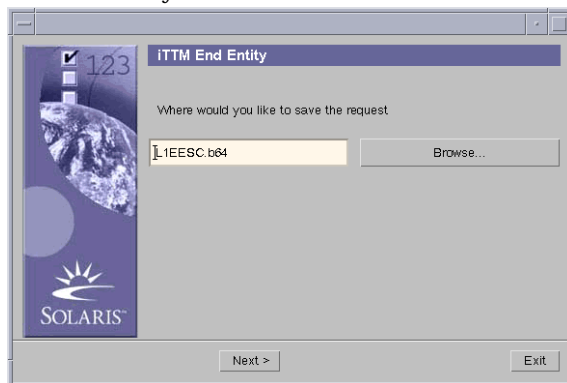
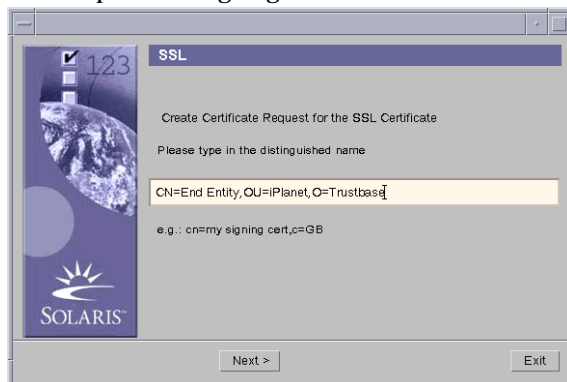


Next you will need to make three requests for signing certificates. This is a two stage process involving pasting data that you have generated from the wizard into the website of the certificate authority that generates a response. The three certificates you need are:

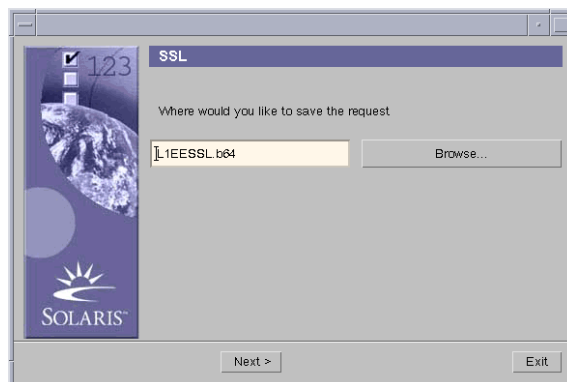
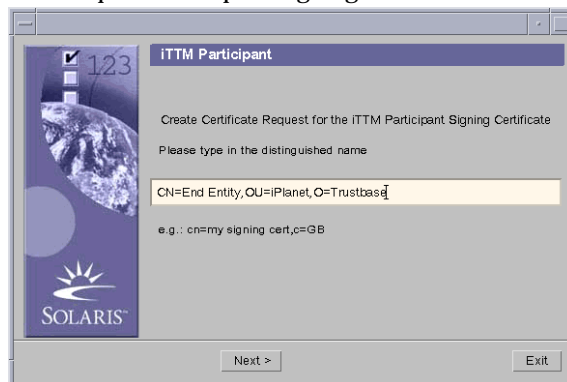
- The End Entity Signing Certificate allows you to sign messages to your customer.
- The Participant Signing Certificate allows you to sign messages that will be sent from one TC to another
- The SSL Signing Certificate allows you to negotiate an SSL connection at the transport level.

**Figure 1-35** Request End Entity Signing Certificate

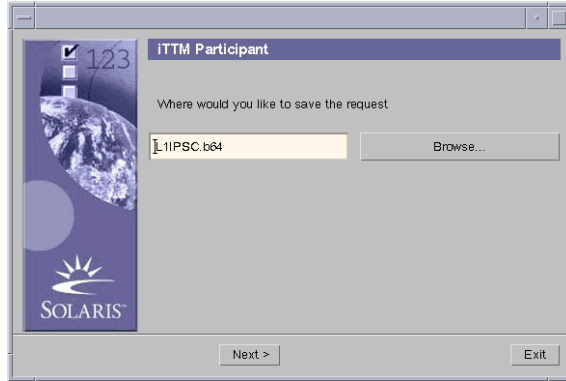


**Figure 1-36** End Entity Location**Figure 1-37** Request SSL Signing Certificate

- This can be any valid distinguished name. Normally this is comprised of:
  - Common Name. Name to give the certificate
  - Organisation Unit. Represents the department or subsidiary of the organisation to which the user/entity belongs.
  - Organisation. Represents the organisation to which the entity/user creating the certificate belongs.

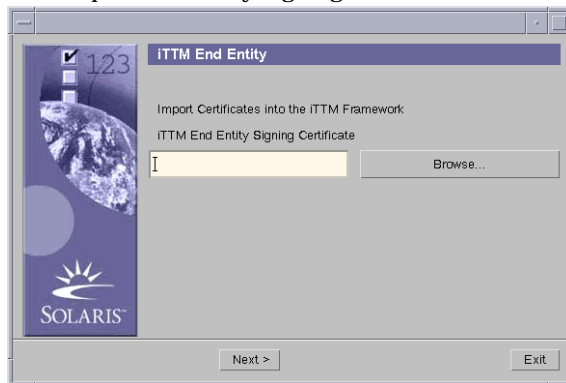
**Figure 1-38** SSL Location**Figure 1-39** Request Participant Signing Certificate

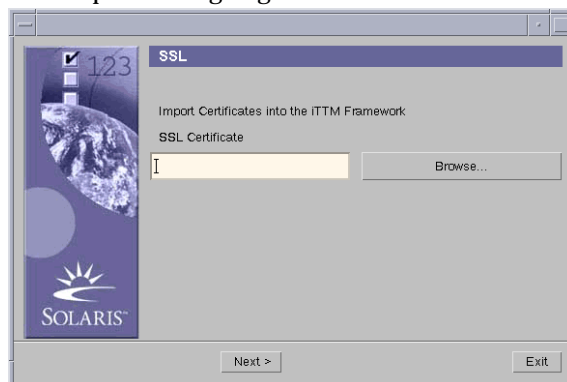
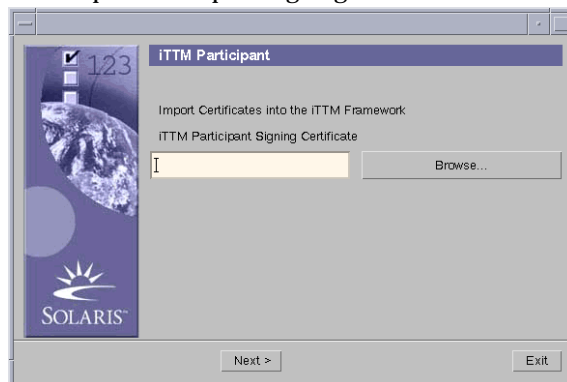
**Figure 1-40** Participant Location



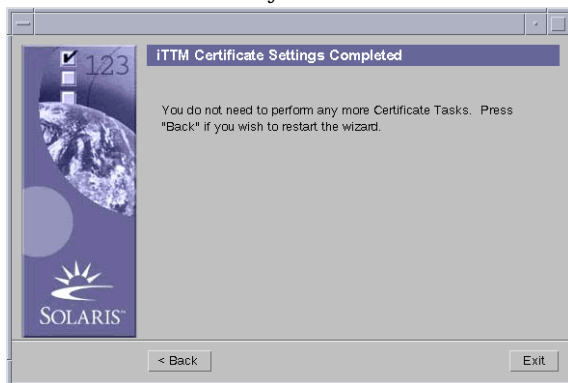
- Go to your CA and submit your PKCS10 request. Collect the response. Save the corresponding response in a file made available to the install wizard.

**Figure 1-41** Import End Entity Signing Certificate



**Figure 1-42** Import SSL Signing Certificate**Figure 1-43** Import Participant Signing Certificate

**Figure 1-44** Certificate Summary

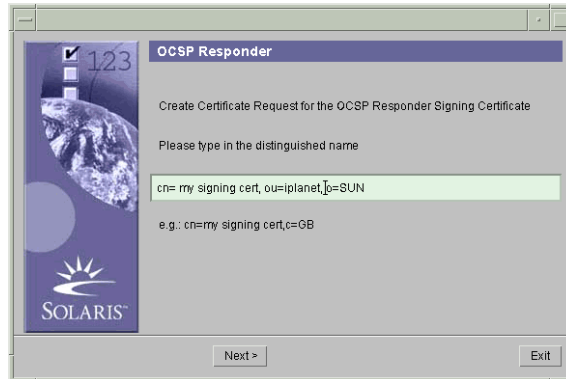


# runOCSPResponderCertWizard

To run OCSP Certificate Wizard type

```
cd /opt/ittm/Scripts  
./runOCSPResponderCertWizard
```

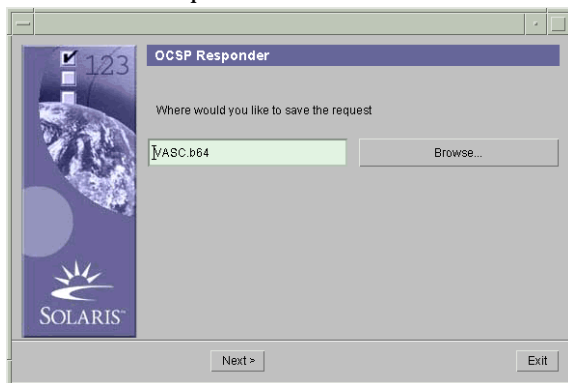
**Figure 1-45** OCSP Certificate request



The OCSP responder signs all of its responses with a private key associated with a certificate. This provides additional security when the OCSP responder is separated from the Trustbase installation.

To generate a certificate with private key for OCSP responses, a certificate request needs to be generated. Enter the distinguished name desired for this certificate and select "Next"

**Figure 1-46** Save the Request as a file

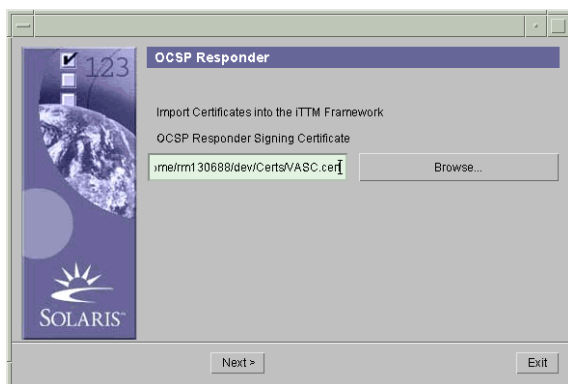


Select a location to store the certificate request file and select “Next”

It will now be necessary to provide this request to the CA. Typically this will be the Identrus Root CA.

Once a certificate has been generated by the CA in response to the request, the certificate should be placed into a file available to the OCSP responder wizard.

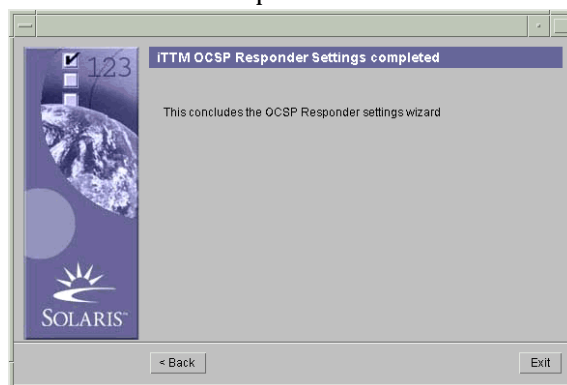
**Figure 1-47** OCSP Certificate Response





Enter the file location of the certificate response and select “Next

**Figure 1-48** OCSP Install Complete



If the certificate is successfully imported, the wizard will complete successfully. If there are any problems then using the “back” button will step through previous screens so they can be re-attempted.

## Secure properties (Optional)

Users installing iTPS should not attempt to do this until they have completed installing iTPS.

Log in under your username e.g. tbase

Property data is held within one of the following properties files:

```
/opt/ittm/myhost/tbase.properties
```

```
/opt/ittm/myhost/identrus.properties
```

The information is divided into property sections, and sub-divided into entries (key/value pairs). In order to access the data the system uses a set of utility classes that only provide read only access. The installation script or the systems administrators are the only entities that make modifications to the properties files. Given that the properties files are also used by the systems administrator it is useful (certainly in installation and set up) that the information is human readable.

This allows certain properties to be re-directed to a sub-file. This sub-file

```
/opt/ittm/myhost/secure.properties
```

is in a known system location and is encrypted.

The properties utility layer will automatically redirect to the encrypted file when the *value* of a property in either tbase.properties or identrus.properties starts with a known re-direction string e.g. PROTECTED\_. The value of a redirected property will start with the known string and everything after this known string will be taken to be the lookup key in the encrypted file.

All properties files will share a single secure.properties file, this file will be located in the same place as all other properties – which is the machine name specific directory of the iTTM installation. secure.properties is encrypted and decrypted using the same password.

Determining which properties to secure will normally depend on a security audit in your organisation.

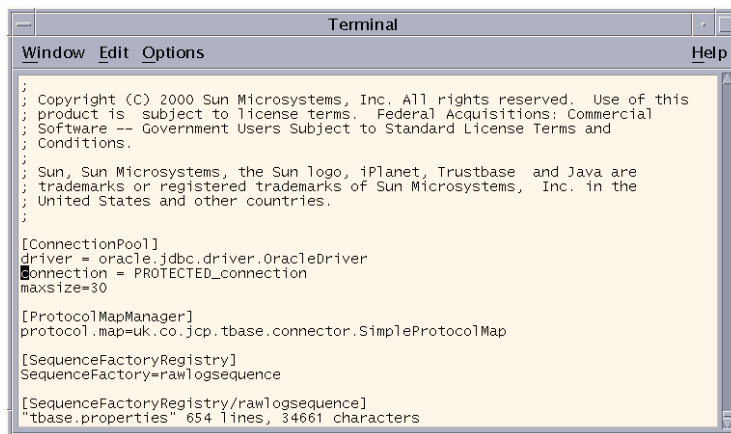
# Installation Procedure

1. Go to machine name directory. Mark the properties you want to secure in the properties file as

= PROTECTED\_name

as illustrated below:

**Figure 1-49** Securing property files (e.g. tbase.properties)



```
Terminal
Window Edit Options Help
;
; Copyright (C) 2000 Sun Microsystems, Inc. All rights reserved. Use of this
; product is subject to license terms. Federal Acquisitions: Commercial
; Software -- Government Users Subject to Standard License Terms and
; Conditions.
;
; Sun, Sun Microsystems, the Sun logo, iPlanet, Trustbase and Java are
; trademarks or registered trademarks of Sun Microsystems, Inc. in the
; United States and other countries.
;
;
[ConnectionPool]
driver = oracle.jdbc.driver.OracleDriver
connection = PROTECTED_connection
maxsize=30

[ProtocolMapManager]
protocol.map=uk.co.jcp.tbase.connector.SimpleProtocolMap

[SequenceFactoryRegistry]
SequenceFactory=rawlogsequence

[SequenceFactoryRegistry/rawlogsequence]
"tbase.properties" 654 lines, 34661 characters
```



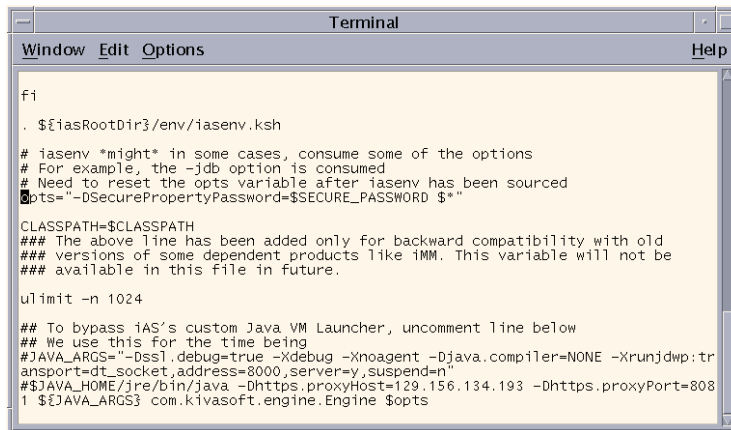
- To facilitate decryption of protected properties in iAS, the KJS script in iAS needs to be modified.

```
/opt/ias6/ias/bin/kjs
```

This takes effect when you run `./startias`.

The change requires that the `opts` variable in the `kjs` script be added to include the `'-DSecurePropertyPassword=$SECURE_PASSWORD'` as illustrated below:

**Figure 1-51** iAS kjs script



```

fi
. ${iasRootDir}/env/iasenv.ksh

# iasenv *might* in some cases, consume some of the options
# For example, the -jdb option is consumed
# Need to reset the opts variable after iasenv has been sourced
opts="-DSecurePropertyPassword=$SECURE_PASSWORD $*"

CLASSPATH=$CLASSPATH
### The above line has been added only for backward compatibility with old
### versions of some dependent products like IMM. This variable will not be
### available in this file in future.

ulimit -n 1024

## To bypass iAS's custom Java VM Launcher, uncomment line below
## We use this for the time being
#JAVA_ARGS="-Dssl.debug=true -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdw:transport=dt_socket,address=8000,server=y,suspend=n"
#$$JAVA_HOME/jre/bin/java -Dhttps.proxyHost=129.156.134.193 -Dhttps.proxyPort=8081 $$JAVA_ARGS com.kivasoft.engine.Engine $opts

```

4. Once the redirection to the secure.properties file is complete, the administrator will run the provided utility (protectProperties) to secure the secure.properties file with password based encryption. A similar tool (unprotectProperties) will be run to return the secure.properties file to its plaintext state to allow further changes to the property values. The scripts are as follows:

```
/opt/ittm/Scripts/protectProperties  
/opt/ittm/Scripts/unprotectProperties
```

The tool will use PKCS#5 to generate an Triple DES key of 168 bits in length from the supplied password. The password can include space characters, to allow a long pass phrase to be selected. Null passwords are not acceptable.

Run the protected property script:

```
cd /opt/ittm/Scripts  
./protectProperties
```

To unprotect using the same password:

```
cd/opt/ittm/scripts  
./unprotectProperties
```

If you forget your password or you type in a different password then decryption will fail and the secure properties become unrecoverable.

You might also consider keeping a backup of the text version of secure.properties in case you forget your password.

5. The startias script

```
/opt/ittm/Scripts/startias
```

will be changed to check for the existence of the secure.properties file. If this file exists, then the password will be elicited from the user – null passwords are not acceptable. Start iAS and enter <password>

```
cd /opt/ittm/Scripts  
./startias
```

6. Start iTTM and enter <password>

```
cd /opt/ittm/Scripts  
./starttbase
```

7. Must delete all copies of silent install answers

```
rm -r /outputdir
```

Passwords in package setting directory must be obscured

`/var/sadm/pkg`

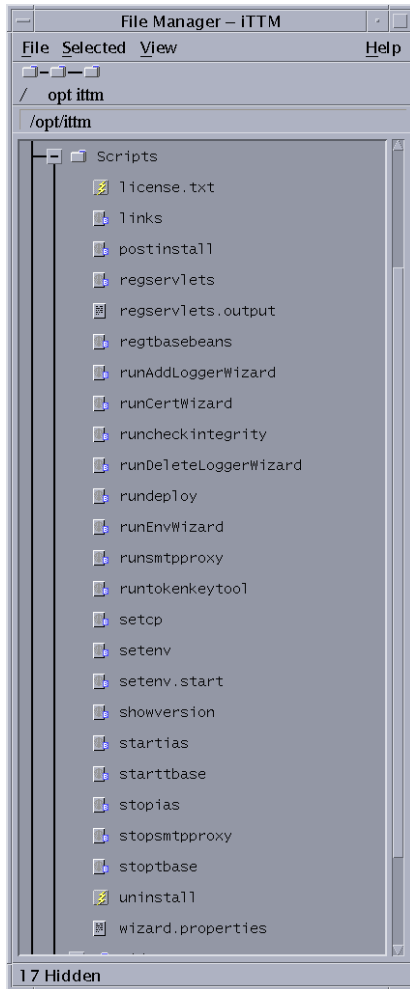
8. Finally, `secure.properties` prevents `-m` and `-s` setup install options from working.

# Start iPlanet Trustbase Transaction Manager

Having completed the installation and configuration iPlanet Trustbase Transaction Manager can be controlled using the scripts available in:

- /opt/ittm/Scripts

**Figure 1-52** iPlanet Trustbase Transaction Manager Commonly Used Scripts





## First Time Start Procedure

The following procedure stops the system.

