Oracle® Application Server Certificate Authority

Administrator's Guide

10*g* Release 2 (10.1.2)

B14080-02

July 2005

ORACLE®

Oracle Application Server Certificate Authority Administrator's Guide, 10*g* Release 2 (10.1.2)

B14080-02

Primary Author: Vinaye H. Misra

Contributing Authors: Amit Agarwal, Olfat Ally, Howard Bae, Pratik Datta, Lakshmi Kethana, Belinda Leung, Mehul Poladia, Deepak Ramakrishnan, Gary Truong

Contents

3 OracleAS Certificate Authority Deployment Guidelines

4 Introduction to Administration and Certificate Management

5 Configuring Oracle Application Server Certificate Authority

6 Managing Policies in Oracle Application Server Certificate Authority

7 OracleAS Certificate Authority Administration: Advanced Topics

8 End-User Interface of the Oracle Application Server Certificate Authority

A Command-Line Administration

B Setting up a CA Hierarchy

C Troubleshooting OracleAS Certificate Authority

D Extensions

E Enabling SSL and PKI on SSO

F External Access to Protected OracleAS Certificate Authority

G S/MIME with OracleAS Certificate Authority

H Configuring OracleAS WebCache for OracleAS Certificate Authority

I The Oracle Application Server Certificate Authority Web Interface

Index

List of Tables

List of Figures

Preface

Oracle Application Server Certificate Authority (OCA) enables an organization to issue and manage digital certificates based on PKI (public key infrastructure) technology. With Oracle Application Server Certificate Authority's ease of administration and management, such certificates improve security and reduce the time and resources required for user authentication.

Oracle Application Server Certificate Authority enables end-entities (users and servers) to authenticate themselves using certificates that OracleAS Certificate Authority issues based on OracleAS Single Sign-On, SSL, or other pre-existing authentication methods. Use of these certificates makes authentication a speedier and more secure process, relying on certificate identification. Each certificate is published to Oracle Internet Directory when it is issued and removed when it expires or is revoked. Users can access the OracleAS Certificate Authority web interface to request issuance, revocation, or renewal of their own certificates. No special privilege is required for end-users to access the OracleAS Certificate Authority web interface. However, to get a certificate issued, revoked, or renewed, they must be already authenticated by OracleAS Single Sign-On or by SSL using a previously issued certificate from OCA. Otherwise, manual authentication by the OCA administrator is required.

This Oracle Application Server Certificate Authority Administrator's Guide explains how to perform administration and management of public key certificates.

This preface contains these topics:

Audience

This document is intended for

- administrators of Oracle Application Server Certificate Authority, who will manage certificate requests and certificate-related operations, and
- users of certificates issued by OCA, for authentication, encryption, and diverse other purposes.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Oracle Identity Management

Oracle Application Server Certificate Authority is a component of Oracle Identity Management, an integrated infrastructure that provides distributed security services for Oracle products and other enterprise applications. The Oracle Identity Management infrastructure includes the following components and capabilities:

- **Oracle Internet Directory**, a scalable, robust LDAP V3-compliant directory service implemented on the Oracle Database.
- **Oracle Directory Integration and Provisioning**, part of Oracle Internet Directory, which enables synchronization between Oracle Internet Directory and other directories and user repositories. This service also provides automatic provisioning services for Oracle components and applications and, through standard interfaces, for third-party applications.
- **Oracle Delegated Administration Services**, part of Oracle Internet Directory, which provides trusted proxy-based administration of directory information by users and application administrators.
- **Oracle Application Server Single Sign-On**, which provides single sign-on access to Oracle and third party web applications.
- **Oracle Application Server Certificate Authority**, which generates and publishes X.509 V3 PKI certificates to support strong authentication methods, secure messaging, and so on.

In addition to its use of SSL, OC4J, and HTTP Server, OCA has a built-in reliance on OracleAS Single Sign-On and Oracle Internet Directory. OracleAS Certificate Authority publishes each valid certificate in an Oracle Internet Directory entry for the DN in use, and supports certificate enrollment and saving or installing through Netscape, Internet Explorer, or Mozilla. OracleAS Single Sign-On and other components can rely on these Oracle Internet Directory entries because OracleAS Certificate Authority removes revoked certificates immediately from Oracle Internet Directory and, on a regular basis, expired certificates as well. The administrator also has the option of configuring OracleAS Certificate Authority to publish its URL through OracleAS Single Sign-On. This configuration choice causes every OracleAS Single Sign-On-authenticated user who lacks a certificate to see the OracleAS Certificate Authority page for requesting one. OracleAS Certificate Authority certificates can be used to authenticate to any Oracle component or to authorize use of any application that is OracleAS Single Sign-On-enabled.

In a typical enterprise application deployment, a single Oracle Identity Management infrastructure is deployed, consisting of multiple server and component instances. Such a configuration provides benefits that include high availability, information localization, and delegated component administration. Each additional application deployed in the enterprise then leverages the shared infrastructure for identity management services. This deployment model has a number of advantages, including:

- **One-time cost:** Planning and implementing the identity management infrastructure becomes a one-time cost, rather than a necessary part of each enterprise application deployment. As a result, new applications such as portals, J2EE applications, and e-business applications can be rapidly deployed.
- **Central management:** Managing identities is done centrally, even if administered in multiple places, and changes are instantly available to all enterprise applications.
- **User single sign-on:** Having a centralized security infrastructure makes it possible to realize user single sign-on across enterprise applications.
- **Single point of integration:** A centralized identity management infrastructure provides a single point of integration between the enterprise Oracle environment and other identity management systems, eliminating the need for multiple custom "point-to-point" integration solutions.

For more information about planning, deploying, and using the Oracle Identity Management infrastructure, see the Oracle Identity Management Administrator's Guide.

For the default deployment configuration of OCA, installation instructions appear in section 6.20 of the *Oracle Application Server Installation Guide*. For the recommended deployment configuration and installation procedure, see section 11.9 of that Guide.

Related Documentation

- *Oracle Application Server Installation Guide*
- *Oracle Application Server Administrator's Guide*
- *Oracle Application Server Security Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle Application Server High Availability Guide*
- *Oracle10i Backup and Recovery Advanced User's Guide*

- *Oracle Internet Directory Administrator's Guide*
- *Oracle Advanced Security Administrator's Guide*.

Many of the examples in this book use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle10i Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Public Key Infrastructure and OracleAS

public key infrastructure (PKI) is designed to enable secure communications over public and private networks. In addition, PKI provides for secure email, digital signatures for non-repudiation, and data integrity, among other things. One of the challenges that PKI has faced over the past 25 years has been an inability to deploy the necessary infrastructure associated with PKI. In fact, the cost and complexity of that infrastructure has been the primary factor limiting widespread use of PKI.

The Oracle Identity Management infrastructure provides an ideal environment for PKI, combining high availability, scalability, directory services, single sign-on, delegated administration service, and directory integration services. These advantages make this infrastructure an ideal place for the Oracle Application Server Certificate Authority to reside. As a result, the OCA is part of the Oracle Identity Management infrastructure, whose centralization and scalability automatically reduce the complexity and cost of deploying PKI.

This chapter takes a closer look at PKI and covers the following topics:

What Is a PKI?