- Log in under your username e.g. tbase, under a separate window

```
cd /opt/ittm/Scripts
LOGNAME=tbase;export LOGNAME
/opt/ittm/Scripts/stopias
LOGNAME=root;export LOGNAME
/opt/ittm/Scripts/stopias
/opt/ias6/slapd-myhost/start-slapd
/opt/iws6/https-myhost.mycompany.com/start
/opt/iws6/https-admserv/start
./startias
./starttbase
```

## Restart procedure

- Log in under your username e.g. tbase and run the following shell:

```
/opt/ias6/slapd-myhost/start-slapd
/opt/iws6/https-myhost.mycompany.com/start
/opt/iws6/https-admserv/start
cd /opt/ittm/Scripts
./startias
./starttbase
```

- Logon onto iTTM as follows:

<http://myhost.mycompany.com/NASAdapter/logon.html>

# Error checking

## During Installation

If there was a problem during your installation you should check the following error logs

1. Your iTTM install logs

`/opt/ittm/Scripts/*.log`

2. Your iWS error logs

`/opt/iws6/https-myhost/logs/errors`

3. The Java iAS log file

`/opt/ias6/ias/logs/kjs_1_CCS0`

4. Your Oracle iTTM Error tables

`ERRORVIEW`

5. Other logs to look for

`/var/sadm/install/logs`

`/tmp/reg*.output`

## During Operation

1. In the first instance you should go to the Trustbase configuration screens and select from the main menu <Logs> <Error Log Query>

2. Your Oracle iTTM Error tables

`ERROR_SUPPORT`

## Normal Shutdown procedure

iAS, iTTM, Directory Server, Web Admin Server and the Web Server itself, under normal operations can be stopped as follows.

```
/opt/ittm/Scripts/stopias
```

```
/opt/ittm/Scripts/stoptbase
```

```
/opt/ias6/slapd-myhost/stop-slapd
```

```
/opt/iws6/https-admserv/stop
```

```
/opt/iws6/https-myhost.mycompany.com/stop
```

---

**NOTE**      bourne shell (/bin/sh) uses './setcp' and c shell (/bin/csh) uses './setup'

---

# Uninstallation

## A complete Uninstall

### 1. Stop the system

```
./opt/ittm/Scripts/stopias  
./opt/ias6/slaped-myhost/stop-slaped  
./opt/iws6/https-admserv/stop  
./opt/iws6/https-myhost.mycompany.com/stop
```

### 2. Uninstalling your Oracle database may be achieved by deleting all user scheme objects using Drop\_tbaseAll.sql as illustrated below

```
mydatabase% su - oracle  
mydatabase% cd /opt/ittm/current/Config/sql  
mydatabase% sqlplus  
SQL*Plus: Release 8.1.7.0.0 - Production on Fri Feb 15 12:07:11 2002  
(c) Copyright 1999 Oracle Corporation. All rights reserved.  
Enter user-name: tbase  
Enter password:  
Connected to:  
Oracle 8i Enterprise Edition Release 8.1.7.0.0 - Production  
With the Partitioning and Java options  
PL/SQL Release 8.1.7.0.0 - Production  
SQL>spool myoutput.txt  
SQL>set echo on  
SQL>@Drop_tbaseAll.sql
```

### 3. From User:root Uninstall iTTM as follows

```
./cdrom/cdrom0/ittm/setup -u
```

### 4. Uninstalling iAS and iWS can be achieved by following the instructions in the iAS and iWS install guides. For instance,

```
cd /opt/iws6
./uninstall
rm -rf /opt/iws6
cd /opt/ias6
./uninstall
rm -rf /opt/iws6
```

## A partial Uninstall

In some instances you prefer to Uninstall but save your certificates, configurations and property files. Static configuration can easily be backed up, from user: tbase account, by copying the

1. /opt/ittm/myhost/\*.properties files
2. /opt/ittm/store certificate database

to a storage directory. Typically this would involve

```
mkdir /opt/temp
cd /opt/ittm/store
cp -r * /opt/temp
```

3. As mentioned in an earlier section, to export dynamic configuration items for backup use the Config | Export setting in the iTTM admin screens.



This will return a file for download. Some browsers may display this file rather than saving - in this case use View Source to generate a text document containing the exported configuration and save to the desired location.

4. Save your previous install settings ./cdrom/cdrom0/ittm/setup -m /dir
5. repeat previous section on Clean Uninstall

# Reinstallation

## A complete Reinstall

From user: root

1. Follow the instructions for a complete Uninstall
2. Follow the instructions for a standard install

## A Partial reinstall

1. Follow the procedure for a partial uninstall in previous section
2. From user: tbase
  - `./cdrom/cdrom0/ittm/setup -m /dir`
  - `./cdrom/cdrom0/ittm/setup -u`
  - `./cdrom/cdrom0/ittm/setup -s /dir`
  - `cp /temp/*.properties /opt/ittm/myhost`
  - `cp -r /temp/store/* /opt/ittm/store`
  - `cp -r /temp/xmlconfig`
3. From the iTTM configuration screen select <config><import>

Note Please consult the iPlanet Application Server Installation, iPlanet Web Server, Oracle and nCipher documentation for procedures on how to reinstall. Note under normal circumstances when reinstalling iTTM you should not need to reinstall iWS 6.0 and iAS 6.5.



# Identrus Authorisation

In order to send Identrus Messages to iPlanet Trustbase Transaction Manager you will need to create an Authorisation for your own customers that allows the sending of Identrus Enabled Messages. These are illustrated in the next three figures.

- The following TokenKeyTool (see appendix at the end of this guide) command will list you your Identrus root certificate

```
cd /opt/ittm/Scripts
./runtokenkeytool
listcerts -alias IRCA
```

- You will need the serial number and the Issuer DN for two certificates that need to be added to iTTM
  - a. A IRCA certificate
  - b. A L1CA certificate
- Login to iTTM as Administrator
- Select <Authorisation> <Add Certificate>



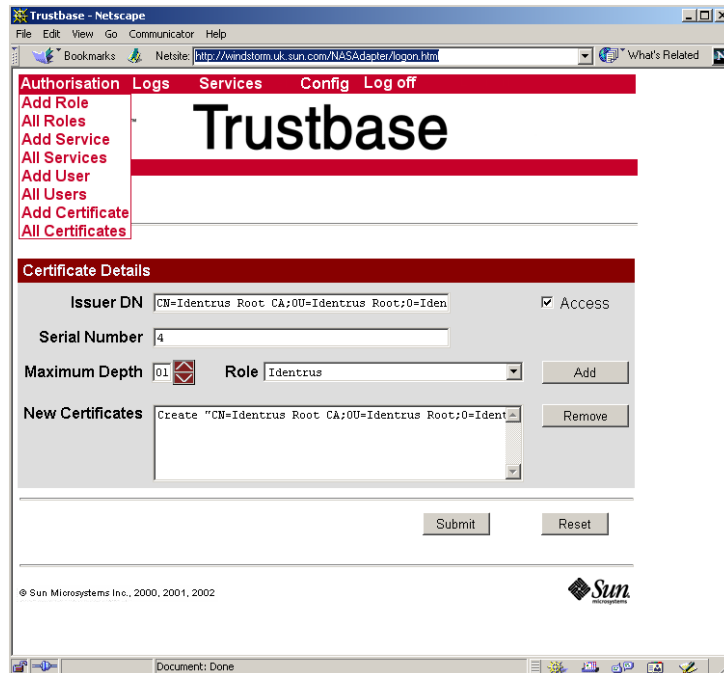
Note Remember that the serial number shown in tokenkeytool is in Hexidecimal. For example, to convert 0x1000, copy and paste the entire hexidecimal into the iTTM screen and iTTM will convert it automatically for you.

If you just use IRCA certificate with depth 2, any customer from any bank can send messages to the Trustbase. If you use IRCA with depth 1, and L1CA with depth 1, only customers of your bank and other bank TC's can communicate with the Trustbase. This is a normal and correct deployment.



- For this example enter the issuer Distinguished Name "CN=Identrus Root CA;OU=Identrus Root;O=Identrus;C=GB" and the serial number of the RP Bank CA which is 4. Set the Max Depth to 1 and the Role to Identrus. Note, see also "Adding a Certificate," on page 169 on how to set the MAX depth and what the checkbox access is for. This is illustrated below:

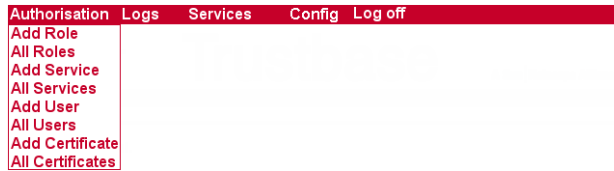
**Figure 1-53** Installing a Certificate so as to send Identrus Messages within Trustbase



If the Access checkbox is not ticked then this user would not be allowed to access iTTM.

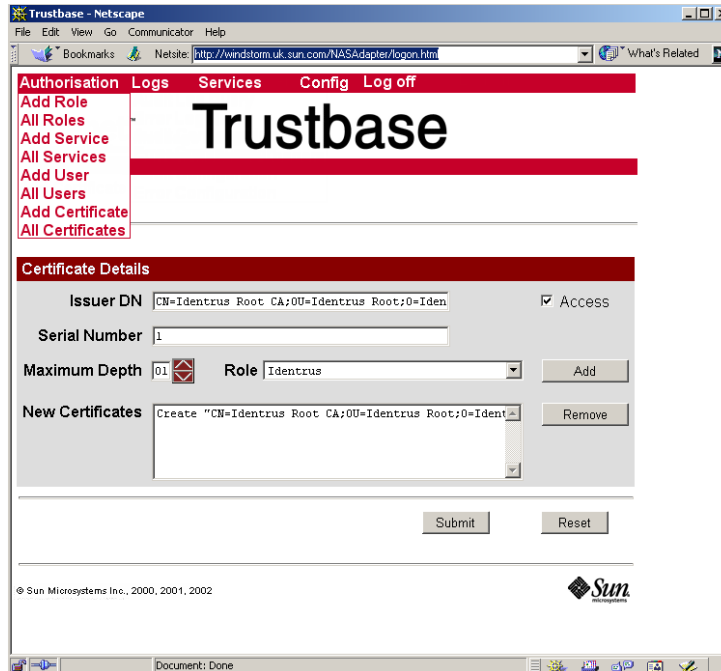
You will also need to create an additional Authorisation for other banks to send Identrus Enabled messages to you. These are illustrated in the next three figures:

- Select <Authorisation> <Add Certificate>



- Enter the Distinguished Name "CN=Identrus Root CA;OU=Identrus Root;O=Identrus;C=GB" and the serial number of the RP Bank CA which is 1. Set the Max Depth to 1 and the Role to Identrus. Note, see also "Adding a Certificate," on page 169 on how to set the MAX depth and what the checkbox access is for. This is illustrated below:

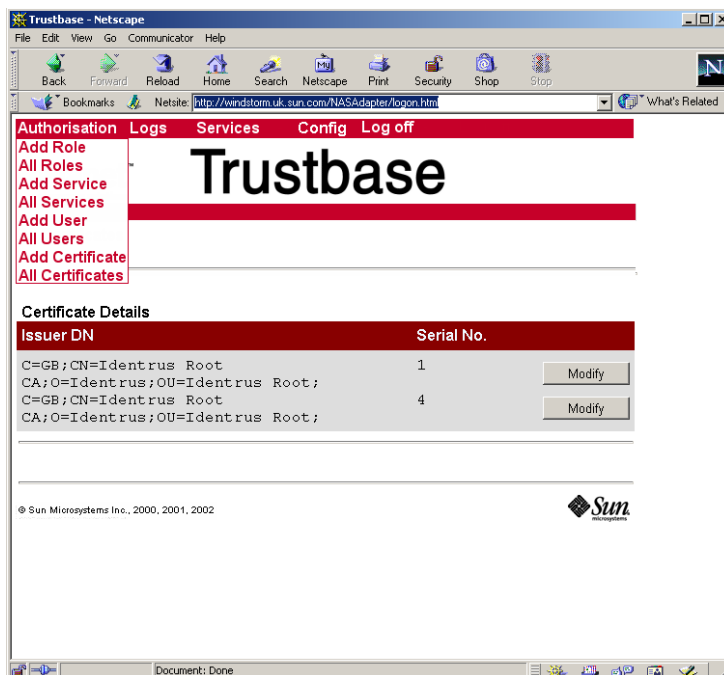
**Figure 1-54** Installing a Certificate within Trustbase for sending Identrus Enabled Messages



Note the DN used is the Issuer DN and therefore it is the same for both L1CA and IRCA (L1CA issued by IRCA, IRCA issued by itself). Only the serial number is different.

Finally you should restart iPlanet Trustbase Transaction Manager for the setting to take effect and you should make sure both certificates are installed within iPlanet Trustbase Transaction Manager as illustrated below:

Figure 1-55 Identrus Enabled Messages installed within Trustbase



# TokenKeyTool Checklist

The following checklist provides you with a summary of the kinds of commands available within TokenKeyTool that is used to help you set up your Identrus Authorisation.

Information Type	Example Set-up Value for TokenKeyTool
To run TokenKey-Tool	<pre>cd /opt/ittm/Scripts ./runTokenKeyTool</pre>
To list Trusted Certificates with an Alias	<pre>listcerts -alias IRCA</pre>
To list all trusted Certificates	<pre>listcerts</pre>
To list private keys and their associated Certificate chains	<pre>listkeys</pre>
To delete private keys and its associated Certificate chains	<pre>deletekey -issuer "CN=End Entity" -serial 0x1000</pre>
To obtain help	<pre>help</pre>
Documentation	See the appendix at the end of this guide



# Architectural Configuration

iPlanet Trustbase Transaction Manager can be deployed over a variety of hardware configurations:

- It may be configured on single and clustered iPlanet Application Server installations
- It has a variety of configuration issues within the DMZ environment
- It can also be configured using Hardware Security Modules

# iPlanet Application Server configuration

iPlanet Application Server may be used in a wide variety of configurations to support differing requirements for:

- Scalability
- Throughput
- Failover

The iPlanet Trustbase Transaction Manager has been designed to take advantage of these facilities and has been tested on two of the main configurations, these being:

- Single iPlanet Application Server
- Clustered iPlanet Application Servers

A single iPlanet Application Server is the most likely configuration option for low volume pre-operational Transaction Manager environment. This is considered a standard iPlanet Trustbase Transaction Manager installation and requires no specific configuration options other than those outlined in the iPlanet Application Server installation guide. The recommended settings for iPlanet Application Server are:

- Single KJS for each CPU on the host machine
- Minimum 8 and Maximum 64 Threads per KJS

The iPlanet Trustbase Transaction Manager is generally a CPU bound processing environment. This means that installing a greater number or faster CPU's will improve performance.

The iPlanet Trustbase Transaction Manager utilises Oracle databases extensively. Oracle itself is both CPU and disk intensive. If possible, Oracle should be located on a separate computer to the iPlanet Trustbase Transaction Manager installation.

Clustered iPlanet Application Server installations provide a means of improving Scalability, throughput and fail-over over a single iPlanet Application Server running the iPlanet Trustbase Transaction Manager. Prior to installing a clustered iPlanet Application Server the following items require consideration:

- Each Machine running iPlanet Application Server can have a nCipher HSM. This is a requirement for Identrus Compliance.
- A clustered iPlanet Application Server environment may provide slower response times in marginal loading conditions



- 
- NOTE** For further information about performance tuning and effective deployment consult:
- Oracle 8.1.7 Administrators Reference Manual Chapter 2
  - SQL tuning can also be found in the Oracle 8.1.7 programmers Guide
  - Tuning Server Performance is discussed within the iPlanet Web Server 6.0 Administration Guide
  - Application Deployment is discussed within the iPlanet Application Server 6.5 Administration and Deployment Guide
-

## Using a DMZ

The general architectural model for deploying an iPlanet Trustbase Transaction Manager is to place the application server within a Demilitarised Zone (DMZ) created using a proxy machine between two firewalls. This configuration provides a means of ensuring that an outside user cannot directly access or modify the logic that forms the application in order to circumvent the authentication and authorisation requirements.

The DMZ primary firewall must offer two unauthorised open ports to support SSL access and SMTP access. These ports are generally:

- SSL - Port 443
- SMTP - Port 25

Behind this firewall is a single machine that runs the SSL proxy, the SMTP listener, and the iPlanet Web Server. The secondary firewall has three ports open configured for the DMZ machine only. These three ports are:

- iPlanet Application Server Directory Port 389
- iPlanet Web Server
  - HTTP - Port 80
  - Admin - Port 8888
- Oracle

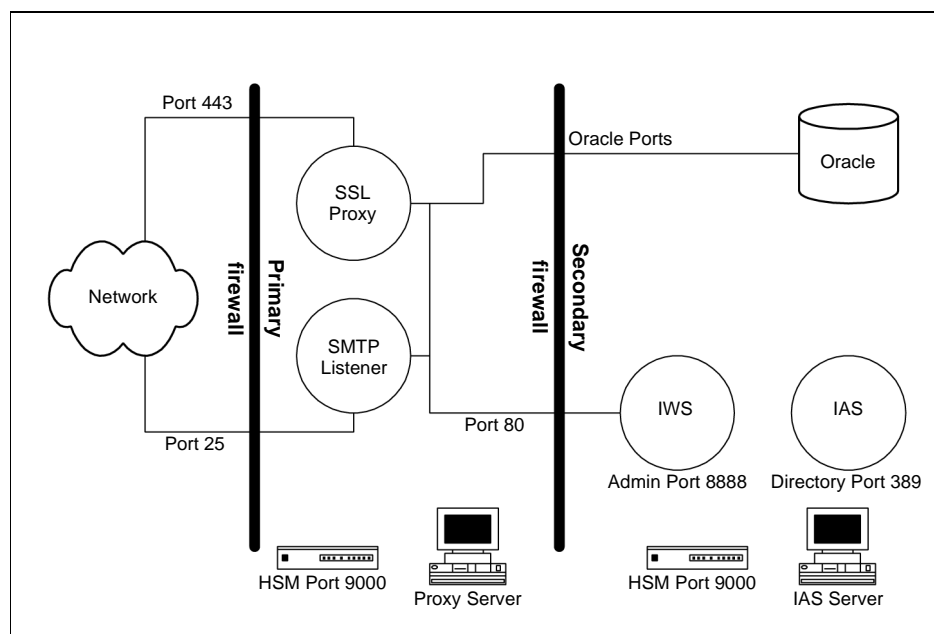
---

**NOTE** Oracle uses many ports. Consult your Oracle DBA about this.

---

The HTTP port is used by the SSL and SMTP proxies to communicate to the iPlanet Web Server located behind the secondary firewall. Both the SSL and SMTP listeners use the Oracle port to store information about connections as well as receive their configuration. The architecture is shown below.

**Figure 2-1** DMZ Architecture



Using this configuration the systems administrators may use the HTML based configuration screens without the need for SSL certificates. This does not circumvent the authentication mechanisms as the configuration management mechanisms are authenticated using a Username and Password based authentication mechanism.

---

**NOTE** The default configuration is such that it all runs on one machine. When you install iPlanet Application Server 6.5 and iPlanet Web Server 6.0 it installs a directory server that has a default port of 389 and an administrator port of 8888.

---

# Machine Installations

In some instances it may be necessary to install product component software on different machines. The following table summarises possible combinations:

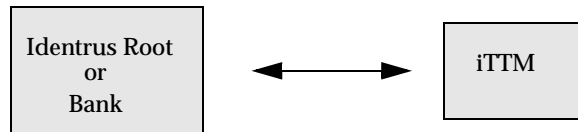
**Table 2-1** Acceptable Machine Installations

Product Component to be separated	Separate Machine Installation?	Considerations
iTTM separated from iAS	Not Possible	If iAS and iWS are installed on Separate machines make sure iTTM is installed with iAS
iWS separated from iAS	Yes	See Section on Configurator Plug-in <a href="http://docs.iplanet.com/docs/manuals/ias.html">http://docs.iplanet.com/docs/manuals/ias.html</a> and also <a href="http://docs.sun.com/source/816-5788-10/app1.htm#13421">http://docs.sun.com/source/816-5788-10/app1.htm#13421</a> The webserver's documents directory must be made accessible by the iTTM installer. This accessibility only needs to be present when the installer is functioning ( i.e. it will need to be re-instated during upgrade or silent installs). When configuring URL rewrites magnus.conf and obj.conf need to be accessible to both machines
Oracle separated from iTTM	Yes	See Section on "Oracle Database Configuration," on page 78
SSL Proxy separated from iTTM	Yes	See Section "HTTP/HTTPS Proxy Implementation," on page 141
SMTP Proxy separated from iTTM	Yes	See "Configuring the SMTP Proxy Separately," on page 145
iAS with other Web Servers	Not Supported	

# HTTP/HTTPS Proxy Implementation

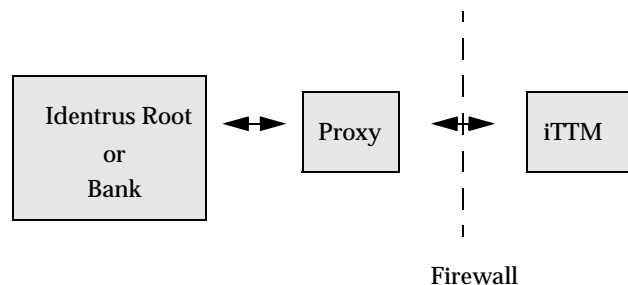
The standard out of the box configuration is illustrated in the diagram below. iTTM communicates directly with the outside world over an HTTP socket.

**Figure 2-2** Standard iTTM installation



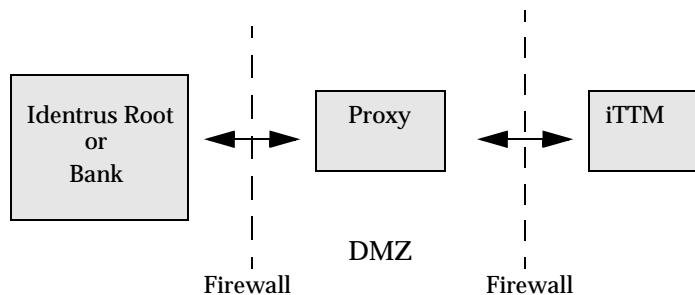
When the computer hosting the iTTM server has a direct connection to the outside world, malicious users may attack the server and gain access to its functionality. To prevent this, a firewall can be installed that blocks access except via a known channel (e.g. only HTTP inbound access on port 80). To provide a further level of protection, a proxy server may be implemented outside the firewall. This accepts and creates connections on behalf of the server, communicating over a channel that only it has access to.

**Figure 2-3** Proxy with single firewall



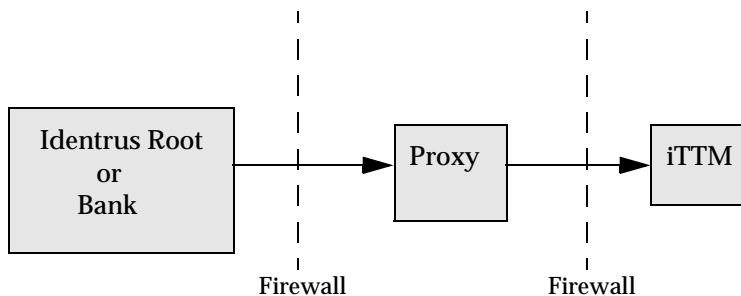
A further layer of security may now be put in place by placing the proxy behind a second firewall. This has the effect of stopping the proxy from receiving unwanted requests.

**Figure 2-4** Proxy with two firewalls



This has sometimes been referred to as a DMZ (demilitarised zone). In most deployments inbound and outbound requests will be handled by separate proxies on separate machines. In the case of inbound proxies, the proxy intercepts requests to the iTTM server and forwards them to the iTTM server along a secure channel.