A PKI integrates the following elements:

- Encryption algorithms to secure data transmission and storage
- Encryption keys to enable unique encryption for different users
- Key distribution methods to permit widespread secure use of encryption while preserving secure decryption by only the appropriate recipient
- Trusted entities to vouch for the relationship between a key and its legitimate owner

Together these components provide a high level of security for intranet, extranet, and e-commerce applications, as this chapter explains. The benefits include secure and reliable authentication of users, data integrity, non-repudiation of signed messages, and prevention of unauthorized access to transmitted or stored information.

This section examines key features of a PKI in the following topics:

- Registration Authority (RA)

Key Pairs

Encryption refers to obscuring data to protect it from unauthorized access or alteration, using some method that nevertheless allows authorized recipients to recover the original data. Techniques for scrambling or substituting for that original data often use a text or number called a key, known only to the sender and recipient. When both use the same key, the encryption scheme is called "symmetric." One difficulty with relying on a symmetric system is how to get that key to both parties without allowing an eavesdropper to get it, too, destroying the desired secrecy. Another problem is that a separate key is needed for every two people, so that each communicator must maintain many keys, one for each recipient.

The heart of a PKI is the use of private/public **key pair**s, termed "asymmetric" because the public and private keys are different. Each person has only one key pair, regardless of how many others he communicates with.

Each key in a PKI consist of a binary number, typically from 512 to 2048 bits; 512 is considered weak encryption, 1024 is considered very strong encryption, and 2048 is considered military grade. An algorithm combines these key bits with data bits in a way that encrypts the data.

Each key pair owner keeps his private key secret while making his public key available. Others can use the public key to encrypt private messages that they wish to send to the key pair owner. The key pair owner, in turn, uses the private key to decrypt the messages or to sign critical messages he sends out. The efficacy of the system rests on the idea that the public key can be distributed easily and securely while the private key required for decryption is never shared at all.

Certification Authority (CA) and Digital Certificates

A certification authority is a trusted third-party that vouches for the public key owner's identity. Oracle Application Server Certificate Authority, the subject of this book, is one such entity. Others include Verisign and Thawte. The certification authority validates the public key's link to a particular person by creating a **digital certificate**. This digital certificate contains the public key and information about the key holder and the signing certification authority. Using a PKI certificate to authenticate one's identity is analogous to identifying oneself with a driver's license or passport. Such certificates are almost impossible to forge or alter.

This section covers the following topics:

- CA Signing
- Levels of Trust
- Contents and Uses of a Digital Certificate
- Containers for PKI Credentials

CA Signing

The CA signs the digital certificate with its private key. This signature enables anyone to use the CA's public key to verify that the signature is authentic and that the certificate is therefore valid. Once the certificate is validated, the owner's public key can be used with confidence to encrypt messages to the certificate's owner or to validate the owner's signature on messages.

Levels of Trust

There can be many levels of CAs. A hierarchy of trust is established when each CA receives its certificate from a more trusted source, that is, a higher-level CA. Each line of trusted links from the root CA through subordinate CA's down to lower level trust points is called a trusted path.

The top-level CA is called the root CA, and is the origin of the trust relationship. CAs below the root CA are called subordinate CAs. All end users sharing the same root CA can communicate with each other in trusted ways because they all trust the same ultimate source of authentication.

Trusting a certificate to legitimately represent prior verification of an identity linked with a public key means trusting the authority that issued the certificate: the CA. CAs in turn often rely on another entity, a registration authority (RA), to validate the information supplied on requests for certificates.

Contents and Uses of a Digital Certificate

Digital certificates issued by OracleAS Certificate Authority comply with the X.509, Version 3, ISO standard and with RFC 2459, promulgated by the PKIX working group of The Internet Engineering Task Force, `http://www.ietf.org/`.

The X.509 v3 standard introduced extensions enabling separate certificates for SSL, encryption, and digital signatures. An X.509 v3 certificate contains the following user information:

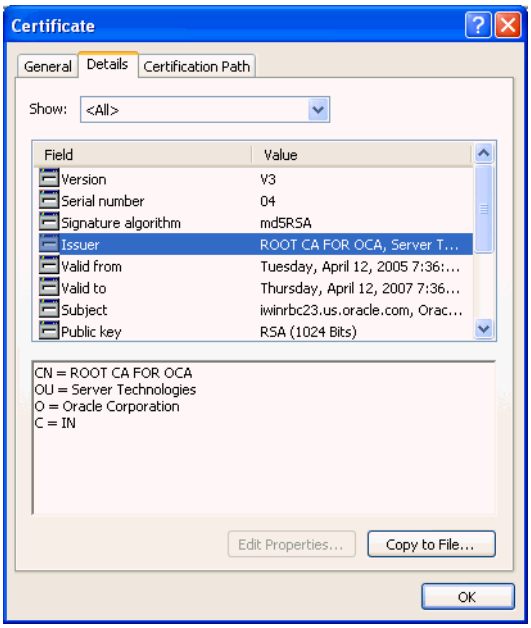
- Certificate owner's distinguished name (DN)
- DN of the certification authority that issued the certificate.

> **Note:**
>
> For a DN, the DC and EMAIL components must use only printable (ASCII) characters.
>
> This restriction means that even in a locale that uses a multibyte character set, the DC and EMAIL components for Distinguished Names must still use ASCII characters.

- Certificate owner's public key
- Certificate issuer's digital signature
- Dates during which the certificate is valid
- Certificate serial number

Figure 1–1 shows a newly issued certificate that contains all of these elements.

Figure 1–1 A Certificate Issued by OracleAS Certificate Authority



OCA issues and works with X.509 certificates, supporting multiple certificate types, and with X.509 CRLs (certificate revocation lists).

Containers for PKI Credentials

Containers are used to hold the various related credentials used for PKI operations like signing or verifying messages. The data structures in such a container securely store a user's private key, certificate, and a list of root certificates that the user trusts. The trusted certificates are used to verify a peer identity in an SSL connection or to verify a received signature. In browsers such as Netscape or Internet Explorer, the container for certificates can be called a certificate database or certificate cache. In the Oracle Identity Management Infrastructure, such a container is called an Oracle wallet.

Registration Authority (RA)

A **Registration Authority (RA)** is an optional system to which a CA delegates certain management functions, such as verification and certification of end-entity identification. It acts as an interface between a CA and the user. The RA receives requests to issue new certificates, to renew expired certificates, and to revoke certificates. The RA evaluates identification supplied by the requestor to verify that the requestor is who it claims to be. For existing certificates, the RA verifies the association of the requestor with the supplied identification and public key and sends the approved request to the CA.

NOTE: In OracleAS, the RA functions are performed within the Oracle Application Server Certificate Authority product itself.

Benefits of a PKI

A PKI has the following benefits:

- Secure and reliable authentication of users

Reliable authentication relies on two factors. The first factor is proof of possession of the private key part of the public/private pair, which is verified by an automatic procedure that uses the public key. The second factor is validation by a certification authority that a public key belongs to a specific identity. A PKI-based digital certificate validates that identity connection based on the key pair.

- Data integrity

  Using the private key of an established public/private key pair to sign digital transactions makes it difficult to alter the data in transit. This "digital signature" by the user X is a coded digest of the original message encrypted by X's private key. Recipients can readily use X's corresponding public key to verify that the message has not been altered and that it was in fact sent by X. Any change to the message or the digest would have caused a failure of the attempted verification using the public key, telling the recipient not to trust it.