**Figure 2-5** Example Inbound Proxy

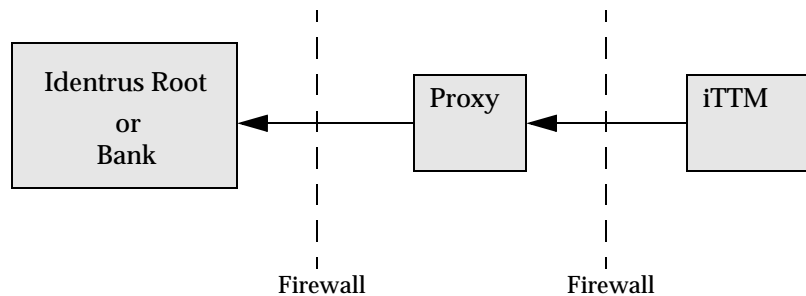


Since iTTM sees an inbound request as identical to either the proxy or the Identrus root or Bank, no iTTM settings need be changed for inbound proxies.

## HTTPS Tunnelling

In the case of Outbound proxies iTTM acts as the SSL Client and utilises the proxy as a 'tunnel' to the outside. When proxying, it is important to know which machine is the trusted client. If the iTTM is the trusted client then all SSL client certificates must come from iTTM and the proxy will not be able to decode the contents of the messages. In this situation SSL tunnelling is used to pass the encrypted communications packets through the proxy. In this case iTTM needs to know where to send requests.

**Figure 2-6** Example Outbound Proxy



For outbound proxies the following parameters need to be added in the section [XURLHttps] within:

```
/opt/ittm/myhost/tbase.properties
```

For example:

```
[XURLHttps]
proxyhost=myhost.mycompany.com
proxyport=1312
notunnelling=false
```

Note that this section already exists in tbase.properties and as such should be modified rather than created from scratch.

## HTTPS Forwarding

In the case that the proxy is the trusted client then SSL certificates must be stored on the proxy machine. iTM will then communicate with the proxy on an encrypted channel and the communication is connected to a secure channel at the proxy. In such circumstances, `tbase.properties` should be modified as follows:

```
[XURLhttps]
proxyhost=myhost.mycompany.com
proxyport=1312
notunnelling=true
```

Note that this section already exists in `tbase.properties` and as such should be modified rather than created from scratch.

## Default Settings

The following default settings are used if they are not explicitly set:

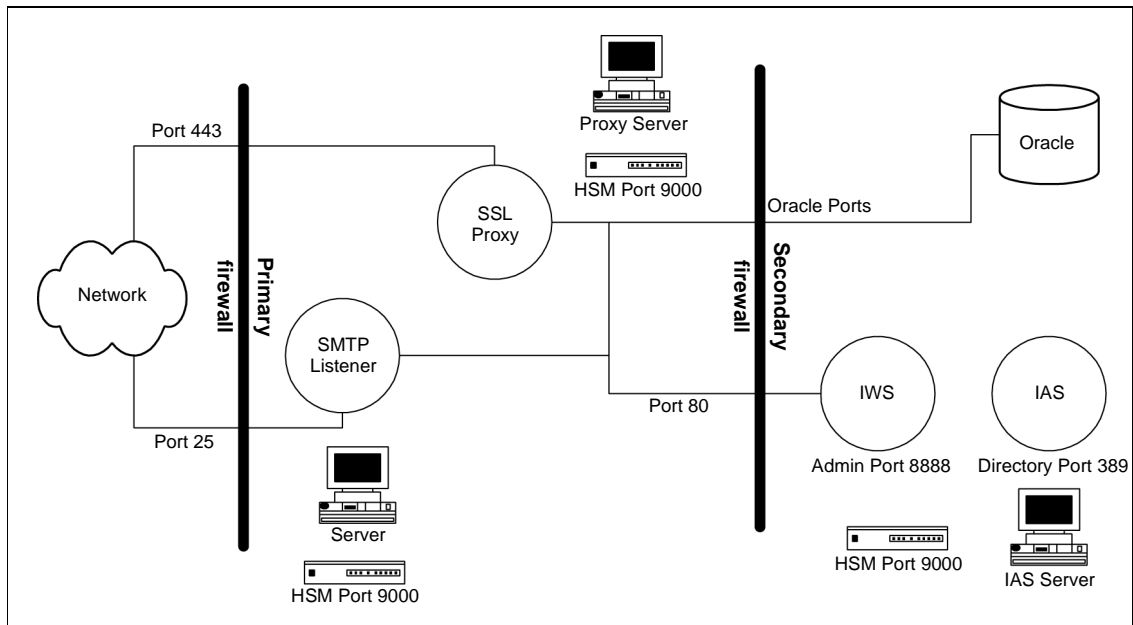
```
proxyport=443
notunnelling=false
```



# Configuring the SMTP Proxy Separately

In some situations it may be more convenient to place the SMTP proxy on a separate machine.

**Figure 2-7** DMZ Architecture for separating the SMTP Proxy



Using this configuration the systems administrators need to do the following. Configure nCipher Security world so that the nCipher boxes share the nCipher security world. A separate script is needed to ensure the appropriate iPlanet Trustbase Transaction Manager software is on both machines by performing the following steps:

1. Create a tar of a completed single machine install by typing `tar -cvf Trustbase.tar Trustbase` from the installation directory (`/app`)
2. Unpack this tar file on the computer you wish to run the proxy on. Unpack it into the same install directory structure as the original computer. (e.g `/app`). Do not install iPlanet Trustbase Transaction Manager, iPlanet Web Server or iPlanet Application Server on this new machine.

3. Enter the directory `<new_install_directory>/ittm` and rename the directory that is named after the hostname of the computer and ensure it is the same as the new host. If the `<new_install_directory>` is different, edit the file `<new_install_directory>/ittm/Scripts.setenv` and change the directories names in `TBASE_INSTALL` and `TBASE_HOME` to refer to the `<new_install_directory>`
4. Edit the script `<new_install_directory>/ittm/Scripts/runsmtpproxy` to change the localhost to the machine hostname where iPlanet Trustbase Transaction Manager has been installed, as illustrated below:

```
#!/bin/sh
. ./setcp
ulimit -n 128
echo $$ > pids/runsmtpproxy.pid
cd $TBASE_INSTALL
exec java uk.co.jcp.tbasetimpl.smtp.server.SmtpServer -debug 6 -url
http://myhost.mycompany.com/NASApp/TbaseSmime/SmimeServlet -timeout
120000
```

5. To start the proxy on its own, enter this new hostname directory `<new_install_directory>/ittm/Scripts`. Run the `./runsmtpproxy` script.
6. On the host that is now no longer running the SMTP proxy, edit the script in `<install_directory>/ittm/Scripts` entitled `./starttbase` and remove the reference to the SMTP Proxy. You will want to do the same with the `./stoptbase` script.

# iAS and iWS on separate machines

As the iWS is on a separate machine, during installation of iTTM, the installer will not be able to locate the iWS docs directory. In this case, supply an empty directory of your choice and follow these steps after installation of iTTM:

1. `tar -cvfh docs.tar <temp-dir>` the temporary docs directory
2. ftp to the iWS host and place the docs.tar in `/opt/iws6/docs`
3. on the iWS host, untar the docs.tar with `tar -xvf docs.tar`

You should also consult the Section on Configurator Plug-in  
<http://docs.sun.com/source/816-5788-10/app1.htm#13421>

## URL Rewrites with iAS and iWS on separate machines

When deploying iTTM using the iAS and iWS on separate machines, the following should be kept in mind:

1. The URL Re-writing has to be done for the Web Server (magnus.conf & obj.conf ). However the tb-urlrewrite.so file is located in the iAS installation

```
/opt/ittm/current/Config/WebserverSetup/JWS/plugins/tb_url_rewrite.so
```

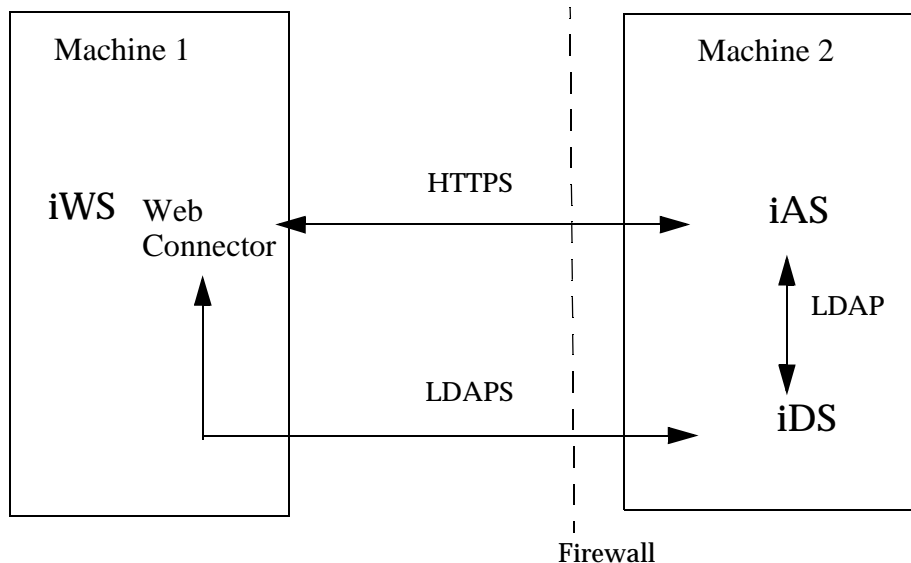
therefore you need to FTP this into

```
/opt/iws6/plugins/lib
```

2. Care should be taken in referencing the TC (either in AIA or using /etc/hosts file), such that the entry is through the iWS machine

# LDAPS

The webconnector is the portion of iAS which intercepts servlet requests to the webserver. This sits on the same machine as the webserver and communicates with an LDAP directory to obtain information such as the location of the application server and names to be interpreted as servlet requests. In some circumstances, it may be desirable to encode communications between the LDAP directory server and the webconnector. This is accomplished by setting up an SSL communication channel to the LDAP server.



This is known as LDAPS. For further background reading about this, you should consult:

1. Setting up SSL in LDAP

<http://docs.sun.com/source/816-5606-10/ssl.htm#996824>

2. To Configure Registry Entries consult iPlanet Application Server Administration Guide

<http://docs.sun.com/source/816-5784-10/adconfig.htm#25855>

3. iPlanet Directory Server Administration Guide

<http://docs.sun.com/source/816-5606-10/index.html>

#### 4. Webless installs

<http://docs.sun.com/source/816-5788-10/app1.htm#13421>

In environments where security is important, often iAS is run behind a firewall, with iWS sat on a machine outside the firewall or in the DMZ. In the setup, the webconnector component must talk to the directory server for some of its configuration and, since it is outside the firewall, it may be desirable to make this communication secure. Out of the box, iAS does not support secure LDAP transaction. However, iTTM supplies a patch which may be installed to provide LDAPS for the webconnector only.

1. Separate iWS and iAS. See chapter in iAS manual on Webless Installs

<http://docs.sun.com/source/816-5788-10/app1.htm#13421>

2. The version of the application server supplied with iTTM does not support LDAPS out of the box. A patch is provided to enable LDAPS with iAS6.5 To install the upgrade untar the patch and copy the libraries provided in ias/gxlib to your ias install.

```
cd /opt/ittm/current/Config/WebserverSetup/patches
tar xvf ldaps_ias6_5.tar
cp ldaps_ias6_5/ias/gxlib/*.so /opt/ias6/ias/gxlib
rm -r ldaps_ias6_5
```

Note: these .so files need to be put in the /opt/ias6/ias/gxlib of the machine that has the Webserver & Connector.

3. Setup the directory Server for LDAPS

- a. Configure the directory server on the iAS machine for SSL
- b. Consult iPlanet Directory Server Administration Guide

<http://docs.sun.com/source/816-5606-10>

- c. To install a certificate, use the certificate manager from the iDS administration console. This will allow you to request a certificate from your CA. This should be marked as usable by an SSL Server
- d. You will also need to install the trusted CA cert chain in the CA Certificate section of the console.
- e. After you have installed your certificate, turn on encryption by clicking the Enable SSL checkbox in the Encryption tab. We currently do not support Client Authentication so select any Client Authentication option except "required"

- f. Ensure that you have an SSL3.0 cipher selected rather than TLS.

#### 4. Checking LDAPS.

- a. To check that LDAPS is setup correctly use Netscape Navigator. This has built in LDAP and LDAPS support.
- b. Go to the Location box on the tool bar and type in a URL of the following form. we illustrate a default example where myhost.mycompany.com is the DNS/IP of your directory server and port 389 is the unencrypted port number.

`ldap://myhost.mycompany.com:389/`

- c. You should be talking to the unencrypted port here just to confirm that the directory server is up and running. If everything is as it should be, we will get a display of the directory server details. If you cannot connect, check all of your settings and/or try restarting the directory server
- d. To test secure communication, we need to inform Netscape to accept the SSL certificate as trusted. Go to your CA and import its certificate to your Netscape browser. Now try the following URL structure

`ldaps://myhost.mycompany.com:636/`

- e. If the connection is working correctly then the directory server details should be displayed.

5. To configure the webconnector for secure LDAP connection, the iAS registry must be edited using kregedit. The following values must be added or modified

- a. Integer:

```
SOFTWARE/iPlanet/Application  
Server/GDS/Backends/LDAP/<name_of_backend>/secure = 1
```

- b. String:

```
SOFTWARE/iPlanet/Application  
Server/GDS/Backends/LDAP/<name_of_backend>/0/Host=<hostname of  
iAS machine>
```

```
SOFTWARE/iPlanet/Application  
Server/GDS/Backends/LDAP/<name_of_backend>/0/Port=<Encrypted  
port of iAS machine>
```

```
SOFTWARE/iPlanet/Application  
Server/6.5/CSSO/HTTPAPI/GXIP=<IPAddress of iAS machine>
```

**kregedit can be invoked by typing the command:**

```
./opt/ias6/ias/bin/kregedit
```

**These settings are illustrated below**



Figure 2-8 LDAPS settings

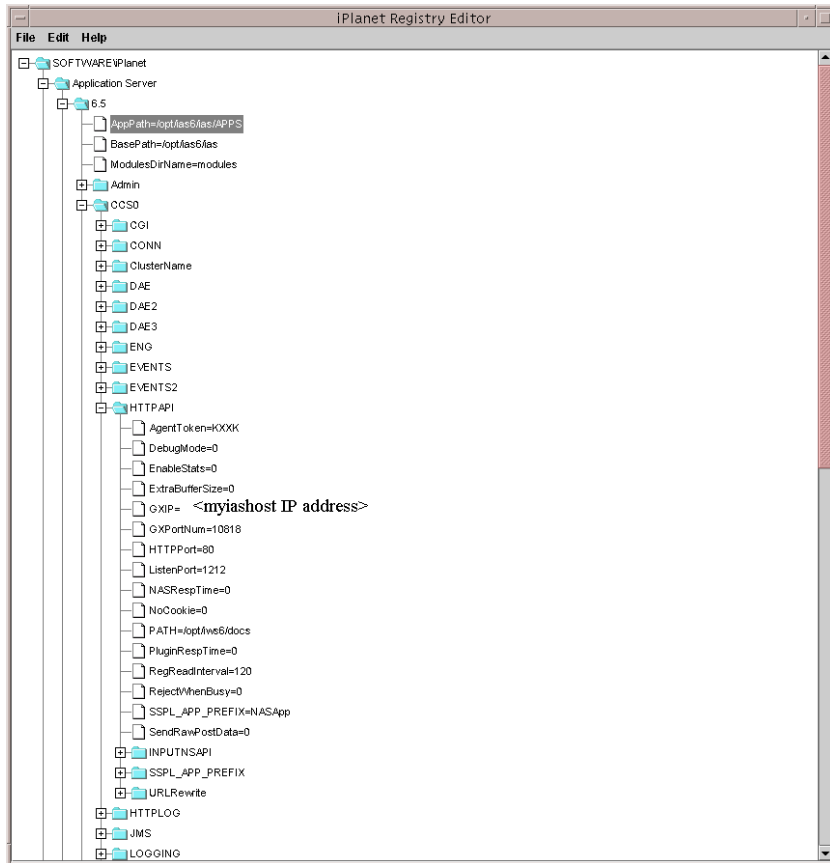
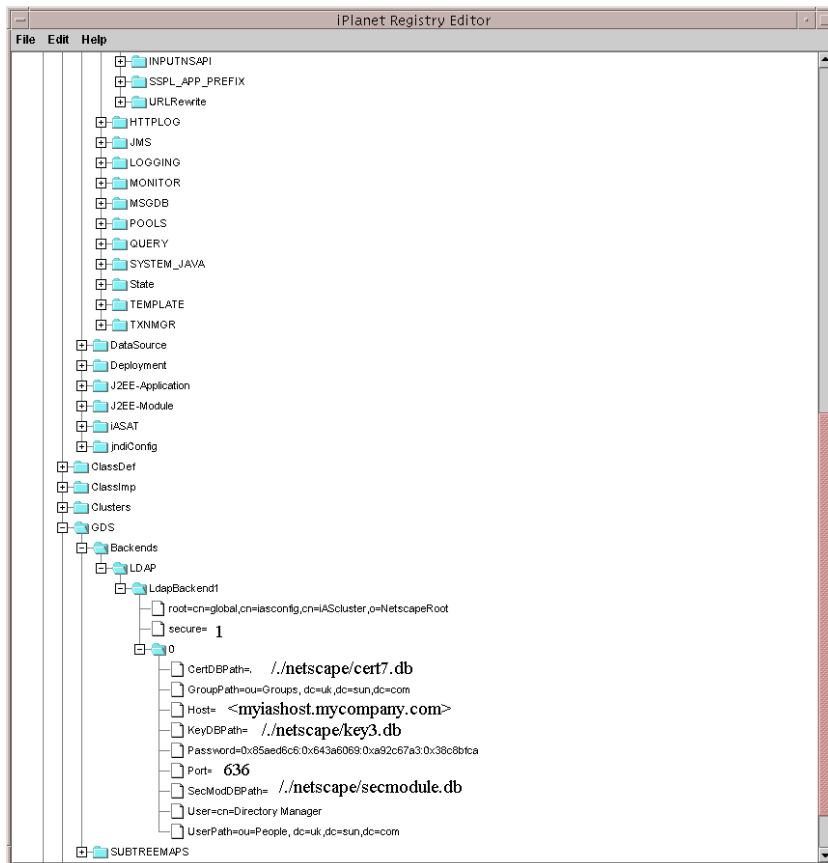


Figure 2-9 More LDAPS settings



Security can be turned off merely by setting the key `secure = 0` and setting the port back to 389 i.e.

```
SOFTWARE/iPlanet/Application
Server/GDS/Backends/LDAP/<name_of_backend>/secure = 0

SOFTWARE/iPlanet/Application
Server/GDS/Backends/LDAP/<name_of_backend>/0/Port=<Unencrypted
port of iAS machine>
```

6. Finally, install the CA certificate (as used in step 4) into the web server certificate database.
7. Now restart the web server and check the error log. This should inform you that LDAPS is enabled and that the LDAP connection is successful.

## Technical Note on LDAP

The LDAP client does not support TLS, your LDAP server must use SSLv3.0

kregedit does not have LDAPS support. Therefore, once LDAPS is turned on, kregedit will no longer run. This can be dealt with by editing `ias/registry/reg.dat` and replacing the port value with the unencrypted port (389) & setting the secure value = 0 kregedit will now work again. Make your changes using kregedit and then change the port and security back to the secure version.



# Logging on

Once you have completed your iPlanet Trustbase Transaction Manager Installation you are ready to logon on and consider all your various configuration options. This involves authorising services, users, roles and certificates. Defining your logging options. Setting your SSL transport configuration options. Deploying your services and defining configuration interaction options via templates.

## Configuration Management overview

In order to inspect or modify items within a iPlanet Trustbase Transaction Manager installation the administrator must log in using the HTML Logon form:

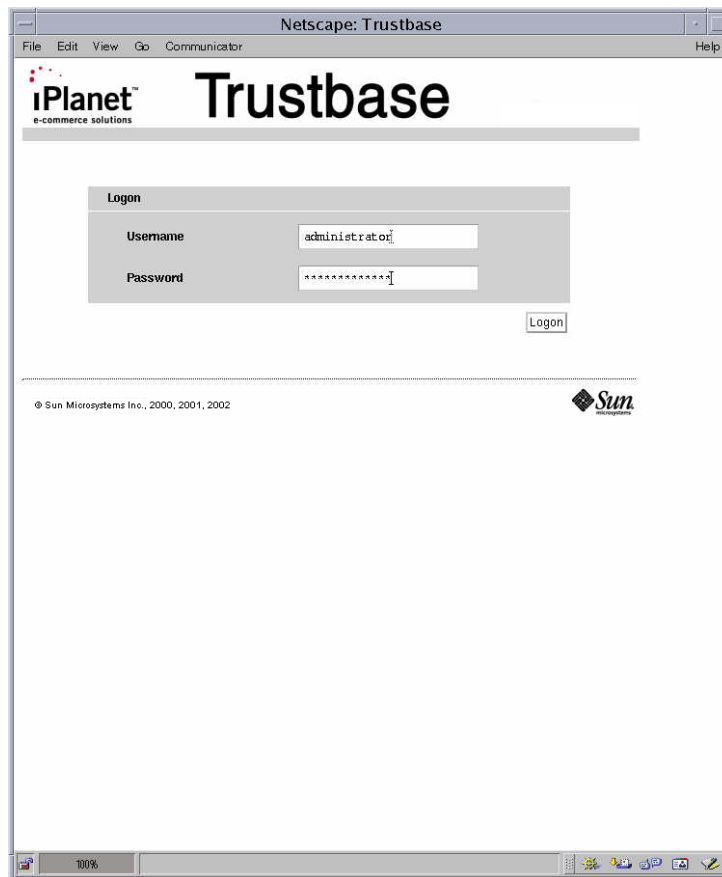
```
http://myhost.mycompany.com/NASAdapter/logon.html
```

The Administrator may perform a variety of configuration operations each time returning to the appropriate home screen. Selecting logout on the Home Screen will terminate the operator's session and require re-authentication to use any of the configuration screens. Once logged out of a session it is not possible to configure iPlanet Trustbase Transaction Manager using HTML screens cached in the browser as the administration authentication context has been removed from the iPlanet Trustbase Transaction Manager server. If an administrator leaves the session logged in for longer than several minutes without any activity, the session is automatically terminated and the administrator must re-authenticate using the login screen to gain access to the system. If the session times out, operating any buttons on the configuration screens will result in a server error being returned to the users browser. It is necessary to login to iPlanet Trustbase Transaction Manager again.