- Non-repudiation

  A digital signature also makes it difficult for the message originator to disown the message.

- Prevention of unauthorized access to transmitted or stored information

  The time and effort that would be required to derive the private key from the public key makes it unlikely that the message would be decrypted by anyone other than the key pair owner.

Introduction to the OracleAS PKI

This section introduces the OracleAS PKI. It covers the following topics:

Earlier Costs and Difficulties

Prior to the OracleAS PKI, acquiring a certificate to use for authentication was a process with many steps and delays. You had to acquire the appropriate form, fill it in precisely, and deliver it to the proper registration authority. Once that authority had validated your identity and returned the approved form to you, you then had to deliver it to the certificate authority, which would process this approved form and issue the actual certificate. Delivery often entailed cutting and pasting the approved request's contents into a different form. Once the certificate authority had received this new form, it could take days or weeks to receive the actual certificate.

Benefits of the OracleAS PKI

The OracleAS PKI removes and replaces most of those earlier steps and delays with their inherent costs and difficulties. It tightly integrates the authentication function, the user repository, and applications. It relieves users of the burden of requesting a certificate from a third party and personally submitting it to applications and to a central directory.

Oracle Application Server Certificate Authority, the centerpiece of the OracleAS PKI, provides an easy, one-stop solution, with an easy-to-use Web interface and a Registration Authority (RA) integrated into the CA. The user submits a request online, provides authentication information, and acquires a certificate automatically. This

certificate is then automatically linked to the user's entry in Oracle Internet Directory, enabling Single Sign-on to authenticate a user by checking against the corresponding directory entry. Indeed, this Identity Management Infrastructure and OCA are used by many other Oracle components, including the database and Oracle Collaboration Suite.

Once the user is issued a certificate, it can take the place of single sign-on credentials. It thus enables immediate access to all single sign-on applications configured for PKI as well as to those whose authentication requirements are less stringent than PKI. As noted earlier, the user's key pair also enables digital signatures, with their attendant integrity and non-repudiation assurances.

Components of the OracleAS PKI

The OracleAS PKI complies with industry-standard specifications, using the following components:

- Containers, Oracle Wallets, and Oracle Wallet Manager (OWM)
- Secure Sockets Layer (SSL)
- Oracle Internet Directory and Single Sign-on (SSO)
- Oracle Application Server Certificate Authority

Containers, Oracle Wallets, and Oracle Wallet Manager (OWM)

Several international standards define the form and content of a certificate and a container for certificates. As described in "Contents and Uses of a Digital Certificate", the X.509 version 3 standard provides these specifications for certificates. The PKCS#12 (Personal Information Exchange Syntax) standard provides specifications for containers.

Users with standard existing PKI credentials can export them in PKCS#12 format and import (install) them into browsers, such as Netscape Communicator or Microsoft Internet Explorer, or into Oracle Wallet Manager. The PKCS#12 standard thus increases interoperability and reduces the cost of PKI deployment for organizations.

> **See Also:** The following sections in Chapter 8:
>
> "Importing a Newly Issued Certificate to Your Browser"
>
> "Exporting (Backing up) Your Wallet from Your Browser"
>
> "Importing a Certificate from Your File System"

Oracle Wallet Manager facilitates acquiring, using, and storing such certificates. It provides a graphical user interface that standardizes the normal operations done with or to such certificates and their containers, which in OracleAS are termed Oracle wallets.

> **See Also:** *Oracle Advanced Security Administrator's Guide*

In fact, a server administrator can use OWM to create a PKCS#10 certificate request. After OWM generates the completed request, the administrator can save it to the file system or copy it for pasting into OCA's Server/SubCA form for requesting an OracleAS Certificate Authority certificate. See the last link in the See Also references given earlier.

These wallets conform to the PKCS#12 standard, and are the containers used by OCA. Their interoperability with third-party applications such as Netscape Communicator and Microsoft Internet Explorer provides valuable portability across operating systems.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) is the most widely used protocol for securing the Internet. It uses public key cryptography to enable authentication, encryption, and data integrity. Using these tools, SSL also enables secure session key management by encrypting a unique one-time session password for use by both server and client. After this password is securely sent and received, it is used to encrypt all subsequent communications between server and client, making it infeasible for others to decipher those messages. All server components like Oracle HTTP Server, WebCache, Oracle Internet Directory, and the Oracle database use SSL to enable secure communication.

Oracle Internet Directory and Single Sign-on (SSO)

Oracle Internet Directory is an LDAP, Version 3, directory. LDAP stands for Lightweight Directory Access Protocol. This directory enables PKI-based single sign-on by providing the central repository for such authentication credentials, including publishing the certificates issued by OCA. Oracle Internet Directory enforces attribute-level access control, restricting read, write, or update privileges on specific attributes to specific users. It supports the use of SSL to protect and authenticate directory queries and responses.

Oracle Application Server Certificate Authority

A new addition to the OracleAS product suite, OracleAS Certificate Authority can be used to administer and manage the entire certificate life-cycle. This life-cycle includes recording and processing requests for new certificates, verifying user credentials, and issuing, renewing, or revoking these certificates. In the past, these processes required separate record-keeping and cut-and-paste operations that were tedious and sometimes error-prone.

With OCA, a few clicks generates, submits, and stores a certificate. As a result, credential verification and authentication is simple and fast.

OCA is an optional infrastructure component in Oracle Application Server.

2

Identity Management and OracleAS Certificate Authority Features

Oracle Application Server Certificate Authority provides secure mechanisms whereby it creates and signs X.509 v3 digital certificates for clients and servers. OracleAS Certificate Authority enforces policies chosen or created by its administrator, as described in Chapter 6, and is controlled by that administrator through the scalable web-based interface described in Chapter 5. OracleAS Certificate Authority provides a secure infrastructure for supporting and managing such certificates, including the web-based user interface described in Chapter 8.

This chapter describes the architecture enabling OracleAS Certificate Authority features and operations, in the following sections:

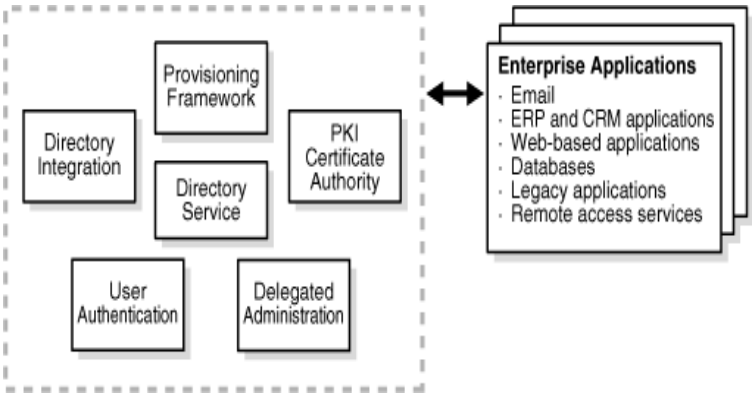
Identity Management Components and Architecture

A complete identity management solution includes the following components:

- A scalable, secure, and standards compliant directory service for storing and managing the user information.
- A user provisioning framework that can either be linked to the enterprise provisioning system (such as an HR application), or that can be operated standalone.
- A delegated administration model and application that allows the administrator of the identity management system to selectively delegate access rights to the administrator of the individual application, or to the end-user directly.
- An appropriate security model, and user-interface model, that can support diverse requirements is critical.
- A directory integration platform that enables the enterprise to connect the Identity Management directory with legacy or application-specific directories.
- A run-time model and application for user authentication.
- A system to create and manage PKI certificates.

A model for an enterprise identity management solution is shown in Figure 2–1.

Figure 2–1 A Model for Enterprise Identity Management Solution



The Oracle Identity Management Infrastructure is discussed further in the following sections:

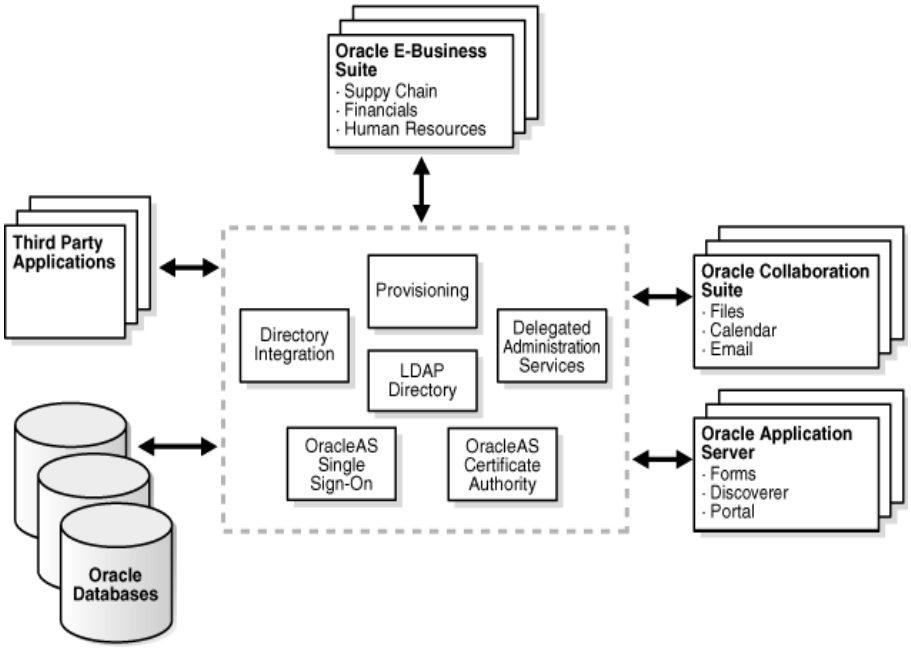
- Oracle Identity Management
- Leveraging Oracle Identity Management in the Enterprise
- Role of Oracle Identity Management in the Oracle Security Architecture
- Role of OracleAS Certificate Authority in Oracle Identity Management
- Simplified Provisioning through SSO Integration

Oracle Identity Management

Oracle Identity Management is an integrated infrastructure that Oracle products rely on for securing users and applications across the enterprise. Oracle Application Server is the primary release vehicle for Oracle Identity Management; however, it also ships as part of the infrastructure with other Oracle products. The Oracle Identity Management infrastructure includes the following components:

- Oracle Internet Directory, a scalable, robust LDAP V3-compliant directory service implemented on the Oracle Database.
- Oracle Directory Integration and Provisioning that permits synchronization between Oracle Internet Directory and other directories and automatic provisioning services for Oracle components and applications and, through standard interfaces, third-party applications.
- Oracle Delegated Administration Service, which provides trusted proxy-based administration of directory information by users and application administrators. This can be leveraged by applications such as portal, email, and others.
- Oracle Application Server Single Sign-On, which provides end-users single sign-on access to Oracle and third-party web applications.
- Oracle Application Server Certificate Authority, which generates and publishes X.509 V3 certificates to support PKI based strong authentication methods.

Figure 2–2 Enterprise-Integrated Identity Management



Leveraging Oracle Identity Management in the Enterprise

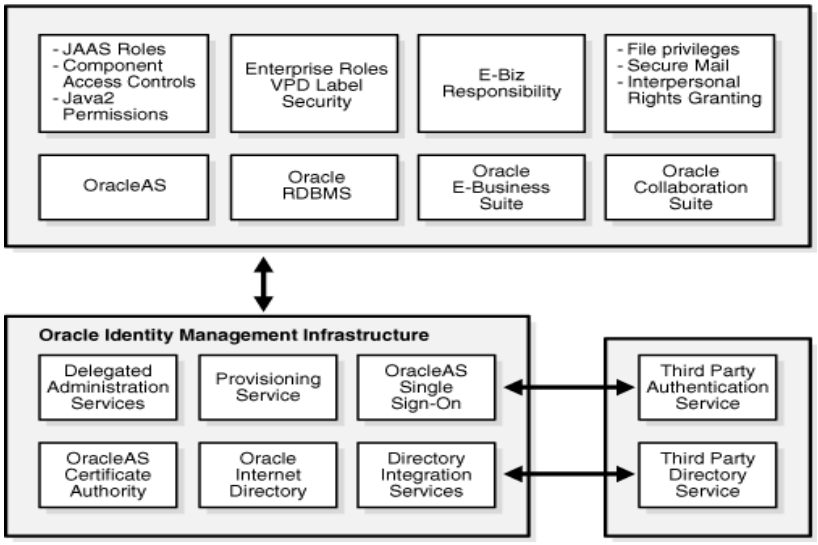
While Oracle Identity Management is designed to provide an enterprise infrastructure for Oracle products, it also serves as a robust and scalable identity management solution for custom and third-party enterprise applications, hardware and network operating systems of the enterprise.

In addition, Oracle works with third-party application vendors to ensure their applications can leverage Oracle Identity Management out of the box.

Role of Oracle Identity Management in the Oracle Security Architecture

Each of the Oracle technology stacks (namely, Oracle Database (the RDBMS), Oracle Application Server 10g, the E-business Suite, and Oracle Collaboration Suite) supports a security model that is appropriate for its design center. Nevertheless, they all employ the Oracle Identity Management infrastructure for implementing their respective security models and capabilities. Figure 2–3 diagrams this architecture:

Figure 2–3 Oracle Identity Management Security Model



OracleAS supports a J2EE-compliant security service called Java Authentication and Authorization Service (JAAS). JAAS can be configured to utilize users and roles defined in Oracle Internet Directory. Similarly, the database security capabilities, "Enterprise User" and "Oracle Label Security" provide the means to leverage users and roles defined in the Oracle Internet Directory. Both these platforms, thus, facilitate the applications developed using their respective native security capabilities to transparently leverage the underlying Identity Management infrastructure.

Oracle Collaboration Suite and the Oracle E-Business Suite are application stacks layered over the RDBMS and iAS platforms. As described earlier, this layering itself brings a level of indirect integration with the Oracle Identity Management infrastructure. In addition, these products also have independent features that are Oracle Identity Management reliant. For instance, Oracle Collaboration Suite components such as E-Mail and Voice mail use the Oracle Internet Directory to manage product-specific user preferences, user personal contacts, address book and so on. These components rely on OCA for enabling secure email.

These Oracle technology products also leverage the Provisioning Integration services to automatically provision and de-provision user accounts and privileges. The Delegated Administration Service is employed extensively for self-service management of user preferences and personal contacts. Also, the security management interfaces of these products leverage the user and group management building blocks called the "service units."