# Logon Screen

The iPlanet Trustbase Transaction Manager is installed with one standard username. All administrators enter the system using the Logon Screen shown below:

**Figure 3-1** Logon Screen



The username and password have been set as illustrated below. This can be changed by selecting the appropriate menu option (see section “Adding users to roles,” on page 167).

```
Username administrator
Password administrator
```

---

**NOTE** This password has the same implications as a root password on a Solaris machine and as such you would need to contact support.

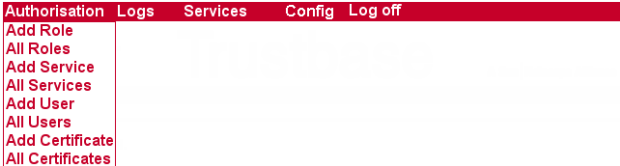
---



# Configuration Options

The following Options are available:

- Authorisation - defines roles, services, users and Certificates



- Logs - allows you to query, configure and view messages that are thrown round the system



- Services - allows you to register and deploy services



- Config - allows you to import and export xml configuration settings



- Log Off - allows you to exit from the configuration screens.





# Authorisation

Authorisation revolves around the idea of authenticating a service by assigning a role that can authenticate aspects of a service by linking a certificate to a role. Each certificate is assigned a distinguished name and a role that can authenticate to any number of levels down a certificate hierarchy.

# Introduction

The iPlanet Trustbase Transaction Manager authorisation facilities provide the operator with the ability to prevent unknown users accessing services. The authorisation management screens allow the operator to modify the set of known users, and the services they are allowed to access. Changes in the authorisation screens modify the iPlanet Trustbase Transaction Manager authorisation database and are made immediately.

The iPlanet Trustbase Transaction Manager will perform an authorisation check on every request that starts a new user session e.g. when the operator logs on to the system or when a CSC is received. This check maps a username or certificate to a role, and this role is passed around the system with the request. Prior to the router invoking a service the authorisation database is checked to see which roles are allowed to access the service. If the roles match then the operation is allowed. If there is a mismatch then the router will log an authorisation failure and the request will be rejected.

All of the facilities required to organise the authorisation parameters are accessible from the main authorisation menu shown below.

**Figure 4-1** Authorisation Main Menu

Authorisation	Logs	Services	Config	Log off
Add Role				
All Roles				
Add Service				
All Services				
Add User				
All Users				
Add Certificate				
All Certificates				

# Authorising users to access a service

Authorising users for a particular service is a multistage process. The steps are:

- Define a role - Create a group under which the users will be identified
- Add users to the role - Identify the individuals that will be members for the group
- Map the role to the service - Provide an authorisation to use a service if they hold a particular role
- Allocate a certificate to each role.

The following sections describe this process in more detail.

## Defining a role

Selecting <Add role> from the main menu will provide a form that contains the following:

- Name - The text label for the new role
- Description - Free text description of the role. This is not used by the authorisation mechanisms, but is useful for describing the use of a particular role
- Active - Unchecked means the router will throw an unauthorised response even if the role to service link is correct.

Submitting the form will update the authorisation tables immediately. After adding a new role, using the view role option from the authorisation menu will contain the updates immediately. The view roles option on the main menu allows the operator to select an existing role and modify the original values. By default the iPlanet Trustbase Transaction Manager has three pre-set roles that map to specific users.

**Figure 4-2** List of default Roles



- **Administrator** This role allows Administration access to all configuration screens.
- **NoRole** - This is a role that is not active and is currently for internal use only. It is a holder for assigning "items" no role.
- **Identrus** - This role allows access to Identrus Services. At present there is one main service IdentrusCSCService that performs certificate status checks and forms the basis of all Validation, verification, integrity and authentication.

---

**NOTE** The iPlanet Trustbase Transaction Manager contains a default Role 'NoRole'. This is used internally by the iPlanet Trustbase Transaction Manager Services and should not be modified or removed.

---

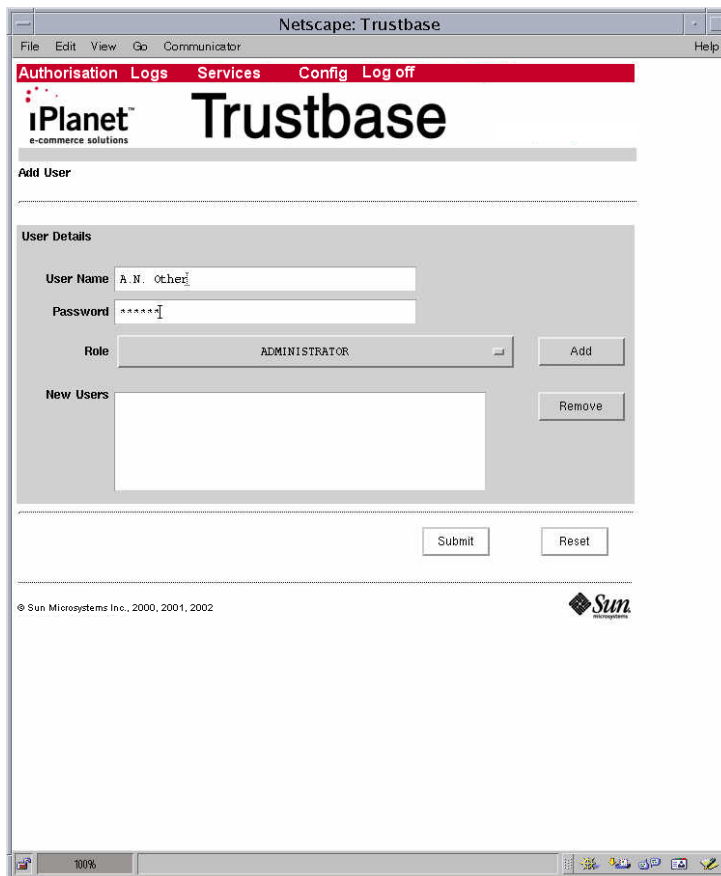
## Adding users to roles

Users may be identified in one of two ways:

- Username and Password
- Certificate

The username and password authorisation is generally used for operational management of the iPlanet Trustbase Transaction Manager. This allows operators to log onto the systems and interact with the management screens as shown in this manual.

**Figure 4-3** Add New User



New usernames can be added using the <New user> button on the Authorisation home page. The following input is required for each user:

- Username
- Password
- Role - This is only from the selection of existing roles

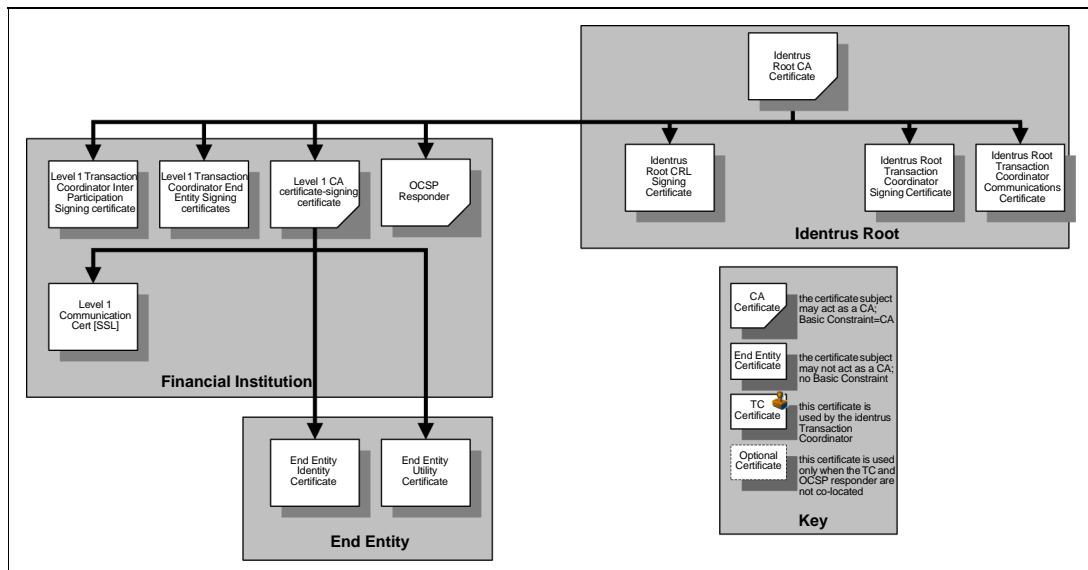


A number of users may be added prior to submitting the form. Once a set of users have been added they immediately become active in the Authorisation tables and are capable of using the role assigned to them.

## Adding a Certificate

Certificate authentication is used for Identrus messages, and before a third party may interact with the iPlanet Trustbase Transaction Manager they must have the certificate details entered in the Authorisation system. iPlanet Trustbase Transaction Manager removes the need to enter every known certificate in the Authorisation table by allowing the use of parent certificates in lieu of the actual certificate.

Figure 4-4 Identrus PKI hierarchy



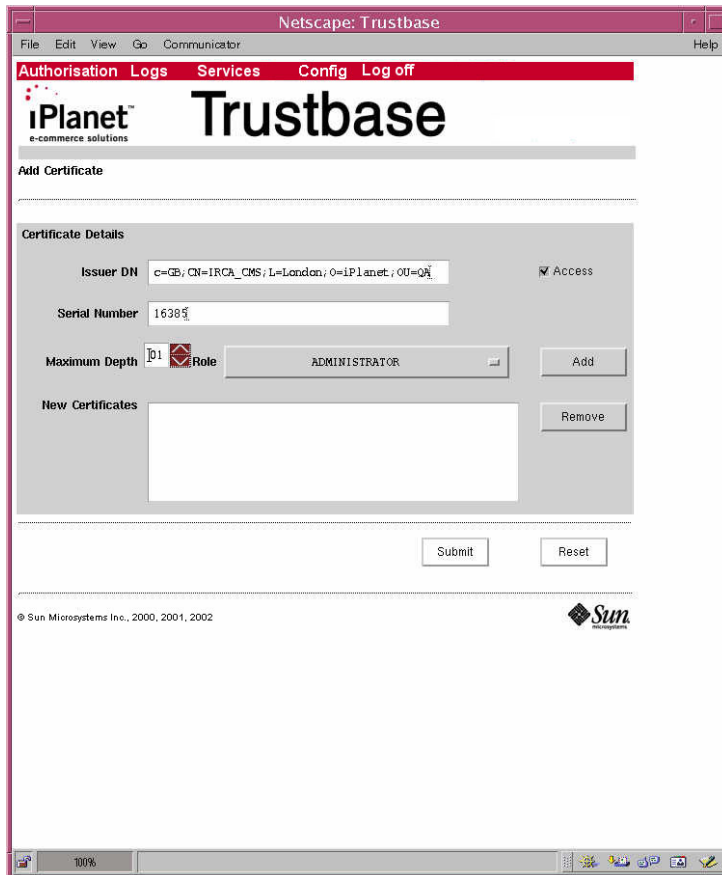
In the Identrus PKI hierarchy the End Entity Identity certificates issued by the Level 1 CA are used by the Relying Customer to sign requests to the iPlanet Trustbase Transaction Manager. The Level 1 Transaction Manager Inter-Participant Signing Certificate is used to sign requests made between the various iPlanet Trustbase Transaction Managers during a certificate status check.

The complete set of Identrus operations may be authorised by placing a single certificate in the authorisation system. This certificate is the Identrus Root CA Certificate. This Root CA certificate must be entered with a Maximum Depth value of 2. This indicates that certificates issued up to 2 levels below the Root CA certificate i.e. The Level 1 Transaction Manager End Entity Signing Certificate, and the Inter-Participant Signing Certificate(s) will be mapped onto the same Role as the Identrus Root CA certificate.

To add an authorisation based on a certificate select <Add Certificates> on the Authorisation main page. The form presented requires the following information for each certificate:

- Issuer DN - This field is case sensitive and DN information entered in the incorrect case will cause an authorisation failure for the certificate.
- Serial Number
- Maximum depth - Maximum length of the chain between this certificate and the child certificate capable of using this role.
- Role - Selected from the list of previously defined roles
- Access - Toggle on for this certificate to be active

**Figure 4-5** Adding a Certificate



The active toggle allows the operator to explicitly override an inherited authorisation. This means that a parent certificate may be used to authorise a large number of issued certificates except those that have an explicit entry in the Authorisation table with the Access toggle off. This mechanism is useful in situations where a certificate requires suspension for a period of time prior to the CA revoking it.

## Mapping roles to Services

Having created a role and mapped a set of users to the role the final step is to define the set of services that may be accessed using the role.

Each service is capable of being accessed by a single role. This requires some design and thought to the authorisation mappings prior to modifying an existing authorisation mechanism. By default the iPlanet Trustbase Transaction Manager contains an existing set of role to service mappings that allow the following:

- Operators to configure the installation using the configuration facilities described in this guide. This maps onto the "Administrator" role name.
- Holders of End Entity Certificates to access the approved set of Identrus service subject to mapping the certificate details onto the "Identrus" role.

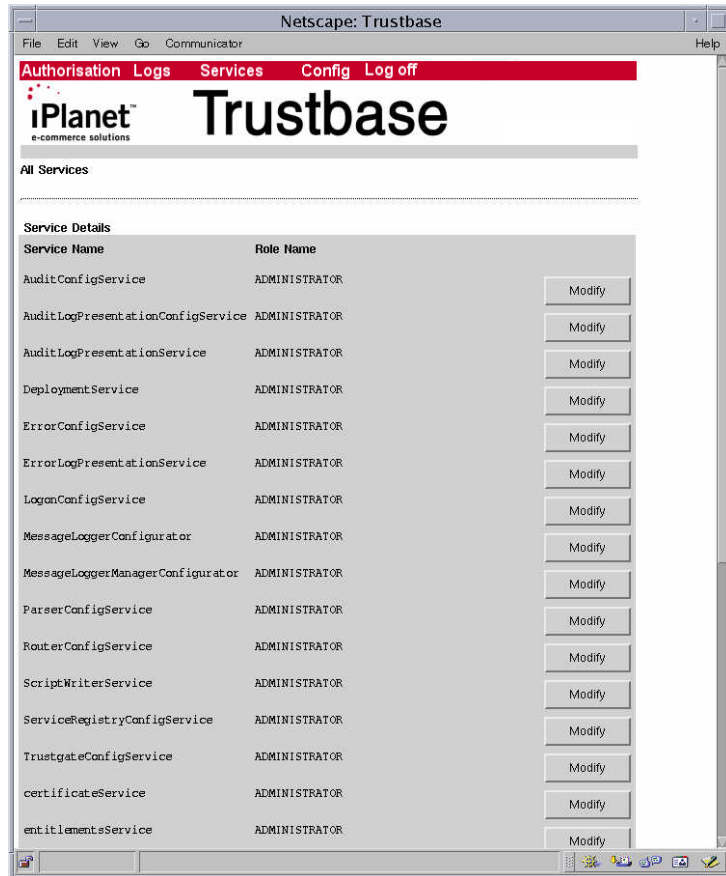
To map a new service to a role select new service on the Authorisation main menu. The form requires the following information:

- Service name - The short name of the service found in the tbase.properties file
- Role - The name of the role

A number of service to role mappings may be added to the list at the bottom of the form prior to submitting the form. Once the mappings have been submitted to the authorisation database they are effective immediately.

To modify an existing service to role mapping, select the edit services from the Authorisation main menu. This presents a list of the entire role to service mapping database for inspection. Selecting the Modify link on a particular item in the list will allow the details of the specific entry to be updated.

**Figure 4-6** Services to role mapping list



Authorising users to access a service

# Logs

Logs allow you to maintain control over messaging in terms of what errors are being generated together with an audit. These can be viewed. Options are available to configure the level of detail that can be seen on the screen.

There is also a raw log that provides detail of all message transactions that take place. These can be viewed using any standard Oracle tool.

# Introduction

iPlanet Trustbase Transaction Manager allows configuration of three kinds of Logs, two of which are directly viewable:

- The Audit log contains entries about the flow of a message as it passes through the iPlanet Trustbase Transaction Manager framework (e.g. message handler, Router). This log can be useful in diagnosing problems in configurations.
- The Error log contains entries of any runtime problems both from the iPlanet Trustbase Transaction Manager framework and the Identrus specific components.
- The Raw log is only used for Logging Identrus messages both received and sent. This log is not directly viewable.

**Figure 5-1** Logs Main Menu





# Audit log

Audits can be configured in terms of their types. They can also be queried. The following audit types are available:

## Trustbase Audits:

- **ROUTER\_ABORT\_ROUTING**  
This audit occurs when the rule based router aborts routing due to illegal rules.
- **ROUTER\_CONFIG**  
This audit occurs when a change is made to the rules via the rule configuration screens.
- **ROUTER\_CONSTRUCTION**  
This audit occurs when new rule sets are constructed at start up.
- **ROUTER\_CONTEXT\_DIRECTIVE**  
This audit occurs whenever the router executes one of the following router directives: EndContext, StartContext or ReturnToUser.
- **ROUTER\_ROUTE\_MESSAGE**  
This audit occurs whenever a message is routed to a service.
- **ROUTER\_START**  
This audit occurs whenever the rule based router component is initialized.
- **CONFIGURATION\_CHANGE**  
This audit occurs whenever a Trustbase configuration is changed.
- **OPERATION\_ABORT**  
This audit occurs whenever a service has to abort the processing of a message.
- **OPERATION\_BEGIN**  
This audit occurs whenever a service begins processing a message.
- **OPERATION\_COMPLETE**  
This audit occurs whenever a service successfully completes the processing of a message.
- **PARSER\_STARTUP**  
This audit occurs whenever the Message Analyzer component is started.
- **SECURITY\_CHANGE**  
This audit occurs whenever a generic security related event occurs.

- **TAS\_SHUTDOWN**  
This audit occurs when Trustbase is shutdown.
- **TAS\_STARTUP**  
This audit occurs when Trustbase is started.
- **ROLE\_SERVICE\_MAPPING\_CHANGED**  
This audit occurs whenever a mapping between a service and a role is changed or added in the entitlements configuration.
- **DEFAULT\_SECURITY\_ROLE\_USED**  
This audit occurs whenever the authentication component cannot find a specific mapping between a user and a role - it indicates that the default security role has been applied to that user.
- **CERT\_BASED\_ROLE\_MAPPING\_CHANGED**  
This audit occurs whenever a mapping between a certificate and a security role is made in the entitlements configuration.
- **USER\_PASS\_BASED\_ROLE\_MAPPING\_CHANGED**  
This audit occurs whenever a mapping between a username/password and a security role is made in the entitlements configuration.

## Identrus Transaction Co-ordinator Audits:

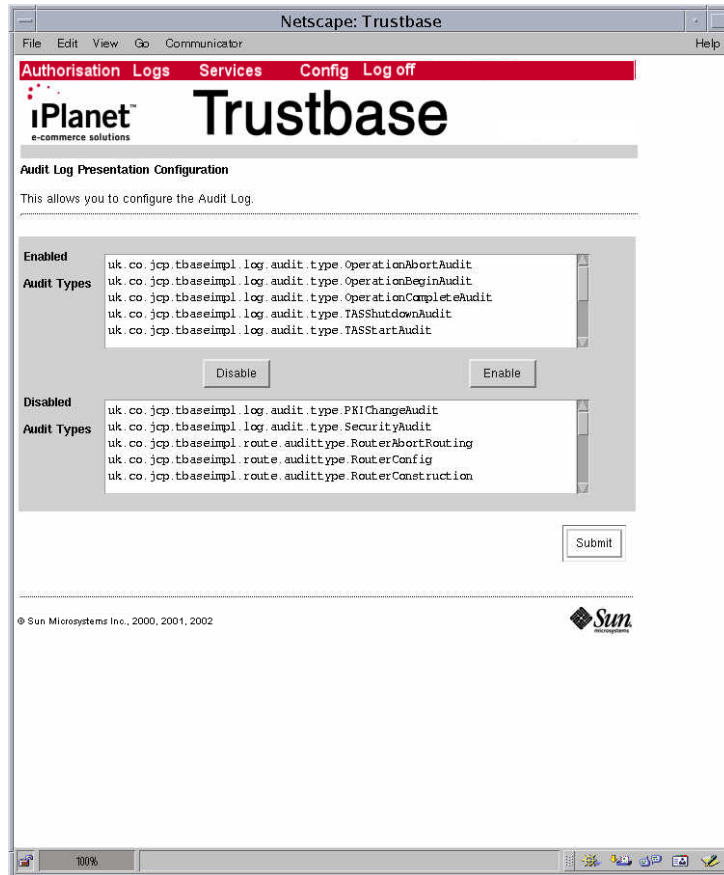
- **CSC\_PROCESSING**  
This audit occurs whenever a Certificate Status Check is being made.
- **CSC\_DEBUGGING**  
This audit occurs if you wish to debug a certificate Status Check.

## Audit Configuration

Audit Log Configuration allows you to select which audit types are physically viewable. Audit types are either enabled, i.e. they are logged and can be viewed, or disabled, i.e. no information is logged about these types.

In order to configure what gets logged: Select <Audit Configuration> from the main Log Menu.

**Figure 5-2** Configure Audit



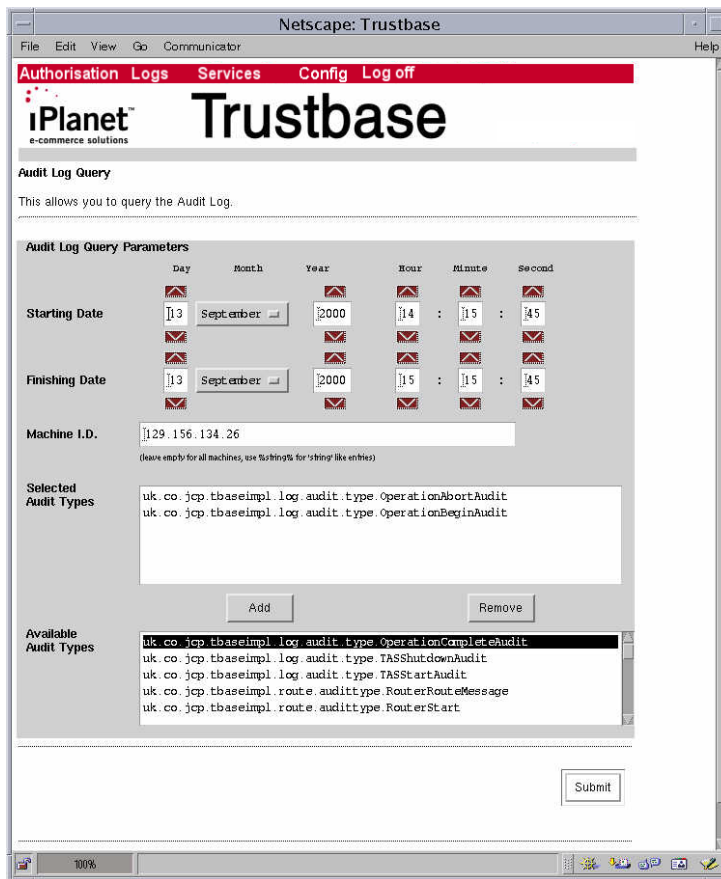
- Mouse <Left Click> on the audit type you wish to enable/disable.
- Select <enable> or <disable>

## Audit viewing

You can view the audit log by selecting a date range (Start and end date) and machine ID (IP Address). The machine I.D. refers in this case to the machine that is making the log. You can restrict or expand your view by removing or making available the appropriate audit types. Having made your selection the Date, Machine ID, Audit type and message content are displayed on the output screen.

In order to select what you want to view: Select <Audit Log Query> from Main Log menu

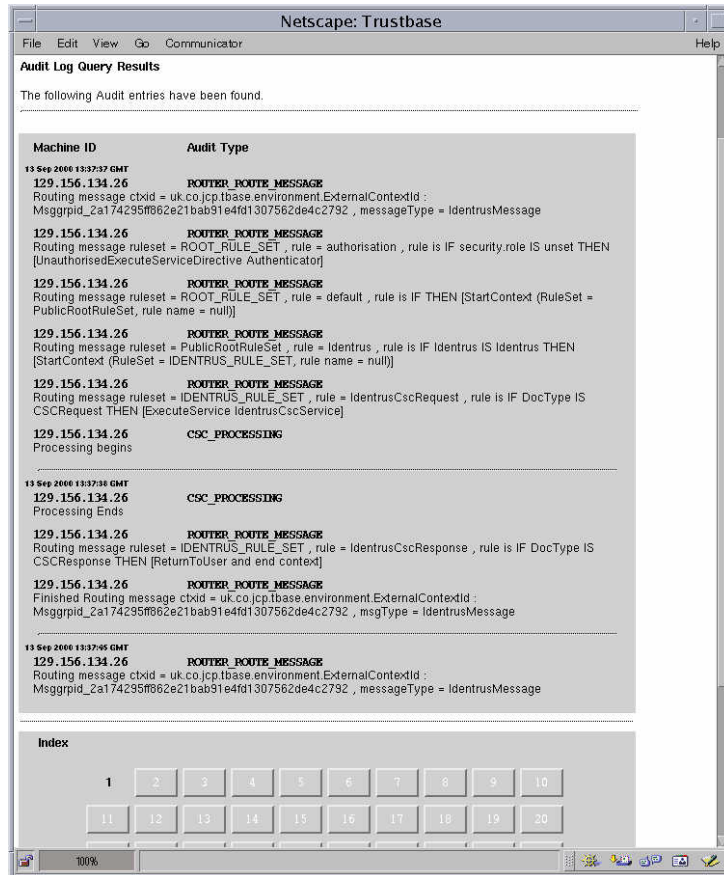
**Figure 5-3** Audit View



For more detailed log viewing, as all information is stored in a standard Oracle database, any third party database reporting tool may also be used.

The screen, as illustrated in Figure 5-4, might produce an output similar to the following audit:

**Figure 5-4** Audit Results



If results do not fit on one page there is an index tab, as illustrated at the bottom of the screen. Users intending to search using SQL should refer to the sql table AUDITDATA.

# Raw Logging

As part of being an Identrus member, you are required to maintain and archive a raw log. The goals fulfilled by logging the raw data are:

- Non Repudiation support - a complete transactional log that provides evidentiary support for transactions.
- Auditing - a complete transactional log that assists auditing the activities of iPlanet Trustbase Transaction Manager.

Normally these options, listed below, do not need to be changed. they can also be configured from `tbase.properties`:

- Signature Algorithm - By default the SHA-1/RSA algorithm is used to sign entries in the raw log. The options depend on the cryptographic security provider being utilised.
- Digest Algorithm - By default the SHA-1 algorithm is used to digest entries in the raw log. The digest is used as part of the raw log mechanism that ensures no tampering of the log contents.
- Certificate Attribute - This option is only used if the issuer DN and serial number fields are blank. By default, this field contains the value L1IPSC that indicates the certificate purpose ID, inter-participant signing certificate.
- Sequence Factory Type - This option should not be changed. It is for internal purpose only and affects the way data is sequenced for different database providers (e.g Oracle).
- Sequence Factory Name - This option should not be changed. It is for internal purpose only and affects the way data is sequenced for different database providers (e.g Oracle).

The message logger places the raw data it receives into the logs for safe-keeping. It will log data for Identrus specific transactions that it supports and for only those transactions. This raw data contains information in plain text and base64 encoding that gets signed by the message logger to provide the kinds of guarantees mentioned previously. At present there are facilities for multi-logging using the script.

```
/opt/ittm/Scripts/runAddLoggerWizard
```

---

**NOTE** The raw log can be displayed from Oracle using the `RAW_DATA` table e.g. "select \* from raw\_data order by timestamp desc;"

---

# Errors

Errors are now discussed in four sections:

- How to view errors
- What the severity of an error means
- Where to find a list of core iPlanet Trustbase Transaction Manager error messages
- A table summary of all Identrus specific error messages
- Checking errors during installation and operation

## Viewing

You can view the error log by selecting a date range (start and end dates) and machine ID (IP Address). You can restrict or expand your view by specifying a minimum and maximum error severity. Additionally, by specifying a Java class, errors can be viewed that are produced by that class only. Having made your selection the Date, Machine ID, class type and error message are displayed on the output screen. For example the following selection:

**Figure 5-5** Error Log Query

Netscape: Trustbase

File Edit View Go Communicator Help

Authorisation Logs Services Config Log off

iPlanet™  
e-commerce solutions

# Trustbase

**Error Log Query**

This allows you to query the Error Log.

**Error Log Query Parameters**

	Day	Month	Year	Hour	Minute	Second
Starting Date	25	September	2000	13	30	15
Finishing Date	25	September	2000	14	30	15
Machine I.D.	129.156.134.27 <small>(leave empty for all machines, use %sbing% for %string% like entries)</small>					
Class Type	<input type="text"/> <small>(leave empty for all class types, use %string% for %string% like entries)</small>					
Severity Range	Minimum		INFORMATION	Maximum		FATAL

Submit

© Sun Microsystems Inc., 2000, 2001, 2002

Sun  
microsystems

100%



The Error log can be displayed from Oracle using the ERRORVIEW view e.g. "select \* from ERRORVIEW"

The screen shown on the previous page might produce an output similar to the following errors:

**Figure 5-6** Error Log Query Results

The screenshot shows a Netscape browser window titled "Netscape: Trustbase". The browser's menu bar includes "File", "Edit", "View", "Go", "Communicator", and "Help". A red navigation bar at the top contains links for "Authorisation", "Logs", "Services", "Config", and "Log off". The main content area displays the "iPlanet™ Trustbase" logo and the heading "Error Log Query Results". Below this, a message states: "The following Error Log entries have been found." A table follows, listing two entries:

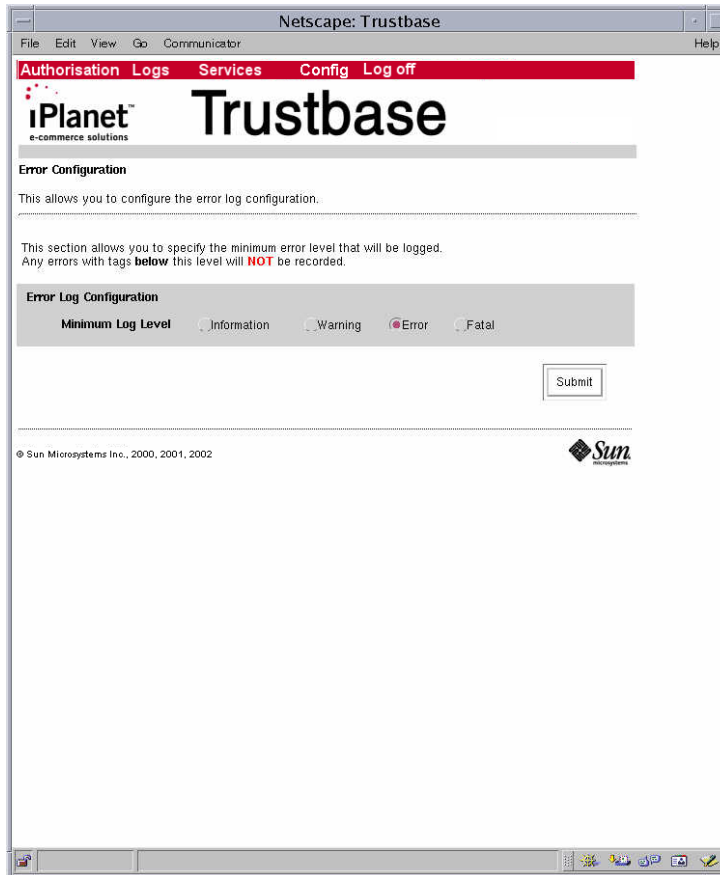
Machine ID	Severity
12 Sep 2000 14:14:03 GMT 129.156.134.26	ERROR
uk.co.jcp.tbbaseimpl.service.ConcreteServiceRegistry no config object available - proceeding with properties file config	
12 Sep 2000 17:50:40 GMT 129.156.134.26	WARNING
uk.co.jcp.tbbaseimpl.service.ConcreteServiceRegistry Unable to find service PingService	

Below the table is an "Index" section with a link to "1". At the bottom of the page, there is a copyright notice: "© Sun Microsystems Inc., 2000, 2001, 2002" and the Sun Microsystems logo. The browser's status bar at the bottom shows "100%" zoom and various system icons.

## Configuring Error Event Types

This section allows you to specify the minimum error level that will be logged. Any errors with tags below this level will **NOT** be recorded.

**Figure 5-7** Error Log Configuration



iPlanet Trustbase Transaction Manager defines an error as a severity, the class of object defining the error, and a programmer defined message. The default implementation defines four constants that indicate the various severity levels:

- **INFORMATION** - This constant is to be used to log informational events, which are not necessarily errors - this should be used sparingly.
- **WARNING** - This constant is to be used for error conditions that are expected and handled, but require logging for behaviour analysis.
- **ERROR** - This constant is to be used for serious errors which indicate that something is inherently incorrect with the system, but that allow processing to continue, or be retried.
- **FATAL** - This constant is to be used for fatal errors from which processing cannot recover, these errors would result in the abandoning of processing.

## Error Messages

Error messages fall into two categories, those that are produced by the iPlanet Trustbase Transaction Manager framework and those produced by Identrus services. For instance, Identrus message codes fall into a number of categories:

- Message Writer Errors
- Message Reader Errors
- Certificate Status Check Errors

Details of what all TTM core iPlanet Trustbase Transaction Manager error messages mean can be found in your Oracle Database in a table called `error_codes` as illustrated below:

**Figure 5-8** Selecting Error codes from your Oracle Database

```
su -oracle
sqlplus tbase/tbase
select * from error_codes;
```

## Error checking

### During Installation

If there was a problem during your installation you should check the following error logs

1. Your iTTM install logs

`/opt/ittm/Scripts/*.log`

2. Your iWS error logs

`/opt/iws6/https-myhost/logs/errors`

3. The Java iAS log file

`/opt/ias6/ias/logs/kjs_1_CCS0`

4. Your Oracle iTTM Error tables

`ERRORVIEW`

5. Other logs to look for

`/var/sadm/install/logs`

`/tmp/reg*.output`

### During Operation

1. In the first instance you should go to the Trustbase configuration screens and select from the main menu <Logs> <Error Log Query>
2. Your Oracle iTTM Error tables

`ERROR_SUPPORT`

## Typical SQL command to view logs

The following SQL statements should be used in conjunction with SQLplus to extract the relevant log information that will help support identify any problems.

```
set long 1000
set linesize 1000
```

```
select * from errorview
where errorid >= ( select MAX(ERRORid) -100 from
errorview)
order by timestamp desc;
```

```
select * from error_support
where errorid >= ( select MAX(ERRORid) -100 from
error_support)
order by errorid desc;
```

```
select * from raw_data
where recordmarker >= ( select MAX(recordmarker) -100 from
raw_data)
order by timestamp desc;
```



# SMTP Proxy Configuration

As part of the SMTP Proxy configuration various S/MIME Settings determine how iPlanet Trustbase Transaction Manager will accept mail based requests as well as the format of the responses. For example: Whether messages should be encrypted or not, or how responses should be signed.

# S/MIME Settings

The file `/opt/ittm/myhost/tbase.properties` contains a number of S/MIME settings that are now discussed:

```
[TbaseSmime]
mail.smtp.host=smtphost.smime.com
mail.from=ttm@smime.com
loopback=false
debug=false
connector.test=false
smime.capability.store.impl=com.iplanet.trustbase.security.smime.SimpleSmimeCapabilityStore
smime.mode=SIGN:ENVELOPE
smime.permit.unencrypted=true
smime.signing.cert=TTMEMAIL
smime.encryption.alg=3DES/CBC/PKCS5
```

- **SMTP server.** The hostname of your outgoing mail server.

```
mail.smtp.host=smtphost.smime.com
```

- **Default From address.** This should match the email address in the Distinguished Name (DN) of the default signing certificate.

```
mail.from=ttm@smime.com
```

- **Loopback test mode.** This setting is for diagnostic purposes and is not normally used.

```
loopback=false
```



- **Debug test mode.** This setting is for diagnostic purposes and is not normally used.

```
debug=false
```

- **Connector Test Mode.** This setting is for diagnostic purposes and is not normally used.

```
connector.test=false
```

- **This setting for internal use by iPlanet Trustbase Transaction Manager and should not normally be changed.**

```
smime.capability.store.impl=com.iplanet.trustbase.security.smime.SimpleSmimeCapabilityStore
```

- **The S/MIME mode parameter takes the form:**

```
MODE ::= [PROT] [:PROT] *
PROT ::= PROT_TYPE [, PROT_PROPERTY=VALUE]
PROT_TYPE ::= SIGN | CLEAR_SIGN | ENVELOPE
PROT_PROPERTY ::= smime.signing.cert | smime.encryption.alg
VALUE ::= string
```

- **S/MIME mode parameter.** This parameter is concerned with the outgoing response messages. If an email is signed using the SIGN parameter then if the signature does not verify, the message content cannot be read. However if the CLEAR\_SIGN parameter is used then even if the signature does not verify, the content can still be read. The ENVELOPE parameter indicates that the outgoing Trustbase response message will be encrypted
- **A simple S/MIME mode parameter specifying that a message should be signed and then enveloped.** Unless an application specifies the signing key, the key specified in the smime.signing.cert property will be used.

```
smime.mode=SIGN:ENVELOPE
```

- A more complete S/MIME mode parameter, specifying that messages should be signed with the key with an alias TTMEMAIL, and encrypted using DES

```
smime.mode=SIGN, smime.signing.cert=TTMEMAIL:ENVELOPE, smime.encryption.alg=DES
```

- Allow unencrypted requests. If true, and an ENVELOPE protection has been requested, but there is no key for the recipient, then the message will be sent unencrypted. If false, the message will not be sent.

```
smime.permit.unencrypted=true
```

- S/MIME default signing certificate alias. This alias should be assigned to the certificate that will sign and encrypt outgoing responses. The following TokenKeyTool (see iTTM Javadocs) commands will add the alias TTMEMAIL to your Identrus interparticipant signing certificate:

```
cd /opt/ittm/Scripts  
./runtokenkeytool  
addalias -alias IPSC -newalias TTMEMAIL
```

tbbase.properties can be amended as follows:

```
smime.signing.cert=TTMEMAIL
```

- The default encryption algorithm for outgoing S/MIME responses.

```
smime.encryption.alg=3DES/CBC/PKCS5
```

# Service Deployment

Services that involve message interaction between one machine and another can be deployed by configuring them in terms of their class files that determine message protocol, the java code that defines the service itself together with some rulesets that define various authorisation mechanisms and a "tbase.properties" file that allows the user to define configuration options for that particular service.

# Overview of Deploying a Service

Deploying a service can involve any number of configuration features depending on what kind of service you have to build. Sometimes it may involve defining your own configuration options and in such circumstances you may wish to configure a Template option. In other cases you may require authenticating that service in which case you will have to select your Authorisation options.

**Figure 7-1** Service Main Menu



There are essentially two main requirements for deploying a service:

- You deploy it by loading the services jar file into iPlanet Trustbase Transaction Manager.
- You then assign a role to authenticate the service, if necessary.

---

**NOTE** Developing a service prior to deploying it involves a number of steps.

1. Create your DTD definitions that specify the syntactic structure of the messages you wish to send round the system. These DTD's are thrown away once the class files are generated and messages are interpreted from the class files themselves. This has two main benefits: (a) its faster (b) its easier to develop.
2. Use Classgen `com.ipplanet.trustbase.app.classgen.ClassGen` to generate your java classes from your DTD definitions.
3. Write the Java code for the service deploying the Identrus API that assists the Identrus processing and validating of messages, certificates, keys and digital signatures.
4. Deploy the service within iPlanet Trustbase Transaction Manager by selecting the relevant configuration options as described below in this guide.
5. Finally, once it has been deployed within iPlanet Trustbase Transaction Manager itself you can run your service.

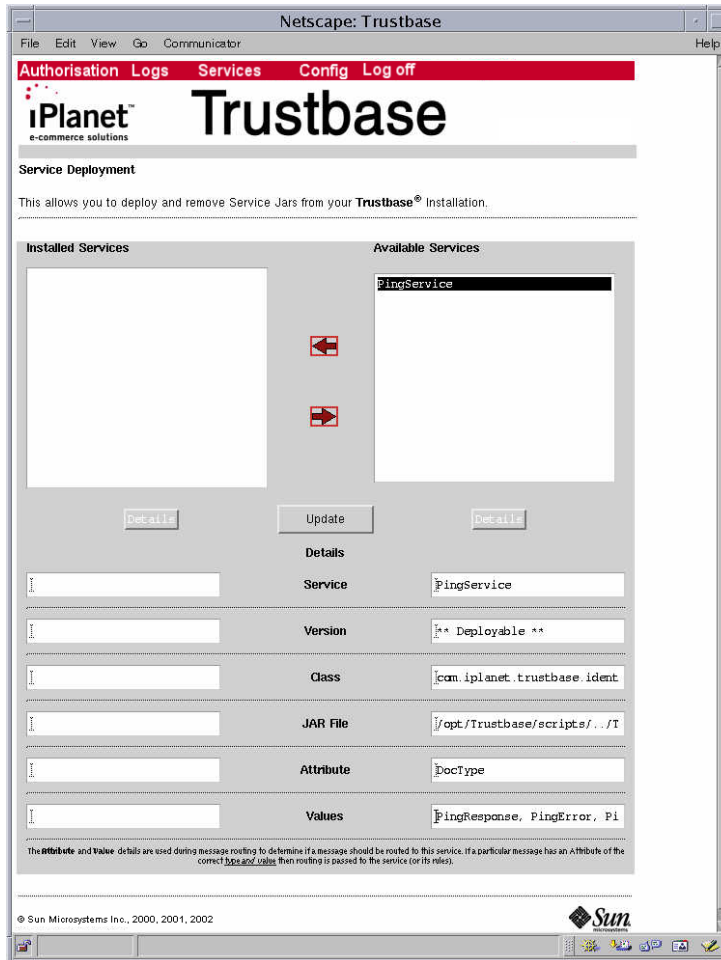
See the Developer Guide for more information on this.

---

# Deploying

- You deploy it by selecting <Services><Deployment>

Figure 7-2 Service Deployment



- The attribute and value details are used during message routing to determine if a message should be routed to this service. If a particular message has an attribute of the correct type and value then routing is passed to the service (or its corresponding rules).
- The Service is the service name
- The Version is the version of the service that is user defined
- The class is the main calling program
- The jar file that contains the service itself with all its associated data.

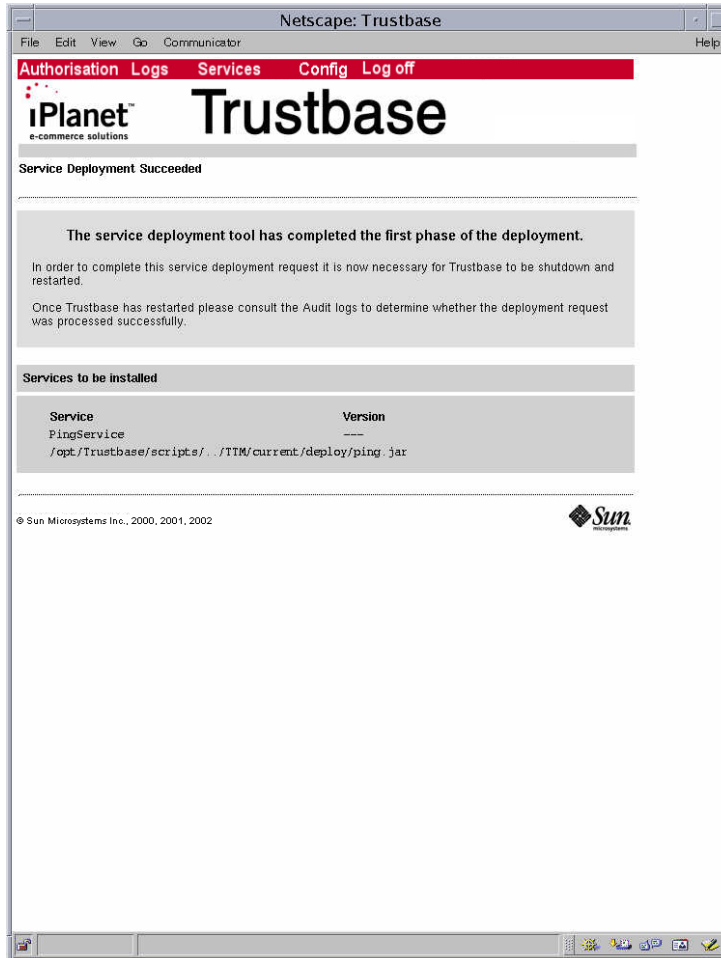
The service itself contains a jar file, placed in /opt/ittm/current/deploy that has a list of classes that contain the java code associated with the service, the java code that defines the service and a .properties file that defines how the service should be configured. This is illustrated below.