Role of OracleAS Certificate Authority in Oracle Identity Management

Oracle Application Server Certificate Authority leverages the Oracle Identity Management Infrastructure through its use of Oracle Internet Directory and Single Sign-On. The directory enables publishing certificates upon issuance and propagating the information to all connected databases. Single Sign-on provides the standard interface relied upon by applications and other Oracle components, such as the enterprise user and secure email facilities in Oracle Collaboration Suite. The certificates issued by OCA support the secure authentication needed for simple, fast, consistent identity management.

Simplified Provisioning through SSO Integration

An application user authenticating to OracleAS Single Sign-On can seamlessly obtain a certificate without technical education or understanding of PKI. The application can thereafter use the newly issued certificate for transparently authenticating that application user to OracleAS Single Sign-On, providing increased security. The issued PKI certificate is automatically published in the Oracle Internet Directory. In providing this powerful functionality, Oracle leverages the security, high availability and scalability of the Oracle Database, Oracle Internet Directory, and OracleAS Single Sign-On.

The OCA administrator can optionally configure OracleAS Certificate Authority to broadcast its URL through OracleAS Single Sign-On. Doing so enables users authenticating through OracleAS Single Sign-On to use OCA's easy graphical interface to apply for a certificate. Having such a certificate makes future OracleAS Single Sign-On authentication even easier, because OracleAS Single Sign-On can then use Oracle Internet Directory to validate the certificate automatically supplied by the user's browser. OracleAS Single Sign-On can rely on the information in the directory because OracleAS Certificate Authority automatically deletes revoked and expired certificates from the directory on a regular basis.

Third Party PKI Support in Oracle Identity Management

While OCA is part of Oracle Identity Management and Oracle products are tightly integrated, Oracle products also work with any standards-compliant certificate authority. Oracle Wallet Manager, the certificate provisioning tool, will work with any X.509-v3-standard-compliant certificate authority.

> **See Also:** For detailed information on Oracle Wallet Manager, see *Oracle Application Server Administrator's Guide* and *Oracle Application Server Security Guide*.

Oracle Wallet Manager can support any existing server certificates that are presented in PKCS#12 format. For instructions on importing such certificates, see the section entitled "Importing Certificates and Wallets Created by Third Parties" in *Oracle Application Server Administrator's Guide*.

Oracle Application Server Single Sign-On and Oracle Internet Directory work with any third-party standards-compliant certificate authority. For instructions on how to load certificates from such a third-party into Oracle Internet Directory and enable them for PKI authentication with Single Sign-On, see Chapter 7 of *Oracle Application Server Single Sign-On Administrator's Guide*.

Key Features of Oracle Application Server Certificate Authority

OracleAS Certificate Authority's key features are accessible through a scalable, web-browser interface. These features support administering industry-standard certificates, integrating with LDAP directories, and applying policies, as described in the following sections:

- Support for Open Standards
- Flexible Policy
- Ease of Use for Administrators and End Users
- Globalization Support for OCA Screens
- Scalability, Performance, and High Availability

- Secure Email Through S/MIME Digital Encryption and Signing

Support for Open Standards

OracleAS Certificate Authority supports open standards, assuring organizations that they will be able to communicate with heterogeneous computing environments. OCA supports the following standards:

- X.509 version 3 certificates and certificate revocation lists (CRLs)
- IETF PKIX standard
- Signature key lengths of up to 4096 bits (RSA)
- Smart cards
- Certificate requests using Microsoft Internet Explorer and Netscape Communicator
- Various PKCS Standards (5, 7, 8, 10, 12, and so on.)
- Multiple enrollment protocols for certificate requests such as Signed Public Key and Challenge (SPKAC) and Public Key Cryptography Standard (PKCS) #10 for certificate requests
- S/MIME (Secure Multipart Internet Mail Extensions)

Flexible Policy

A policy is a set of rules and restrictions that limits the actions, access, or authorizations that users are permitted to use. Oracle Application Server Certificate Authority provides a set of configurable policy rules that can be used to restrict the certificate properties that a user (or a group of users) can get. A site can customize these rules to configure OCA for its particular PKI requirements. A few default policy rules are provided, and customers can develop and apply their own policy rules as well.

Ease of Use for Administrators and End Users

The administrative web interface for Oracle Application Server Certificate Authority provides two primary tabs: Certificate Management and Configuration Management. To use them, the administrator must enroll by filling out a form upon first entry and then importing (installing) his certificate.

The Certificate Management tab gives the administrator the ability to approve or reject certificate requests and to generate or update CRL's (Certificate Revocation Lists). The administrator can also revoke issued certificates for various reasons, for example, if security has been compromised. (Stopping and starting OracleAS Certificate Authority require the administrator to use the command-line tool `ocactl`, which requires his password.)

The end-user web interface for OCA also provides two tabs: a User Certificates tab and a Server/SubCA Certificates tab. When you click the User Certificates tab, you can use your Oracle Single sign-on name and password to authenticate yourself. When you choose OracleAS Single Sign-On authentication and click **Submit**, an OracleAS Single Sign-On window appears in which you can enter your OracleAS Single Sign-On username and password.

When the User Certificates page appears, it shows you all certificate requests and their status (pending, approved, rejected), among other information. You can request a new

certificate, save the CRL (Certificate Revocation List) to disk or install it in your browser, or change your method of authentication.

When you click the Server/SubCA Certificates tab, you can request a new Server/SubCA certificate, save the CRL to disk or install it in your browser, or save or install the CA certificate. You can also search for particular certificates or certificate requests by ID/Serial number or by common name.

See Also:
- Chapter 5, "Configuring Oracle Application Server Certificate Authority" for details of the administrative interface
- Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority" for details of the end-user interface

Globalization Support for OCA Screens

The administrative and user screens for OracleAS Certificate Authority can appear in the language of the client or of the server, if certain prerequisites are met. The database character set must be UTF8, and the required language must be one of the many that OracleAS Certificate Authority supports; otherwise English is the language used. While the administrative command line tool, `ocactl`, uses only commands in English, messages (informational, error messages, and so on) are displayed in the language of the server locale, if supported; otherwise English appears.

See Also: "Configuring Globalization Support for Screens" in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics"

Scalability, Performance, and High Availability

OracleAS Certificate Authority automatically attains these benefits through integration with OracleAS as the application server and with the Oracle database as the repository for the following information:

- Users, roles, and privileges
- Pending and approved certificate requests
- Certificates issued
- Logging of user activities and JAZN authentication information

Secure Email Through S/MIME Digital Encryption and Signing

An OracleAS Certificate Authority administrator can use OCA's command line tool to create an **S/MIME** certificate and wallet readily used by OCA and email clients (Outlook, Mozilla/Netscape). Sending and receiving encrypted or signed email becomes easy. The OCA administrator can use the secure web interface to configure OCA notifications and alerts to use S/MIME.

See Also:

To create the SMIME wallet required for encrypted or signed email, see "Regenerating the CA SSL and CA S/MIME Wallets" in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics".

To configure OCA to use SMIME, see "Notification Sub-tab" in Chapter 5, "Configuring Oracle Application Server Certificate Authority".

For general SMIME operations, see Appendix G, "S/MIME with OracleAS Certificate Authority".

Automatic or Manual Provisioning of Certificates

Manual provisioning has an administrator issuing certificates to users. The automatic provisioning provided by Oracle Application Server Certificate Authority using OracleAS Single Sign-On and SSL reduces the costs and delays of conventional methods for supporting PKI.

For OracleAS Single Sign-On authentication, OracleAS Certificate Authority uses mod_osso and OracleAS Single Sign-On. These methods simplify certificate management by helping OracleAS Certificate Authority issue certificates to users who have been authenticated automatically by OracleAS Single Sign-On.

A user who has previously been issued an X.509v3 certificate can submit that certificate over HTTPS as a means of authenticating to OracleAS Certificate Authority. Assuming the certificate was issued by the same OracleAS Certificate Authority and has not been revoked, the certificate request will be approved automatically. Swift approval allows the user to get additional certificates for encryption or signing without the delay of waiting for the administrator or security officer to approve the request.

OracleAS Certificate Authority can also support smart cards through Netscape and Internet Explorer integration, and display its forms in the language determined by the browser's locale setting.

OracleAS Certificate Authority supports these authentication methods, as explained in the following sections:

- Oracle Single Sign-on Authentication
- Certificate-based Authentication Using Secure Socket Layer (SSL)
- Manual Approval

Oracle Single Sign-on Authentication

OracleAS Single Sign-On and Oracle Internet Directory constitute the default user management and authentication platform. Oracle Application Server Certificate Authority uses Oracle Internet Directory as the storage repository for certificates. This architecture provides centralized certificate management, simplifying certificate provisioning and revocation.

OracleAS Certificate Authority's integration with OracleAS Single Sign-On Server and Oracle Internet Directory provides seamless certificate provisioning mechanisms for applications relying on them. A user provisioned in the Oracle Internet Directory and authenticated to OracleAS Single Sign-On can choose to request a digital certificate from OracleAS Certificate Authority. OracleAS Single Sign-On can make this easy by

displaying a "get certificate" pop-up page, if OracleAS Certificate Authority is configured as explained in the section entitled "Simplified Provisioning through SSO Integration". The user can authenticate with username/password, an existing SSL certificate, or both. The user simply clicks the **Request a Certificate** button and a certificate will be automatically and immediately provisioned in the Oracle Internet Directory.

This method leverages the ability of OracleAS Single Sign-On to identify the user and to populate required fields in the certificate request by using data from Oracle Internet Directory. Similarly, the Oracle Certificate Authority administrator or certificate owner can revoke a certificate in real time, automatically causing it to be deleted from Oracle Internet Directory. Future attempts to use that certificate for OracleAS Single Sign-On authentication will then fail.

For more information, see Appendix E, "Enabling SSL and PKI on SSO".

Certificate-based Authentication Using Secure Socket Layer (SSL)

Oracle Application Server Certificate Authority supports certificate-based authentication, so a user's prior, unrevoked X.509 v3 certificate will authenticate that user to OracleAS Certificate Authority over HTTPS. Having thus authenticated the user, OracleAS Certificate Authority can automatically issue a new certificate for SSL, for signatures, or for other purposes without delay.

Manual Approval

An organization's security policy can dictate that requests for certificates be approved manually rather than allowing certificates to be issued by an automatic process. If this choice is made, the more conventional manual mode of approval and authentication will be used, and the Single Sign-on and SSL modes will be turned off. OracleAS Certificate Authority can enforce such an approval process, requiring an administrator or security officer to manually verify the identify of the requestor.

For manually approved authentication, the certificate requests acceptable to OracleAS Certificate Authority use the basic input fields required by all CAs. This manual process requires the user to provide personal information, such as name, email address, and location. (Users can optionally supply advanced DN attributes, such as domain components, customizing the certificate request.) The manual method is considered more complex than Oracle Single Sign-on Authentication or Secure Socket Layer Authentication. However, it also affords users the additional options to view and save or install existing certificates. Server and subordinate CA's can also request certificates using this manual process.

Hierarchical Certificate Authority Support

OracleAS Certificate Authority supports a hierarchy of certificate authorities. In a hierarchical PKI, the root CA for a security domain is the original single CA that is ultimately trusted by all users. Its identity serves as the beginning of trust paths.

OracleAS Certificate Authority can be a root CA. It can also certify the certificate of another CA, thereby creating a subordinate CA. Alternatively, the signing/SSL certificate of a subordinate installation can be obtained from another OracleAS Certificate Authority installation or any standards-compliant certificate authority. This subordinate CA can in turn issue certificates to even lower-level CAs. Because each authority's certificate is signed by a higher CA, a user can verify the certificate chain by tracing the certificate authority path back to a higher authority he trusts, or to the root CA.

Obtaining the sub/CA certificate from a separate certificate authority is useful when a PKI infrastructure is already in place. Hierarchical CA support is useful in a geographically distributed organization.

See Also: Appendix B, "Setting up a CA Hierarchy"

Using a hierarchical CA also provides important additional benefits in cost and safety, enabling a sub-CA to conduct normal operations while the root CA is especially protected. Such protection can include being off-line in a highly secure location. In this way, even if an online subordinate CA is compromised, it can be revoked and a new sub-CA created to replace it. All earlier operations can continue using certificates as issued. However, if the root CA is compromised, a completely new infrastructure needs to be established, and all applications relying on it need to be updated.

3

OracleAS Certificate Authority Deployment Guidelines

This chapter helps you understand CA deployment issues and gather the information you need for a smooth, effective deployment. Addressing these issues requires a sound understanding of creating, securing, transmitting, and guaranteeing the integrity of certificates. This chapter describes key capabilities and helps you plan your OracleAS Certificate Authority deployment to meet your site's certificate security requirements. It contains these sections:

Road Map for Setting up a Certificate Authority

Oracle Application Server Certificate Authority is a trusted entity that provides a secure infrastructure for the use of digital certificates. Certificates find many diverse applications, for example:

- Securing e-commerce with SSL encryption and authentication for Web servers
- User authentication
- Securing e-mail
- Digitally signing documents

As the administrator tasked with deploying OracleAS Certificate Authority for these and other applications, you must ensure that it is deployed and managed in a manner that enables it to be a trusted entity for your site's digital transactions.

You must have a sound understanding of the following areas:

- Components of the OracleAS PKI, and OracleAS Certificate Authority's features and capabilities

  For a detailed treatment of this subject, see Chapter 1, "Public Key Infrastructure and OracleAS".

  OracleAS Certificate Authority is a component of the Oracle Identity Management infrastructure and leverages capabilities of this infrastructure. For examples, certificates issued by OracleAS Certificate Authority are published to Oracle

Internet Directory. Users authenticated by Oracle Application Server Single Sign-On can obtain certificates seamlessly without prior knowledge of PKI. For details about the Oracle Identity Management infrastructure, see Chapter 2, "Identity Management and OracleAS Certificate Authority Features".

- How many users and servers will need certificates, and for what applications

  You will need a thorough survey to determine end user requirements such as: How many users and servers will need certificates? What types of certificates will they need? What guidelines do users need when requesting or using a certificate? What should be the life span of user and server certificates? Also take into account anticipated growth needs if possible.

  For information about these and other certificate design issues, see "Certificate Requirements and Policies".