**Figure 7-3** Example jar file of service being deployed

Manifest.mf	MF File	11/09/2000 2...	68	0%	68	m...
version.txt	Text Document	11/09/2000 2...	248	54%	113	
ErrorInfo.class	CLASS File	11/09/2000 2...	7,477	54%	3,421	c...
VendorData.class	CLASS File	11/09/2000 2...	6,416	55%	2,912	c...
PingData.class	CLASS File	11/09/2000 2...	5,958	54%	2,765	c...
PingError.class	CLASS File	11/09/2000 2...	8,343	58%	3,527	c...
PingRequest.class	CLASS File	11/09/2000 2...	8,355	58%	3,533	c...
PingResponse.class	CLASS File	11/09/2000 2...	8,365	58%	3,533	c...
PingClient.class	CLASS File	11/09/2000 2...	10,555	55%	4,704	c...
PingService.class	CLASS File	11/09/2000 2...	1,808	59%	747	c...
config.mf	MF File	11/09/2000 2...	87	38%	54	c...
0.dtd	DTD File	11/09/2000 2...	806	60%	319	c...
1.dtd	DTD File	11/09/2000 2...	793	45%	440	c...
2.dtd	DTD File	11/09/2000 2...	667	59%	273	c...
3.dtd	DTD File	11/09/2000 2...	3,150	70%	956	c...
itbasesvc.proper...	PROPERTIES File	11/09/2000 2...	520	54%	238	c...

- Figure 7-4 below illustrates a sample output from deploying a service. The effects of deploying a service only occur when iPlanet Trustbase Transaction Manager has been restarted.

**Figure 7-4** Service Deployment Results



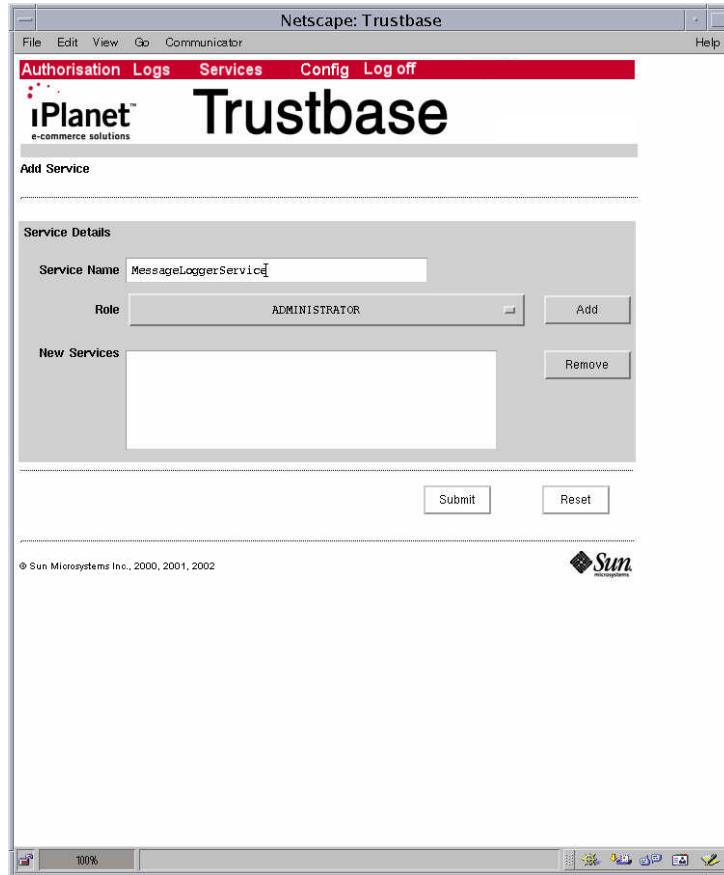


# Authorisation Services

If it requires Authorisation you'll need to:

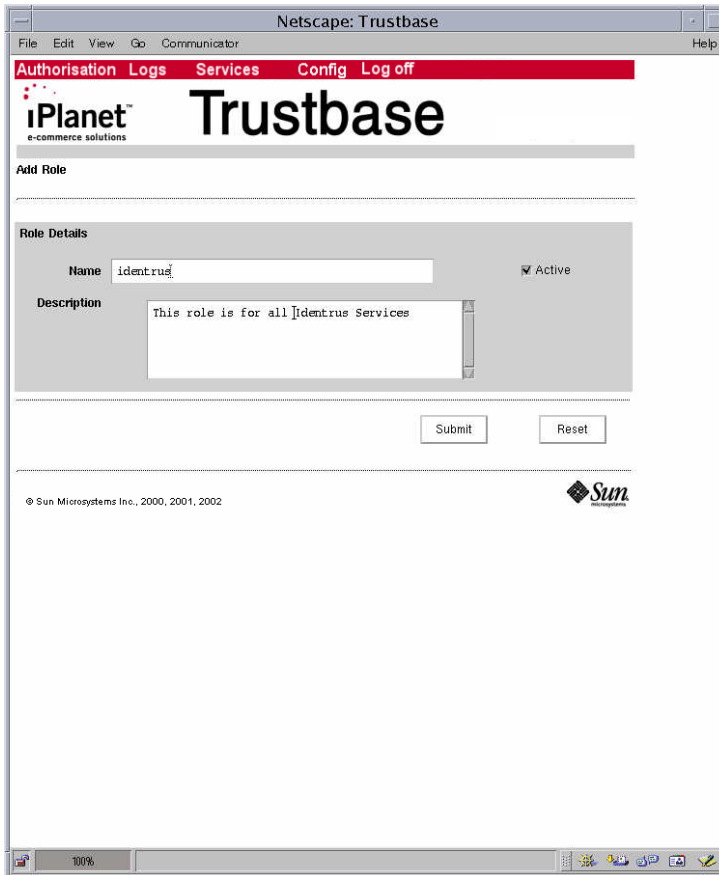
- Add the service by selecting <Add Service>

**Figure 7-5** Add Service



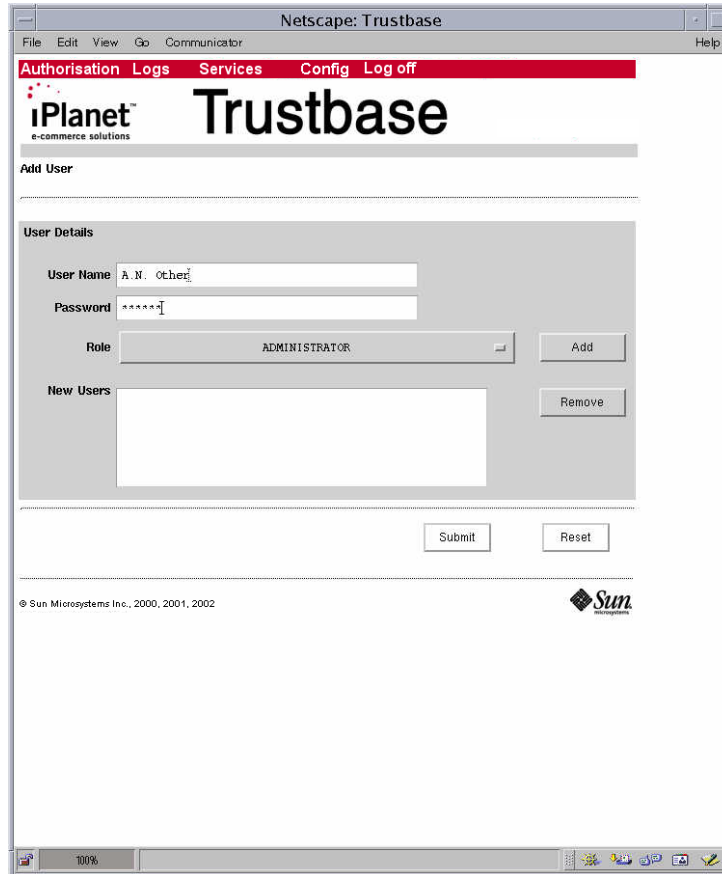
- Create a new role for this service by selecting <Add Role>

**Figure 7-6** Add Role



- Define some users for this role

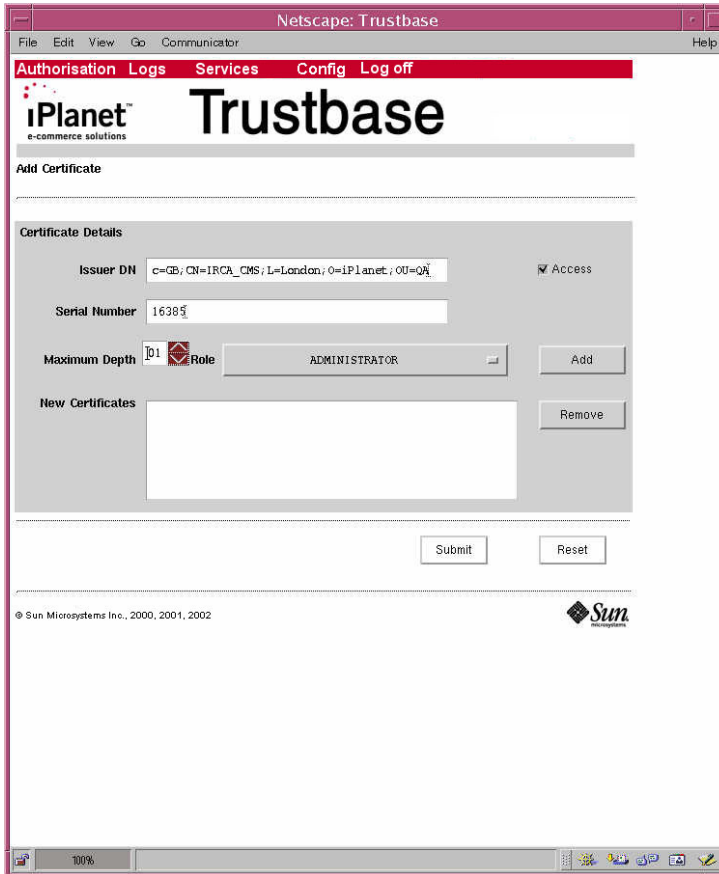
**Figure 7-7** Add User



- Make sure you select <Add> once you've allocated a role to a user
- Select <Submit>

- Add Certificate and associated Role

**Figure 7-8** Add Certificate



- Make sure you select <Add> once you've allocated a role to a user
- Select <Submit>

# Services not requiring Authorisation

If the service you wish to deploy does not require authorisation simply select the <services> <deployment> and don't follow the procedure to assign a role for the service.

# Registering your Service

You will need to edit

```
/opt/ittm/myhostname/tbase.properties
```

and add your service, using the following format:

```
service.description= <service_name>:<service_class>
```

The following is an example of how to register the ping service mentioned within the Developer Guide:

```
service.description=IdentrusPingService:com.iplanet.trustbase  
.identrus.ping.PingService
```

# Configuration Recovery

The objectives of this chapter are to cover:

- Dynamic Configuration Export and Import
- Configuration Security
- What data needs to be backed up
- What data is frequently written
- What configuration data needs to be backed up
- What happens when certificates expire?
- How to do Disaster Recovery?

# Dynamic Configuration Export

To export dynamic configuration items for backup use the Config | Export setting in the iTTM admin screens.



This will return a file for download. Some browsers may display this file rather than saving - in this case use View Source to generate a text document containing the exported configuration and save to the desired location.



# Configuration Security

Since the administration screens are accessible remotely, it is essential to protect the system from a malicious users attempting to change the iTTM config, rendering it open to attack. The administration screens are protected from unauthorised use but it might still be possible for a malicious user to trick an authorised administrator into importing a modified config file. To protect against this kind of attack, the configuration is exported with a signature. Only configuration signed by the designated key will be reimported (using Config | Import).

By default, the IPSC key will be used to sign configuration. If you wish to change the signing key or signing algorithm you will need the following Configuration Parameters found in `/opt/ittm/myhost/tbase.properties` for XMLConfig Import/Export

```
[DynamicConfigImportExport]
```

```
SigningAlgorithm - Default: SHA1withRSA
```

- algorithm with which to generate signature on exported config

```
SigningAlias - Default: CFGSC
```

- alias with which to mark token key store entry for signing exported config

To change the signing key, a token key store certificate / key entry should be generated and marked with the SigningAlias (the previous alias should be removed). This does not need to be issued by a Certificate Authority.

# Dynamic Configuration Import

This is done in the same way that you import any file with the restriction that you can only import what you have exported from that specific instance of iTTM

# Data Model

This section is intended to illustrate the Data Model so that the user can make their own decision about how they wish to proceed with archiving and backup.

**Figure 8-1** Oracle Data tables

ENVIRONMENTS	
E	VARCHAR2
ENVIRONMENT	BLOB
ENV_LOCK	NUMBER
LAST_ALTERED	NUMBER

INIT_TABLE	
SESSIONID	NUMBER
TIMESTAMP	NUMBER
N_CONNECTIONS	NUMBER
SIGDATA	RAW
SERVERCERTISSUERDN	VARCHAR2
SERVERCERTSERIALNUMBER	VARCHAR2

CONFIG	
CONFIGUID	VARCHAR2
SERIALOBJLONG	RAW

CERT_DATA	
ISSUERDN	VARCHAR2
SERIALNUMBER	VARCHAR2
SUBJECTDN	VARCHAR2
CERTDATA	LONG

CATEGORY	
CATEGORY_DESC	VARCHAR2
CATEGORY_NO	NUMBER
CATEGORY_NO_FK	NUMBER
CATEGORY_PICTURE_URL	VARCHAR2

ARCHIVE_TABLE	
ARCHIVEID	VARCHAR2
ARCHIVED	NUMBER
TIDSTART	VARCHAR2
TIDEND	VARCHAR2
TIMESTAMP	NUMBER

RAW_DATA	
SESSIONID	NUMBER
LOGCONNECTIONID	NUMBER
RECORDID	NUMBER
RECORDMARKER	VARCHAR2
TIMESTAMP	NUMBER
RAWDATA	LONG
DIGESTOFRECORD	RAW
SIGNEDDIGESTOFCALCULATION	RAW

**Figure 8-2 Oracle OCSP tables**

OCSP_REQUESTS	
ID	VARCHAR2
REQUESTOR_NAME	VARCHAR2
SIGNED	NUMBER
VERSION	NUMBER
REQUEST	LONG RAW
TIMESTAMP	DATE

OCSP_DATA	
OCSPID	NUMBER
TYPE	VARCHAR2
MESSAGE	VARCHAR2
MACHINE	VARCHAR2
TIMESTAMP	NUMBER
DATA	LONG

OCSP_RESPONSES	
ID	VARCHAR2
STATUS	NUMBER
RESPONSE	LONG RAW
TIMESTAMP	DATE

**Figure 8-3 Comms**

SMIME_TRANSPORT	
CONNECTION_ID	VARCHAR2
PEER_ISSUER_DN	VARCHAR2
PEER_CERT_SERIAL_NUMBER	VARCHAR2
MESSAGE_PROTECTION	VARCHAR2
TIME_STAMP_TYPE	VARCHAR2
TIME_STAMP	DATE

SMTP_MESSAGE	
STREAM_ID	VARCHAR2
CONNECTION_ID	VARCHAR2
RECIPIENTS	VARCHAR2
SENDER	VARCHAR2
TIMESTAMP	VARCHAR2
MESSAGE_VALID	NUMBER
MESSAGE_INVALID_REASON	VARCHAR2
TIMESTAMP	DATE

SMTP_CONNECTION	
STREAM_ID	VARCHAR2
PEER_IP_ADDR	VARCHAR2
TIMESTAMP	VARCHAR2
TIMESTAMP	DATE

SSL_CONNECTION	
CONNECTIONID	VARCHAR2
CLIENTCERTISSUERDN	VARCHAR2
CLIENTCERTSERIALNUMBER	VARCHAR2
CIPHERSUITE	VARCHAR2
CONNECTTIME	DATE
TIMESTAMP	VARCHAR2
CONNECTIPADDR	VARCHAR2
CONNECTIONFAILED	NUMBER
CONNECTIONFAILEDREASON	VARCHAR2

Figure 8-4 Audits

AUDITDATA	
AUDITID	NUMBER
MESSAGEID	VARCHAR2
MACHINEID	VARCHAR2
TIMESTAMP	DATE
AUDITTYPE	VARCHAR2
MESSAGE	VARCHAR2

AUDIT_SERVICES	
AUDIT_TYPE	VARCHAR2
AUDIT_TEXT_ID	VARCHAR2
ACTIVE	NUMBER

AUDIT_PARAMETERS	
AUDITID	NUMBER
PARAMETER	VARCHAR2
PARAMETER_NO	NUMBER

AUDIT_TEXT	
AUDIT_TEXT_ID	VARCHAR2
TEXT	VARCHAR2
LOCALE	VARCHAR2

AUDIT_LOCALE_TEXT	
AUDIT_TEXT_ID	VARCHAR2
TEXT	VARCHAR2
LOCALE	VARCHAR2
IS_DEFAULT	NUMBER

Figure 8-5 Users

REGISTERED_USER1_0	
USER_REF	VARCHAR2
TITLE	VARCHAR2
FIRST_NAME	VARCHAR2
SURNAME	VARCHAR2
ADDRESS	VARCHAR2
POST_CODE	VARCHAR2
COUNTRY	VARCHAR2
TELEPHONE	VARCHAR2
EMAIL	VARCHAR2
CERT_SERIAL_NUMBER	NUMBER
CERT_ISSUER_NAME	VARCHAR2

REGISTERED_USER	
USER_REF	VARCHAR2
TITLE	VARCHAR2
FIRST_NAME	VARCHAR2
SURNAME	VARCHAR2
ADDRESS	VARCHAR2
POST_CODE	VARCHAR2
COUNTRY	VARCHAR2
TELEPHONE	VARCHAR2
EMAIL	VARCHAR2
CERT_SERIAL_NUMBER	VARCHAR2
CERT_ISSUER_NAME	VARCHAR2

Figure 8-6 Roles

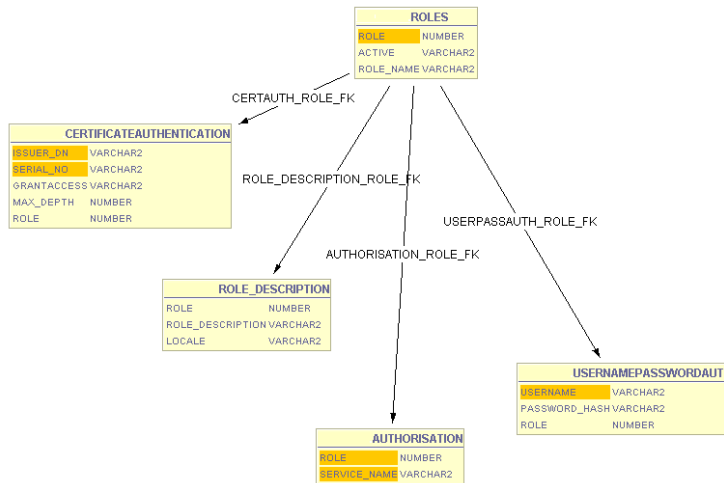


Figure 8-7 Identrus

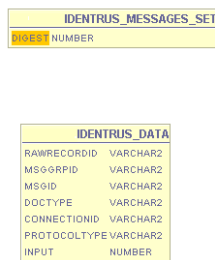
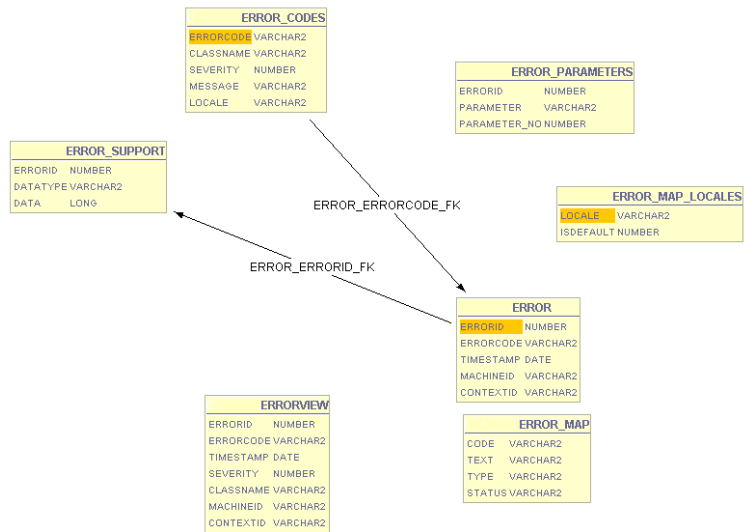


Figure 8-8 Errors



## Database Table Definitions

This section classifies all of the tables in an iTTM/iTPS/BIAB/Tooled Up installation.