- What deployment topology best fits the site's functional needs

  You must consider the unique needs of your organization and be able to answer questions like: What kinds of trust relationships do I need with external CAs? Do I need a centralized or distributed certificate authority? Do any components need to be outside the firewall? Can my CA installation support growth and changes in the organization's structure?

  Issues relating to the design of a CA infrastructure are addressed in "Planning your OracleAS Certificate Authority Architecture". Network and architectural issues are presented in "Deployment Considerations and Base Scenarios".

- Security requirements

  The certificate authority, being the central component of the PKI, must be housed in a secure facility. You will need to define where servers, storage devices and related components are to be located. Consider appropriate safeguards to prevent unauthorized access to CA components and to ensure that they are operated and maintained by qualified personnel.

- Availability requirements

  Given the strategic role played by a certificate authority, it is important to build in failover capabilities to safeguard against malicious attacks, component failures, and natural disasters. For more information, read "High Availability Deployment Options".

- Testing and pilot deployment phase

  These phases are important for verifying the efficiency, effectiveness, and security of the CA's operation and procedures.

- Training

  Training must be tailored to the needs of end users, help desk personnel, security officers, and administrators.

- Putting it all together

  With an appropriate understanding of PKI and OracleAS Certificate Authority, you are ready to address the critical implementation tasks:

  – Defining PKI policies, certificate policies, and the Certificate Practice Statement (CPS). See:

    * Certificate Requirements and Policies
    * Appendix D, "Extensions"

- Designing the certificate authority hierarchy. Your design should consider the optimum number of levels in the hierarchy, which depends on such factors as the user population. See:
  - CA Trust Hierarchy
  - Levels of Trust
- Implementing the CA hierarchy by installing and configuring the necessary CA instances.
- Securing the components of the CA hierarchy to protect against intrusion attempts and attacks. See:
  - Securing the CA
  - *Oracle Application Server Security Guide* for a discussion of the Oracle Application Server security architecture
- Setting up certificate validation with the Certificate Revocation List (CRL). See "Updating the Certificate Revocation List (CRL)".
- Defining roles and responsibilities for the entities who interact with the CA, including the CA administrator, the web administrator, end users and so on.
- Deciding on your site's availability needs and configuring the environment to meet those needs. See "High Availability Deployment Options".
- Establishing trust with external enterprises as needed. See Figure 3–5.

See "Implementation Checklist" and "Use Case: MyPKIsite.com" for a list of essential planning data and a use case illustrating these decision points as an organization plans and implements an OracleAS Certificate Authority deployment.

Certificate Requirements and Policies

You must plan the following components of your OracleAS Certificate Authority deployment to effectively meet your organization's certificate needs:

- Define Certificate Requirements and Properties
- Define Certificate Policies and Practices
- Define CRL Policies
- Define Alerts and Notifications

Define Certificate Requirements and Properties

In the initial planning stages, you must consider the certificate life cycle and the types of certificates to use, taking into account the requirements of your user base. This section describes:

- Certificate Provisioning
- Certificate Types
- Certificate Properties
- Certificate Renewal and Revocation
- Distributing the CA Certificate

Certificate Provisioning

OracleAS Certificate Authority can provision user certificates in manual or automatic mode:

- Manual provisioning, which is the conventional means for providing certificates, dictates that the CA administrator approve certificate requests manually. OracleAS Certificate Authority can enforce a manual approval process, turning off Single Sign-on and SSL provisioning modes.

  This approach is suitable for organizations that need to be conservative when issuing certificates, and when the volume of requests is not too large.

- For automatic provisioning, OracleAS Certificate Authority provides a choice of OracleAS Single Sign-On and SSL mechanisms:

  – A user authenticated to OracleAS Single Sign-On, and provisioned in Oracle Internet Directory, can request a digital certificate from OracleAS Certificate Authority by providing a username and password, or other configured single sign-on authentication mechanism

  – An existing certificate can authenticate the user with SSL and allow OracleAS Certificate Authority to automatically issue a new certificate

  Benefits of automatic provisioning include reduced costs and delays compared to the manual approach. Automatic provisioning is recommended if you have already deployed an identity management solution or a centralized directory.

For details, see "Automatic or Manual Provisioning of Certificates" in Chapter 2.

Certificate Types

As shown in Table 3–1, certificate types depend on the user's role in the organization, whether the certificate consumer is a client or server, and the intended applications:

Table 3–1 Certificate Types

Client Certificate	Server Certificate	Description
Authentication	Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.)
Encryption	Encryption	Enables encrypting and decrypting electronic documents and e-mail.
Signing	Signing	Provides electronic documents including e-mail, using S/MIME with verifiable signature and assures non-tampering
Code Signing	Code Signing	Provides verifiable signature for a provider of Java code, Java Script, and other signed files, and assures non-tampering.
NA	CA Signing	Enables requesting subordinate CA certificates

OracleAS Certificate Authority can combine the first three types of certificates to produce certificates that meet multiple needs:

- Authentication and encryption
- Authentication and signing

- Encryption and signing
- Authentication, encryption, and signing

OracleAS Certificate Authority does not limit how many certificates you issue to each user, and it is quite possible to issue multiple certificates. However, a good rule of thumb is to allow each user to possess only one certificate at a time for each task, consistent with the user's role in the organization.

For example, user A might have one certificate for authentication and another certificate for encryption, whereas user B might only have a single certificate for authentication. (You can enforce this using the UniqueCertificateConstraint policy rule. See Chapter 6.) As another example, if you are initially considering an application like secure login to the enterprise portal, you can just use authentication. However, if you foresee that in future, other applications such as secure e-mail are likely to be added, you might want to add signing and encryption certificates as well.

Common Uses for Certificates

In deciding what types of certificate to use, consider the range of applications for certificates:

- Certificates for SSL-enabling servers

  Enabling secure SSL communication for web servers is probably the most common use of certificates. It allows client browsers to validate the identity of a web server, and also securely encrypts the data flow between the browser and web server. In this type of usage, scalability is not a critical issue. All approvals are manual and no integration with a directory service is required.

  Consider the following issues when enabling servers for SSL authentication:

  – How clients will get the server's CA certificate to authenticate the server. Options include making OracleAS Certificate Authority a subCA of a recognized CA, or distributing the CA certificate.

  – How the communicating parties will test for certificate revocation. Browsers and servers need to be aware of the relevant certificate revocation lists.

  For more information, see:

  – CA Trust Hierarchy

  – Installing a CRL into Your Browser

  – Saving the Binary or BASE64 CRL to Disk

- Certificates for authenticating users

  This common application of certificates can take many forms such as:

  – enabling PKI certificates on OracleAS Single Sign-On to log on to enterprise portals

  – using certificates for Virtual Private Network (VPN) access

  To prepare the user community to work with certificates, users must be trained in how to get their own certificates, and how to renew or revoke their certificates as needed. The certificate holder's responsibility, if the private key is lost or compromised, should be emphasized. If smart cards are to be used for key storage, training is essential to ensure that the cards are used properly and secured at all times.

  For details about user tasks relating to certificates, see Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority".

Although end users can typically obtain and maintain their own certificates, the administrator is responsible for overall certificate provisioning and life cycle maintenance - for example, maintaining updated CRLs to prevent misuse, and revoking a user certificate after an employee leaves the organization.

- Certificates for signing e-mail and documents

  Digital signing of documents with PKI certificates provides the twin benefits of data security, and authentication of the document's origin. See Appendix G, "S/MIME with OracleAS Certificate Authority" for details on how to make certificates available to e-mail clients.