Table Name	Description
archive_table	No Longer Used
attribute_key_attrs	No Longer Used
attribute_name_attrs	No Longer Used
auditdata	Log of all audit events
audit_parameters	Log of all event specific parameters for each audit event
audit_services	Private Internal Data
audit_text	Mapping of audit strings to locale specific strings
authorisation	Private Internal Data
bill_data	An entry for each Identrus Message is made here, it maps the message signer information to a raw log entry.
certificateauthentication	Private Internal Data
cert_data	All unique certificates from Identrus CertBundles are logged here.
cert_table	No Longer Used
config	Private Internal Data
default_locale	Private Internal Data
environments	Private Internal Data
error	Log of all errors
error_codes	Mapping of error codes to locale specific strings and error severity
error_map_locales	Private Internal Data
error_parameters	Log of all error specific parameters for each error entry
error_support	Log of supporting error data for error entries – java stacktraces
identrus_data	For each Identrus message in the raw log table an entry exists here with key Identrus message fields extracted into the table entry



init_table	Private Internal Data
key_table	No Longer Used
ocsp_data	Every OCSP request/response that goes to the local OCSP responder – OR request/response from a remote OCSP responder in fallback mode.
ocsp_requests	Used by the OCSP responder in iTTM to log all requests received.
ocsp_responses	Used by the OCSP responder in the iTTM to log all responses sent out.
raw_data	All messages arriving at iTTM are logged in this tamper evident table
revocation_attrs	No Longer Used
revocation_serial_num	No Longer Used
revocation_table	No Longer Used
roles	Private Internal Data
role_description	Private Internal Data
salt_table	No Longer Used
smime_transport	All smime messages have their security information logged
smtp_connection	All connections to the SMTP proxy are logged
smtp_message	All smtp messages received by the SMTP proxy are logged
ssl_connection	No Longer Used
ttm_organisation	Private Internal Data
ttm_role	Private Internal Data
ttm_user	Private Internal Data
ttm_user_role	Private Internal Data
usernamepasswordauth	Private Internal Data

- All tables marked as 'Private Internal Data' are controlled by the specified product and the structures of these tables are subject to change in future revisions.
- All tables marked as 'No Longer Used' are tables that existed in an earlier version of the product that are deprecated in the current version. Because of upgrade paths these tables are not dropped by their respective products, the DBA is at liberty to drop these tables once the contained data is no longer needed.

## Auditdata

Contains internal audit information & indicates what the TC processed.

**Figure 8-9** TTM table Auditdata

Column Name	Type	Size	NULL	Key Information	Description
auditid	NUMBER			Primary Key	Unique Identifier. Monotonic sequence number.
messageid	VARCHAR2	240			Not Used
machineid	VARCHAR2	50	Not Null		IP address of machine that logged audit event
timestamp	DATE		Not Null		The timestamp of when event was logged.
audittype	VARCHAR2	200	Not Null	FK-> audit_text.audit_t ext_id	The class of audit that this event is
message	VARCHAR2	1000	Not Null	FK - audit_text.audit_t ext_id	The audit message reference

## Auditparameters

Log of all event specific parameters for each audit event.

**Figure 8-10** iTTM table Auditparameters

Column Name	Type	Size	NULL	Key Information	Description
auditid	NUMBER			FK	The audit id in auditdata that this parameter relates to.
parameter	VARCHAR2	2000			The value of the parameter
parameter_no	NUMBER				The “tag” number when this parameter value is inserted into the audit_text

## audit\_text

Maps audit strings to locale specific strings

**Figure 8-11** iTTM table audit\_text

Column Name	Type	Size	NULL	Key Information	Description
audit_text_id	VARCHAR2	1000	Not Null	PK	Unique Id
text	VARCHAR2	1000			The text of the audit message or type in a locale specific form
locale	VARCHAR2	10	Not Null		The locale of the text

## bill\_data

Billing records are a sub-set of the information within the raw message log that provides sufficient information to determine who made each transaction. These tables are designed for used by third party tools that generate the actual Bill for the customer. The definitions for the bill table columns are as follows:

**Figure 8-12** iTTM table bill\_data

Column Name	Type	Size	NULL	Key Information	Description
subjectdn	VARCHAR2	500	Not Null		This will be the originator distinguished name extracted from the mandatory Identrus level 1 message signature. This will determine who should be billed.
issuerdn	VARCHAR2	500	Not Null		This will be the issuer distinguished name extracted from the mandatory Identrus level 1 message signature. This is to enable the identification of the exact key used to sign this message - in conjunction with the serial number field below.
serialnumber	VARCHAR2	100	Not Null		This will be the originator certificate serial number that may be used to identify the exact key used to sign the message - in conjunction with the issuer distinguished name.
rawrecordid	VARCHAR2	240	Not Null		This will be the RawRecordId of the associated raw log table record.

## cert\_data

In order to reduce the volume of data logged with each Identrus message the certificates contained with the message header are stripped out and stored in a certificate table. If the iPlanet Trustbase Transaction Manager has already logged a particular certificate in the table it will not be logged again. The information stored within the table is:

**Figure 8-13** iTTM table cert\_data

Column Name	Type	Size	NULL	Key Information	Description
issuerdn	VARCHAR2	500	Not Null	PK	The issuer distinguished name of the certificate, RFC 2253 format string.
serialnumber	VARCHAR2	100	Not Null	PK	The serial number of the certificate
subjectdn	VARCHAR2	500	Not Null		The subject distinguished name from the certificate, in RFC2253 format
certdata	LONG		Not Null		The Base64 certificate data.

## Error

The actual error log table is described below, this table is not normally viewed by the administrator directly, instead there is an Oracle view called errorview that provides a resolved view of the errors that have been logged.

**Figure 8-14** iTTM table error

Column Name	Type	Size	NULL	Key Information	Description
errorid	NUMBER		Not Null	PK	This is a unique id for the error log entry; it is generated from a monotonic sequence that means that this field may be used to accurately order error messages in the order that they were logged
errorcode	VARCHAR2	7		FK – error_codes.errorcode	This is the errorcode of the error being logged, see the error_codes table description.
timestamp	DATE				This is an ORACLE DateTime field that identifies when the error was logged.
machineid	VARCHAR2	50			This is a string representing the IP address of the iPlanet Trustbase Transaction Manager that logged the error - this may be different in a multi-node IAS installation.
contextid	VARCHAR2	200			This context id field is for future expansion.

## error\_codes

The iPlanet Trustbase Transaction Manager error logging mechanism requires that every different occurrence of an error be given a code which is unique throughout iPlanet Trustbase Transaction Manager.

**Figure 8-15** iTTM table error\_codes

Column Name	Type	Size	NULL	Key Information	Description
errorcode	VARCHAR2	7	Not Null	PK	This is the unique errorcode string that identifies the error; this must be 7 characters exactly. The normal for an error code is XXXnnnn. Where XXX is a three-letter code for the service or subsystem and nnnn is a unique number. e.g. IPH0009 is an error in the Identrus Protocol Handler.
classname	VARCHAR2	500			The class from which this error is logged. This places a constraint that each error code may only be used from one place.
severity	NUMBER				This is the severity level of the error, described previously. Constants for each of the error severities can be located in the uk.co.jcp.tbbaseimpl.log.error.Error Log class.
message	VARCHAR2	2000	Not Null		This is the localised version of the error message that will appear in the error log. Parameters may be used in this message as described by the standard Java class java.text.MessageFormat. The values to be placed in these parameters are passed in an array of strings that one of the ErrorObject constructors allows.
locale	VARCHAR2	10	Not Null		

## error\_parameters

This is a cross referencing table used for querying errors. parameters are used to expand text according to the error text.

**Figure 8-16** iTTM table error\_parameters

Column Name	Type	Size	NULL	Key Information	Description
errorid	NUMBER			FK – error.errorid	Error Number
parameter	VARCHAR2	200			Parameter description
parameter_no	NUMBER				Parameter number



## error\_support

When an error is logged it is often accompanied by some free form string data which helps to store the context in which the error occurred to aid diagnosis. The most common example of such data is exception stack traces.

**Figure 8-17** iTTM table error\_support

Column Name	Type	Size	NULL	Key Information	Description
errorid	NUMBER			FK – error.errorid	This links this entry to an entry in the Error table
datatype	VARCHAR2	200	Not Null		This datatype is an arbitrary string identifier that categorises the data in the data field. The only value for this field defined by iPlanet Trustbase Transaction Manager is "STACKTRACE" which identifies the contents of the data field to be a Java Exception Stack Trace.
data	LONG				Free form string data

## identrus\_data

The Identrus data table records identrus specific message data, which can be related to the raw log records in the raw\_data table, using the rawrecordid foreign key.

**Figure 8-18** iTTM table identrus\_data

Column Name	Type	Size	NULL	Key Information	Description
rawrecordid	VARCHAR2	240	Not Null	FW – rawdata.recordermarker	the id of the associated raw log record
msggrpId	VARCHAR2	120	Not Null		the Identrus MsgGrpId from the NIB of the message
msgId	VARCHAR2	120	NotNull		
doctype	VARCHAR2	120	Not Null		the DOCTYPE of the message.e.g.CSCRequest, PingRequest etc..
connectionid	VARCHAR2	100	NotNull		No Longer Used
protocoltype	VARCHAR2	10	Not Null		The protocol over which the message arrived e.g. HTTP or SMTP
input	NUMBER		Not Null		1 indicates message was inbound to the TC.  0 indicates that the message was outbound from the TC.

## ocsp\_data

This data records all the ocsp transactions -- responses and requests that are carried out between the local ocsp responder and iTTM.

**Figure 8-19** iTTM table ocsp\_data

Column Name	Type	Size	NULL	Key Information	Description
ocspid	NUMBER		Not Null	PK	A unique identifier for the record
type	VARCHAR2	2000	Not Null		OCSPREQUEST or OCSRESPONSE
message	VARCHAR2	2000	Not Null		A text summary of the contents of the request or response
machine	VARCHAR2	2000	Not Null		The URL to which the request was submitted to or the response was received from
timestamp	NUMBER		Not Null		The date and time that the entry was made
data	LONG		Not Null		Base64 encoding of the request or response

## ocsp\_requests

Records messages sent from iTTM to the OCSP Responder.

**Figure 8-20** iTTM table ocsp\_requests

Column Name	Type	Size	NULL	Key Information	Description
id	VARCHAR2	127	Not NULL	PK	Unique Identifier
requestor_name	VARCHAR2	500	Not Null		The DN of the person requesting the OCSP
signed	NUMBER				Whetehr it was signed (0 or 1)
version	NUMBER				Version number of OCSP
request	LONG RAW				DER of the request
timestamp	DATE		Not Null		Time of the request

## ocsp\_responses

Records messages received from the OCSP responder to iTTM

**Figure 8-21** iTTM table ocsp\_responses

Column Name	Type	Size	NULL	Key Information	Description
id	VARCHAR2	127	Not Null	PK	Unique Identifier connected between request and response
status	NUMBER				Overall status of the response (See RFC2560)
response	LONG RAW				DER of the response
timestamp	DATE		Not Null		Time of the response

## raw\_data

The raw log inserts a row into a relational database table for each log operation. The structure of the database table is described here. All raw log tables have the same structure, although each raw log uses a different table, whose name is determined when the raw log is created with the AddLoggerWizard. The raw logging facility records raw incoming and outbound message data.

**Figure 8-22** iTTM table raw\_data

Column Name	Type	Size	NULL	Key info	Description
sessionid	NUMBER		Not Null		The id of the raw log session that wrote record
logconnectionid	NUMBER		Not Null		The id of the connection within the session
recordid	NUMBER		Not Null		The id of the record within the connection
msggrpId	VARCHAR2	120			Message Group Id
msgid	VARCHAR2	120			Message Id
doctype	VARCHAR2	120			doctype of message
recordmarker	VARCHAR2	240	Not Null	PK	A unique monotonically increasing identifier
connectionid	VARCHAR2	100			
protocoltype	VARCHAR2	10			
input	NUMBER				
timestamp	NUMBER		Not Null		An integer which represents the UNIX time at which the record was logged.
rawdata	LONG		Not Null		The Identrus Message XML, without the CertBundle fields. The certificates from the bundle are logged separately in the "cert_data_table"
digestofrecord	RAW	2000			A SHA-1 digest of this record.
signeddigestofcalculation	RAW	2000			An RSA signature of this record and data from the previous record.

## smime\_transport

Logs incoming SMIME connections

**Figure 8-23** iTTM table smime\_transport

Column Name	Type	Size	NULL	Key Information	Description
connection_id	VARCHAR2	100	Not Null		Provides a link back to the smtp_message table
peer_issuer_dn	VARCHAR2	2000			The issuer_dn of the certificate that was used to verify the message
peer_cert_serial_number	VARCHAR2	100			The serial number of the certificate used to verify the message.
message_protection	VARCHAR2	100			The type of protection used to secure the message
time_stamp_type	VARCHAR2	10			The type of timestamp LOCAL or NETWORK
time_stamp	DATE		Not Null		The time at which the entry was made

## smtp\_connection

The ssl\_connection and smtp\_message tables both have connection\_id fields that are passed to the iPlanet Trustbase Transaction Manager running in the application server. This connection\_id is stored within the Identrus Log table allowing queries that link the originator information with the actual requests made.

**Figure 8-24** iTTM table smtp\_connection

Column Name	Type	Size	NULL	Key Information	Description
stream_id	VARCHAR2	100	Not Null		Identifier for the connection
peer_ip_addr	VARCHAR2	20	Not Null		IP address of the peer
timestamptype	VARCHAR2	10			Whether the time of the connection was local or remote
timestamp	DATE		Not Null		The time the connection was made



## smtp\_message

Logs incoming SMTP mail messages.

**Figure 8-25** iTTM table smtp\_message

Column Name	Type	Size	NULL	Key Information	Description
stream_id	VARCHAR2	100	Not Null		A unique id for the smime_transport
connection_id	VARCHAR2	100	Not Null		A unique id for the smtp connection
recipients	VARCHAR2	1000	Not Null		The recipients of this message
sender	VARCHAR2	100	Not Null		The sender of this message
timestamptype	VARCHAR2	10			The type of timestamp LOCAL or NETWORK
message_valid	NUMBER		Not Null		Is the message valid? 1 indicates it is valid
message_invalid_reason	VARCHAR2	1000			The reason for the invalidity of the message
timestamp	DATE		Not Null		The date and time at which the entry was made

## What configuration data needs to be backed up?

Configuration information is split into two sections: static information (changing which requires a restart of Trustbase) and dynamic information (at present this consists of the Audit Logging and Error Logging settings).

Static configuration can easily be backed up by copying the

1. /opt/ittm/myhost/\*.properties files

2. /opt/ittm/store certificate database

to a storage directory. Typically this would involve

```
mkdir /opt/temp
```

```
cd /opt/ittm/store
```

```
cp -r * /opt/temp
```

3. As mentioned in an earlier section, to export dynamic configuration items for backup use the Config | Export

setting in the iTTM admin screens.



This will return a file for download. Some browsers may display this file rather than saving - in this case use View Source to generate a text document containing the exported configuration and save to the desired location.

4. For a list of sql tables see for instance

```
/opt/ittm/current/config/sql  
su oracle  
sqlplus tbase/tbase  
select TABLE_NAME from USER_TABLES;  
exit;
```

**5. We now list the important sql tables**

**a. Identrus Local OCSP Log**

OCSP\_DATA

**b. Identrus Message log**

IDENTRUS\_DATA

**c. Trustbase Raw log (linked to Identrus\_data)**

RAW\_DATA

INIT\_DATA

**d. Trustbase OCSP Responder (optional)**

OCSP\_REQUESTS

OCSP\_RESPONSES

**e. Trustbase Configuration Table**

CONFIG

**f. Trustbase Authorisation Table**

AUTHORISATION

ROLES

ROLE\_DESCRIPTION

**g. Trustbase Error codes**

ERROR

ERROR\_CODES

ERROR\_PARAMETERS

**h. Trustbase Certificate Store**

CERT\_DATA

**i. Trustbase Auditing Subsystem**

AUDITDATA

**j. Trustbase Auditing Subsystem**

AUDITPARAMETERS

AUDIT\_SERVICES

What configuration data needs to be backed up?

AUDIT\_TEXT

**Take a snapshot backup of the environment.**

```
su - oracle
```

```
exp
```

```
Export: Release 8.1.7.0.0 - Production on Wed Feb 27 10:30:38  
2002
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

**Log as a super user with DBA privilege and export the iTTM user. Refer to the Oracle documentation <http://technet.oracle.com> for the Export tool user guide.**

- 6. To backup the LDAP directory used by iAS you can replicate the server, see for instance**  
<http://docs.sun.com/source/816-5770-10/adconfig.htm#13057>
- 7. iWS copy the installation directory i.e. /opt/iws6**
- 8. The HSM security world i.e. /opt/nfast/kmdata**
- 9. iTTM software components are of course available on the CD-ROM supplied with this document.**

# What happens when certificates expire?

It is possible to have two sets of certificates running simultaneously. When certificates are nearing their expiry date the following procedure needs to be adopted.

There are two cases:

## 1. CA certificate expiry

- a. New CA cert issued to same key

Add the new CA certificate as a `TrustedCertificateEntry` using the `TokenKeyTool`

- b. New CA cert issued to new CA key

All certificates issued by the expired certificate in a Trust Domain must be reissued with the CA certificate, and imported to the `KeyEntry` using `TokenKeyTools importkeychain` command.

## 2. Subject Certificate expiry

- a. New Certificate to be issued to same key

Use `TokenKeyTool` to generate a Certificate request for the `KeyEntry` associated with the expired certificate and import the resulting certificate to the `KeyEntry`.

- b. New cert to be issued to new key

Generate new key (with same subject name as old key, if desired) using `TokenKeyTool`. Generate a Certificate request, and import the resulting certificate to the `KeyEntry`. More pertinent aliases from the old to the new `KeyEntry` using `TokenKeyTool`

## How to do Disaster Recovery?

In the event of hardware or disk failure it will be necessary to perform a disaster recovery. By ensuring the following contents are intact through restoration from backup, a iPlanet Trustbase Transaction Manager can continue its operation.

- nCipher Users only. nCipher "Security World" needs to be restored according to the KeySafe User Guide using the Administrator Card Set and the nCipher backup data.
- Reinstall iWS 6.0 SP2, iAS 6.0 SP3, database and iTTM 3.0.1
- Reinstall /opt/ittm/myhost
- Import saved Configuration via



- Reinstall database from the backup of tables created under the user specified in the SQL script in your Installation Guide. If necessary consult your Database Administrator.

---

**NOTE** Refer to the installation worksheet for information about the setup of iPlanet Trustbase Transaction Manager's application server and database.

---

# Glossary and References

The objectives of this chapter are to cover

- Software Platform
- Protocols
  - Transport Protocols
  - Security Related Protocols
  - Trading Protocols
  - Message Protocols
- Glossary

# Software Platform

## **Solaris 8**

<http://www.sun.com/software/solaris/cover/sol8.html>

## **JDK1.3.1**

<http://www.javasoft.com>

## **iPlanet Application Server 6.5**

[http://www.iplanet.com/products/infrastructure/app\\_servers/index.html](http://www.iplanet.com/products/infrastructure/app_servers/index.html)

## **iPlanet Web Server 6.0 SP2**

[http://www.iplanet.com/products/infrastructure/web\\_servers/index.html](http://www.iplanet.com/products/infrastructure/web_servers/index.html)

## **Oracle 8.1.7**

<http://www.oracle.com>

## **Hardware Security nCipher KeySafe 1.0 and CAFast**

<http://www.ncipher.com>



# Transport Protocols

## HTTP

HTTP/1.0 or 1.1 protocol:

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

<http://www.ietf.org/rfc/rfc1945.txt>

## SMTP RFC821

<ftp://ftp.isi.edu/in-notes/rfc821.txt> <http://www.imc.org/ietf-smtp/>

# Security Related Protocols

## **S/MIME Version 2 Message Specification**

<ftp://ftp.isi.edu/in-notes/rfc2311.txt>

<http://www.imc.org/ietf-smime>

<http://www.ietf.org/rfc/rfc2311.txt>

## **DOMHASH**

<http://www.ietf.org/rfc/rfc2803.txt>

## **OCSP**

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

## **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>

# Trading Protocols

## Identrus

<http://www.identrus.com>

Transaction Coordinator requirements (IT-TCFUNC)

Core messaging specification (IT-TCMPD)

Certificate Status Check Messaging specification (IT-TCCSC)

# Message Protocols

## DOM

<http://www.w3.org/TR/REC-DOM-Level-1/>

## DTD

<http://www.w3.org/XML/1998/06/xmlspec-v20.dtd>

## XML

<http://www.w3.org/TR/REC-xml>

## XML Syntax Processing specification

<http://www.w3.org/TR/xmlsig-core>

## HTML

### HTML 3.2 as specified in

<http://www.w3.org/TR/REC-html32.html>

# Glossary

<b>3DES</b>	Similar to DES.
<b>AIA</b>	Authority Information Access
<b>Application protocol</b>	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.
<b>ASN.1</b>	Abstract Syntax Notation One.
<b>Authentication</b>	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.
<b>base64</b>	A representation of characters in digital format using a 65 character subset of U.S. ASCII.
<b>BBS</b>	A random number generating algorithm.
<b>BER</b>	Basic encoding Rules used with X509.
<b>Block cipher</b>	A block cipher is an algorithm that operates on plaintext in groups of bits, called blocks. 64 bits is a typical block size.
<b>Bulk cipher</b>	A symmetric encryption algorithm used to encrypt large quantities of data.
<b>CA</b>	Certificate Authority
<b>Cipher Block Chaining Mode (CBC)</b>	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).
<b>Certificate</b>	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.
<b>Certificate Authority</b>	An organisation authorised to issue certificates (as in CA).
<b>Client</b>	The application entity that initiates a connection to a server.
<b>CN</b>	Common Name See for instance <a href="http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html">http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html</a> for definition or <a href="http://docs.sun.com/source/816-5613-10/contents.htm">http://docs.sun.com/source/816-5613-10/contents.htm</a>

<b>Connection</b>	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.
<b>CRL Certificate Revocation List</b>	A list of certificates that have been declared invalid by their issuing CA before their expiry dates
<b>CSC</b>	Certificate Status Check
<b>Data Encryption Standard (DES)</b>	DES is a very widely used symmetric encryption algorithm. DES is a block cipher.
<b>DER</b>	Distinguished Encoding rules used in X509.
<b>DH</b>	A public-key cryptographic algorithm for encrypting and decrypting data.
<b>Digital Signature Standard (DSS)</b>	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.
<b>Digital signatures</b>	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.
<b>DN</b>	Distinguished Name. See for instance <a href="http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html">http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html</a> or <a href="http://search.ietf.org/internet-drafts/draft-zeilenga-ldapv3-bis-rfc2253-00.txt">http://search.ietf.org/internet-drafts/draft-zeilenga-ldapv3-bis-rfc2253-00.txt</a> for definition. Also <a href="http://docs.sun.com/source/816-5613-10/contents.htm">http://docs.sun.com/source/816-5613-10/contents.htm</a>
<b>DSA</b>	Digital Signature Algorithm.
<b>EE</b>	End Entities are customers. i.e. the last person in the certificate chain.
<b>Handshake</b>	An initial negotiation between client and server that establishes the parameters of their transactions.
<b>HSM</b>	Hardware Security Module.
<b>HTML</b>	HyperText Markup Language.
<b>IDEA</b>	A 64-bit block cipher designed by Xuejia Lai and James Massey.
<b>IRCA</b>	Is the certificate for the Identrus root
<b>Integrity</b>	You know your email message has not changed.

<b>IP</b>	Issuing Participant Bank (or other financial institution) issuing smart cards containing private keys and certificates to Subscribing Customers.
<b>IR</b>	Identrus Root
<b>key</b>	The key used to encrypt data written by the client.
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>L1CA</b>	Is the purpose ID or attribute for CA certificates
<b>L1IPSC</b>	The purpose ID or attribute of Certificate used for interbank message signing
<b>L1EESL</b>	The purpose ID or attribute of Certificate used for bank/RC or bank/SC SSL connections - as server
<b>L1EESC</b>	The purpose ID or attribute of Certificate used for bank/RC or bank/SC message signing
<b>Message Authentication Code (MAC)</b>	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.
<b>MD5</b>	MD5 is a secure hashing function that converts an arbitrarily long data stream into a digest of fixed size.
<b>MIME</b>	MultiPURPOSE Internet Mail Extension
<b>Non-repudiation</b>	A process set up to ensure that the sender cannot disavow a message
<b>OCSP</b>	Online Certificate Status Protocol
<b>OU</b>	Organisation Unit See for instance <a href="http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html">http://www.itu.int/itudoc/itu-t/rec/x/x500up/x500.html</a> or <a href="http://search.ietf.org/internet-drafts/draft-zeilenga-ldapv3-bis-rfc2253-00.txt">http://search.ietf.org/internet-drafts/draft-zeilenga-ldapv3-bis-rfc2253-00.txt</a> for definition or <a href="http://docs.sun.com/source/816-5613-10/contents.htm">http://docs.sun.com/source/816-5613-10/contents.htm</a>
<b>PBE</b>	Password based encryption
<b>PEM</b>	Privacy enhanced mail
<b>Public Key Infrastructure (PKI)</b>	Defines protocols to support online interaction.
<b>Public key cryptography</b>	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.
<b>OSI</b>	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 portal 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
- UTF8** A multi-byte character encoding format. See <http://www.utf-8.org/>
- X509** An authentication framework based on ASN.1 BER and DER and base64.



# Index

## A

- Acceptable Machine Installations, 140
- Add Certificate, 170, 204
- Add New User, 168
- Add Role, 202
- Add Service, 201
- Add User, 203
- Adding a Certificate, 169, 171
- Adding users to roles, 160, 167
- Architectural Configuration, 135
- Audit Configuration, 178
- Audit log, 176, 177
- Audit Results, 181
- Audit View, 180
- Audit viewing, 180
- Authorisation, 163, 169, 170, 201, 205
- Authorisation Main Menu, 164
- Authorisation Services, 201
- Authorising users to access a service, 165

## B

- Backup, 207

## C

- Certificate, 15, 243
- Certificate Prerequisites, 30
- Certificate requests and responses, 242
- Chapter Single Template, 253
- Classes
  - Certificate, 15, 243
- Configuration Management overview, 158
- Configuration Options, 161
- Configure Audit, 179
- Configuring Error Event Types, 186
- Configuring the SMTP Proxy Separately, 145

## D

- Defining a role, 165
- Deploying, 157, 198
- DMZ Architecture, 139, 145
- DMZ Architecture for separating the SMTP Proxy, 145

## E

- Error Log Configuration, 186
- Error Log Query, 184
- Error Log Query Results, 185

- Error Messages, 187
- Errors, 183
- Example Identrus.properties, 99
- Example iPlanet Application Server Script, 40
- Example jar file of service being deployed, 199

## G

- Glossary, 245
- Glossary and References, 239

## H

- Hardware Security nCipher KeySafe 1.0 and CAFast, 15, 240
- How to do Disaster Recovery?, 238
- HTML, 244, 246
- HTTP, 38, 138, 241

## I

- Identrus, 15, 63, 78, 98, 100, 136, 169, 172, 182, 243, 247
- Identrus Configuration, 77, 98
- Identrus Message Specifications, 15
- Identrus PKI hierarchy, 169, 170
- Identrus Transaction Co-ordinator Audits, 178
- identrus.properties file location, 98
- Installation Structure, 71
- Installation Worksheet, 23
- Introduction, 13
- iPlanet Application Server 4.1, 15
- iPlanet Application Server configuration, 136
- iPlanet Application Server v6.0 Installation, 39
- iPlanet Certificate Management System, 15
- iPlanet Trustbase Transaction Manager Directory Overview, 71

- iPlanet Trustbase Transaction Manager Initialisation Files, 74
- iPlanet Trustbase Transaction Manager Installation Process, 44
- iPlanet Trustbase Transaction Manager Overview Directory, 72
- iPlanet Trustbase Transaction Manager Scripts, 73, 120
- iPlanet Web Server 6.0, 15, 240
- iPlanet Web Server Administration Server, 37
- iPlanet Web Server, Enterprise Edition 4.1, 36

## J

- Java, 29, 41

## L

- LDAP, 247
- List of default Roles, 166
- Logging on, 157
- Logon Screen, 159
- Logs, 161, 175
- Logs Main Menu, 176

## M

- Machine Installations, 140
- Mapping roles to Services, 172

## O

- OCSP, 62, 99, 100, 132, 242, 247
- OCSP Responders, 62, 63, 132
- OCSP Responders and Validating signed OCSP Responses, 132

Oracle 8i, 15, 29, 137, 240  
Oracle 8i Installation and Configuration Guides, 15  
Oracle Database Configuration, 77, 78, 140  
Overall Layout, 14  
Overview of Deploying a Service, 196

## P

Packages included, 29  
Packages not included, 30  
Post Installation Steps, 77  
Post iPlanet Web Server and iPlanet Application  
Server Installation Steps, 41  
Pre-requisites, 29  
Protocol, 247

## R

Raw Logging, 182  
Registered Services, 206  
Related Documents, 15  
Restart procedure, 121  
Running the iPlanet Trustbase Transaction Manager  
SQL Scripts, 79

## S

Selecting Error codes from your Oracle  
Database, 187  
Service Deployment, 192, 195  
Service Deployment Results, 200  
Service Main Menu, 196  
Services not requiring Authorisation, 205  
Services to role mapping list, 173  
Software Platform, 240  
Solaris 8 and Java Development Kit 1.2.1, 15  
Solaris Installation, 32

SSL, 138, 248  
Start iPlanet Trustbase Transaction Manager, 120  
SW pre-requisites, 28  
System Resources, 30

## T

Third Party Library jar files, 28  
Trustbase Audits, 177

## U

Using a DMZ, 138

## V

Verifying iPlanet Application Server, 41  
Verifying iPlanet Web Server 4.1, 36  
Viewing, 184

## W

What data needs backup?, 211  
What happens when certificates expire?, 237



# Using TokenKeyTool

# Starting TokenKeyTool

The TokenKeyTool manipulates TokenKeyStores and collections of TokenKeyStores. It is invoked as:

```
java com.iplanet.trustbase.security.store.TokenKeyTool [
<globalswitch>* ] [ <command> ]
```

Commands are strings of alphanumeric characters, while switches are '-' symbols followed directly by a string of alphanumeric characters, and possibly followed by a single argument [ arguments may be not present, alphanumeric, or numeric, depending on the switch ]

If a command is passed on the command line, then it is executed directly, with the switches passed on the command line, and the VM exits. If no command is passed on the command line, then a shell mode is entered, into which commands may be entered as:

```
<command> [ <localswitch>* ]
```

Each command in the shell mode is executed with a set of switches which is comprised of any local switches, plus any global switches which were passed in on the original command line [ or set with the setglobals command ]. If any switch is specified in both local and global switches, the local switch overrides the global switch. Thus, if the TokenKeyTool is invoked as below, then the global switches will specify a TokenKeyStoreManager and TokenKeyStore which will be used by default in all operations [ unless the global switches are overridden, or new defaults are subsequently set ]

```
java com.iplanet.trustbase.security.store.TokenKeyTool -domainspace
"file:~/domains" -manager m -store defaultStore
```

Commands are broadly divided into those that deal with TokenKeyStoreManagers and tokenKeyStores as a whole, those that deal with the contents of TokenKeyStores, utility commands that perform cryptographic operations with the objects in TokenKeyStores, and a few miscellaneous commands useful for examining the TokenKeyTool environment.

There is also a script for running TokenKeyTool

```
cd ../opt/ittm/Scripts
./runtokenkeytool
```

## Commands for dealing with TokenKeyStoreManagers

```
openstoremanager -domainspace <domainspace> -manager <localname>
-nodefault
```

opens a TokenKeyStoreManager, associating it with a nickname for convenient reference, and optionally making the opened TokenKeyStoreManager the default.

- **<domainspace>**: a URL describing a trust domain space. The URL will be used to create a TrustDomainManager. Its protocol determines the type of TrustDomainManager which will be created, and consequently where the TrustDomains are stored. If a file: URL is given then the URL must refer to a directory, within which individual TrustDomains will be stored as files named <TrustDomainName>.domain
- **<localname>**: a nickname by which the TrustDomainManager will be referred to in any subsequent operations in the TokenKeyTool
- **-nodefault**: if specified then the TokenKeyStoreManager instance will not be made the default TokenKeyStoreManager for subsequent operations

Some example openStoreManager commands

```
openstoremanager -domainspace "file:domains" -manager m
```

This one opens a TokenKeyStoreManager backing on to a FileTrustDomainManager which stores TrustDomain definitions in the directory "domains" relative to the current working directory. The TokenKeyStoreManager will be referred to as 'm' hereinafter

```
openstoremanager -domainspace "file:///export/domains" -manager m
```

Similar to the previous command, but the TrustDomains are stored in directory "/export/domains"

```
openstoremanager -domainspace "jss:jss" -manager m
```

This opens a special TokenKeyStoreManager which uses a TrustDomainManager which is dynamically constructed from the contents of a JSS permanent store, and any attached PKCS#11 tokens. It can be used to access cryptographic objects created by programs which use NSS but are not TrustDomains aware

```
setdefaultstoremanager -manager <localname>
```

Sets an already loaded `TokenKeyStoreManager` as the default for subsequent operations which require a `TokenKeyStoreManager`, but do not specify one explicitly

- `<localname>`: the nickname of an open `TokenKeyStoreManager`

```
unsetdefaultstoremanager
```

unsets any default `TokenKeyStoreManager` : after this operation any operation requiring a `TokenKeyStoreManager` will have to specify it explicitly, or a new default must be set using `setDefaultStoreManager`

```
liststores [ -manager <localname> ]
```

lists the `TokenKeyStores` managed by a `TokenKeyStoreManager`. If no `TokenKeyStoreManager` is explicitly specified, then the default is used

- `<localname>`: the nickname of an open `TokenKeyStoreManager`

## Commands for dealing with TokenKeyStores

These commands operate on objects in `TokenKeyStores` [ and their corresponding persistent representations in the associated `TrustDomain` ]. Each command can explicitly specify the `TokenKeyStore` to be operated upon, and the `TokenKeyStoreManager` responsible for that `TokenKeyStore`, or it can rely upon defaults having been set, or the global switches containing appropriate references to a `TokenKeyStoreManager` and `TokenKeyStore`. The following two switches can be used with any of the commands dealing with `TokenKeyStores` to explicitly specify a `TokenKeyStore` and `TokenKeyStoreManager`

```
[ -store <storename> [ -manager <localname> ] ]
```

- `<storename>`: name of the new store
- `<localname>`: the nickname of an open `TokenKeyStoreManager`



```
createstore -store <storename> [ -tokentype <tokentype:JSS> ] [
-pathvalidation <validationlist> ] [ -nodefault ]
```

Creates a new TokenKeyStore in a trust domain space associated with the TokenKeyStoreManager, which is either specified explicitly, or is a default. The type of the Token which will be used to activate the cryptographic objects created and imported into the new TrustDomain associated with the new TokenKeyStore may optionally be specified, as may the list of identifiers of CertificatePathValidators to be used to validate certificate chains in the TrustDomain

- <storename>: name of the new store
- <validationlist>: colon separated list of certificate path validation algorithms to be used for validating certificate chains in the TrustDomain. Defaults if not given
- <tokentype>: type of Token to use for crypto operations. Defaults if not given
- -nodefault: if given the new TokenKeyStore will not be made the default TokenKeyStore for subsequent operations

```
deletestore
```

Deletes a TokenKeyStore and it's associated TrustDomain from the trust domain space associated with the TokenKeyStoreManager specified explicitly or by default

```
setdefaultstore -store <storename> [ -manager <localname> ]
```

Sets the default TokenKeyStore to be used in subsequent commands where a TokenKeyStore is not specified explicitly

- <storename>: name of the store to make the default

```
unsetdefaultstore
```

Unsets the default TokenKeyStore : subsequent commands requiring a TokenKeyStore will have to specify one explicitly, or use setDefaultStore first

```
listaliases
```

List all the aliases in use in a TokenKeyStore

```
listcerts [-alias <alias> | (-issuer <issuerDN> -serial <serial#>)]
```

List TrustedCertificateEntrys from a TokenKeyStore. If a single certificate is referenced, either by alias or by issuer and serial number, then just that certificate's details are printed. If no single certificate is referenced, then the certificates from TrustedCertificateEntrys are printed

- <alias>: a TrustedCertificateEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x

```
examinecerts [ -file <filename> ]
```

Describe the certificates contained in an archive. The archive may contain one or more certificates in X.509 DER, Base-64 X.509 DER, PKCS#7 DER, or Base-64 PKCS#7 DER format

- <filename>: the name of a file to read certificates from. If no filename is given, then certificates will be read from stdin [i.e. should be pasted into the terminal ]

```
listkeys [-alias <alias> | (-issuer <issuerDN> -serial <serial#>)]
```

List KeyEntrys from a TokenKeyStore. If a single KeyEntry is referenced, by alias or by the issuer and serial number of the subject cert of it's certificate chain, then just that KeyEntry's details are printed. If no single KeyEntry is referenced, then a description of all KeyEntrys is printed

- <alias>: a TrustedCertificateEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x

```
genkey -dname <subjectDN> [-keysz <keysize:1024>] [-keyalg
<keyalg:RSA>]]
```

Generate a new KeyPair, and a self-signed certificate to accompany it in the resulting KeyEntry

- <subjectDN>: The RFC2253 encoded subject distinguished name of the generated self-signed certificate
- <keysz>: The size of key to generate. Defaults to 1024 bits
- <keyalg>: The key algorithm. Defaults to RSA. DSA is the only other possible value

```
certreq ( -alias <alias> | ( -issuer <issuerDN> -serial <serial#> )
) [ -dname <subjectDN> ] [ -file <filename> ] [ -der ]
```

Create a PKCS#10 certificate request using the PrivateKey from and existing KeyEntry. The KeyEntry may be identified using an alias, or alternatively by the issuer and serial number of it's associated subject certificate

- <alias>: a KeyEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x
- <subjectDN>: The RFC2253 encoded subject distinguished name of the generated self-signed certificate
- <filename>: The name of a file to save the PKCS#10 request in. If not given, it will be printed to the terminal
- -der: write output in raw DER rather than base-64 encoded DER [ works best with -file ]

```
deletecert [ -alias <alias> | ( -issuer <issuerDN> -serial <serial#> ) ]
```

Delete a TrustedCertificateEntry. The TrustedCertificateEntry to delete can be identified either by alias, or by issuer and serial number

- <alias>: a TrustedCertificateEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x

```
deletekey [ -alias <alias> | ( -issuer <issuerDN> -serial <serial#> ) ]
```

Delete a KeyEntry. The KeyEntry to delete can be identified either by alias, or by issuer and serial number

- <alias>: a TrustedCertificateEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x

```
addalias ( -alias <alias> | ( -issuer <issuerDN> -serial <serial#> ) ) -newalias <newalias>
```

Add an alias to an entry [ either a KeyEntry or a TrustedCertificateEntry ]. The entry can be identified either by alias, or by issuer and serial number. Aliases are unique in TrustDomains, so an attempt to add an alias which already exists in the TrustDomain associated with the TokenKeyStore in which the entry is present will fail.

- <alias>: an entry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x
- <newalias>: the new alias to add

```
removealias -alias <alias>
```

Remove an alias from an entry. The entry may be identified by either the alias to be removed or by issuer and serial number

- <alias>: an entry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x

```
importtrustedcerts [ -file <filename> ]
```

Import trusted certificates from an archive containing one or more certificates, encoded in X.509 DER, Base-64 X.509 DER, PKCS#7 DER, or Base-64 PKCS#7 DER. All the certificates in the archive will be added to the TrustDomain as TrustedCertificateEntries

- <filename>: the name of a file containing the certificate archive. If not given, the archive will be read from stdin

```
exportcerts ( -alias <alias> | ( -issuer <issuerDN> -serial  
<serial#> ) ) [-file <filename>] [-der] [-pkcs7]
```

Export certificates from a TrustedCertificateEntry or a KeyEntry. The entry may be referenced using an alias or an issuer and serial number. The certificates can be exported as X.509 DER, Base-64 X.509 DER, PKCS#7 DER, or Base-64 PKCS#7 DER

- <alias>: an entry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x
- <filename>: the name of a file to write the output to. If not specified, the output will be written to the terminal, which will be noisy with -der
- <-der>: write output as DER rather than Base-64 DER
- <-pkcs7>: write output as a pkcs#7 certificate archive

```
importkeychain [ ( -alias <alias> | ( -issuer <issuerDN> -serial  
<serial#> ) ) ] [-file <filename> ]
```

Import a chain of certificates to a KeyEntry. The KeyEntry may be referenced implicitly, or explicitly by an alias or by an issuer and serial number. If referenced implicitly, the public key in the subject certificate of the supplied chain will be used to select a KeyEntry. The operation will fail if more than one KeyEntry contains the same key, and in that case, the KeyEntry must be specified explicitly. The existing chain of certificates will be replaced by the chain retrieved from the archive, which may be encoded in X.509 DER, Base-64 X.509 DER, PKCS#7 DER, or Base-64 PKCS#7 DER. If the archive contains an incomplete chain of certificates, then an attempt will be made to complete the chain from certificates already in the store, but if this cannot be done then the import operation will fail

- <alias>: a KeyEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x
- <filename>: the name of a file to read the certificates from. If not specified, the output will be read from the terminal
- <-der>: write output as DER rather than Base-64 DER
- <-pkcs7>: write output as a pkcs#7 certificate archive

## Utility commands

These commands do not directly manipulate the objects in a `TokenKeyStore`, or a `TokenKeyStoreManager`, but do use the cryptographic objects in a `TokenKeyStore` for various useful operations. They will hopefully be useful for testing purposes

```
validate [ -file <filename> ]]
```

Validate a chain of certificates in a `TrustDomain`. A chain of certificates is valid in a `TrustDomain` if it correctly signed, has somewhere in its length a certificate which has a `TrustedCertificateEntry`, and passes the constraints imposed by the list of `CertificatePathValidators` associated with the `TrustDomain`

- `<filename>`: the name of a file to read the certificates from. If not specified, the output will be read from the terminal

```
verify -data <filename> -sig <filename> -sigalg <sigalg> ( -alias
<alias> | (-issuer <issuerDN> -serial <serial#> ) )
```

Verify a digital signature using a key from a `TokenKeyStore`. Both source data and the signature are given, and the `PublicKey` contained in the subject cert of the `KeyEntry` referenced by alias or by issuer and serial number is used to verify the signature

- `<alias>`: a `KeyEntry` alias
- `<issuerDN>`: RFC2253 encoded issuer distinguished name
- `<serial#>`: cert serial number. decimal or hex preceded by 0x
- `-data <filename>`: the name of a file to read source data from. must be specified
- `-sig <filename>`: the name of a file to read the signature from. must be specified
- `<sigalg>`: JCA name of the Signature algorithm to use

```
sign -data <filename> -sig <filename> -sigalg <sigalg> ( -alias
<alias> | ( -issuer <issuerDN> -serial <serial#> ) )
```

Generate a digital signature [ which may then be verified with verify ]. The source data is signed using the PrivateKey contained in the KeyEntry referenced by an alias or an issuer and serial number. The generated signature is written to a file

- <alias>: a KeyEntry alias
- <issuerDN>: RFC2253 encoded issuer distinguished name
- <serial#>: cert serial number. decimal or hex preceded by 0x
- -data <filename>: the name of a file to read source data from. must be specified
- -sig <filename>: the name of a file to write the signature to. must be specified
- <sigalg>: JCA name of the Signature algorithm to use

```
digest -data <filename> -digest <filename> -digestalg <digestalg>
```

Generate a message digest from some source data

- -data <filename>: the name of a file to read source data from. must be specified
- -digest <filename>: the name of a file to write the digest to. must be specified
- <digestalg>: JCA name of the Signature algorithm to use

```
httpsclient [-host <host:localhost>] [-port <port:443>] [-clientcert
<certalias>]
```

Perform an HTTPS client connection. Server certificates will be validated in the TrustDomain associated with the chosen TokenKeyStore, and client certificates will be chosen from the KeyEntries in the TokenKeyStore. A simple GET will be issued over the SSL connection, and the results printed to the terminal. If the Java system property ssl.debug is defined then additional debugging information concerning the server certificate validation and the client certificate selection will be printed

- <host>: hostname to connect to. defaults to localhost
- <port>: port to connect to. defaults to 443
- <certalias>: client certificate alias. if not specified, a suitable KeyEntry, which matches the server's specification of acceptable issuers, will be used



```
httpserver [-port <port:443>] -servercert <certalias> [-clientauth]
```

Await HTTPS client connections, returning a simple html file whatever request is received. Client certificates will be validated in the TrustDomain associated with the TokenKeyStore, and server certs will be chosen from the KeyEntries in the TokenKeyStore.

- <port>: port to listen on. Defaults to 443 [ which may require root privileges ]
- <certalias>: server certificate alias. must be specified.
- -clientauth if specified, a client certificate will be requested.

## Miscellaneous commands

Some miscellaneous and useful commands for working with the TokenKeyTool and it's environment

```
setglobals [<anyswitch>*]
```

Any switches specified here will be implicitly passed to subsequent commands [ although the same switches passed explicitly to those commands, as local switches, will override the global switches ]

```
listproviders
```

- List the properties content of all installed JCA Providers

Starting TokenKeyTool