- Certificates for encrypting e-mail

  S/MIME applications, such as Outlook or Netscape mail clients, can use certificates to sign and encrypt e-mail messages. With OracleAS Certificate Authority, separate certificates can be used for signing and encryption, or these functions can be fulfilled with the same certificate, as explained earlier.

  At sites that implement message encryption, users must remember to back up their keys properly, because in the event that the encryption keys are lost, the user can no longer decrypt mails.

Certificate Properties

On a day-to-day basis, the OracleAS Certificate Authority administrator examines certificates and certificate requests from end-users (who may be individuals or server entities), renews or revokes certificates, and approves or rejects certificate requests. In order to carry out these duties, the administrator must understand the different certificate configuration options:

- Certificate Naming
- Certificate Key Size
- Certificate Validity Period
- Supported Extensions
- Smart Card Support

> **Note:** Some configuration decisions are made at OracleAS Certificate Authority installation time, and others are made when a certificate is requested. Some changes can only be effected by regenerating the root certificate.

For details about certificate configuration, see Chapter 4, "Introduction to Administration and Certificate Management".

Certificate Naming A distinguished name (DN) is a globally unique identifier representing an individual or other entity's identity, and appears in each certificate. While OracleAS Certificate Authority issues no explicit recommendations for the DN, there are some general rules that should be followed:

> **Note:** Automatic provisioning eliminates the need for this task, since your directory is already deployed.

- The DC and EMAIL components must use only printable, ASCII characters. This is true even if your site uses a multibyte character set.
- The CA's DN must be unique and clearly identify the organization
- The CN for the CA's DN cannot be the server host name
- The organization must always be included in the DN, otherwise browsers may be confused
- For SSL server certificates, the `cn` component of the DN must be the host name (of the CA, web server, and so on)

For users already provisioned in Oracle Internet Directory, this naming has already been assigned as part of the Oracle Internet Directory provisioning. OracleAS Certificate Authority uses the DN information from Oracle Internet Directory for single sign-on users.

> **Note:** The DN is the central factor in configuring certificate policy. For details, see Table 6–9, " Predicate Attributes" and the subsequent discussion in Chapter 6 for guidelines on forming DNs.

Certificate Key Size In general, you should use the strongest key size possible for your certificates (consistent with performance considerations - larger key sizes lower performance):

- Key sizes of 512 bits provide the weakest level of encryption
- Key sizes of 1024 provide stronger encryption and are commonly employed
- Key sizes of 2048 or better are the best choice for high-security needs.

Recommended key sizes follow:

Intended Usage	Recommended Key Size. bytes	Comments
CA Certificate	>=2048	Must be defined at installation time
Server Certificate	>=1024	
User Certificate	>=1024	A minimum key size of 1024 is recommended

Certificate Validity Period Although the end user indicates a validity period when requesting a certificate manually, the administrator can control the certificate's actual validity using the ValidityRule policy rule. With this rule the administrator can specify the minimum validity period (defaults to 90 days), the maximum validity period (defaults to 3650 days), and the default value, the certificate types, and DN's to which the rule applies.

For automatic certificate requests (using OracleAS Single Sign-On Server or SSL authentication), ValidityRule automatically sets the validity period.

> **See Also:** For more information, see "ValidityRule" in Chapter 6.

Supported Extensions Using extensions, an organization can customize a certificate to provide information beyond that which is allowed in the standard certificate fields. There are two types of extensions, *critical* and *non-critical*:

- a certificate-using system (application) must reject the certificate if it encounters a critical extension that it does not recognize, or if the value does not fit the intended use
- a non-critical extension is processed if possible, but the application may be ignored if it is not recognized or does not fit the intended use

OracleAS Certificate Authority complies with the IETF X.509 V3 certificate format and supports several standard certificate extensions, enabling you to configure certificates to fit their intended applications and to provide more information about the subject. By default, OracleAS Certificate Authority supports:

- key usage, automatically configured based on the certificate type selected
- extended key usage, automatically configured based on the certificate type selected
- CRL distribution, allowing the CRL location to be imbedded in the certificate. This extension is used by the relying applications to find the CRL.
- the subject alternative name (subjectAltName) extension, which lets the administrator configure the system so that a predefined alternate name identifier appears in certificates issued to SSO-authenticated users. This extension typically contains an e-mail address or alternate user names, and enables e-mail encryption, signing, or use by other applications.

  You can specify that the extension is mandatory, meaning that the certificate request is denied if the alternative name cannot be found in Oracle Internet Directory.

  Note: The subject alternative name extension is available for manually authenticated users and SSO certificate requests.

  For more information, see:

  - Appendix D, "Extensions"
  - "Subject Alternate Name Extension" in Chapter 5.

Smart Card Support When requesting a certificate, the end user can specify that the certificate is to be stored on a cryptographic token or smart card; the card can be removed for separate storage and increased security. OracleAS Certificate Authority supports popular smart card vendors, using the browser to interact with the hardware.

The user chooses the card from a list; only cards that are actually installed on the user's system will be listed.

Certificate Renewal and Revocation

A certificate currently in use has a validity period and can no longer be used after it expires. The OracleAS Certificate Authority administrator can configure:

- whether renewal is allowed
- the validity period for renewed certificates. For example, the initial certificate could be valid for five years, whereas the renewed certificate might only be valid for two years.

- a time window, prior to and following the expiration date, during which the certificate can be renewed. If this is not specified, the default window is 10 days before or after the expiration date.

Responsibility for certificate renewal depends on the certificate type:

- A certificate that was requested by manual approval (a manual certificate) must be renewed by the administrator.
- A certificate issued using OracleAS Single Sign-On Server or SSL authentication (an automatic certificate) can be renewed by either the certificate owner or the administrator.

See Also: The following sections provide more information about certificate renewal:

- "Renewing Certificates" in Chapter 4
- "ValidityRule" in Chapter 6, used by administrators to configure the validity
- "Certificate Retrieval, Renewal, and Revocation" in Chapter 8

Either the certificate user or the OracleAS Certificate Authority administrator can revoke a certificate. The administrator typically revokes certificates:

- When the certificate owner no longer has the right or need to use the certificate (for example, due to a change in status)
- When the certificate owner's private key (either the signing key or the decryption key) has been compromised

---

See Also: The following sections provide more information about certificate revocation:

- "Revoking Certificates" in Chapter 4
- "Certificate Retrieval, Renewal, and Revocation" in Chapter 8

Distributing the CA Certificate

To work with OracleAS Certificate Authority, users must be able to trust the CA certificate. This is necessary so users can trust the certificates they receive from an issuing certificate authority, or simply to trust the servers whose certificates, in turn, are issued by the certificate authority.

There are several ways to distribute the CA certificate so users can establish this trust relationship with OracleAS Certificate Authority:

1. A user can explicitly install the certificate into the browser. See "Configuring Your Browser to Trust OracleAS Certificate Authority" and "Installing a CA Certificate" in Chapter 8.
2. The CA certificate can be installed in the machine's base image. End users utilizing the installed browser automatically obtain the CA certificate.
3. You can follow any patching and security update push mechanism to push these changes to the client machines.
4. The CA certificate can be pushed out through SMS

5. The entire Windows domain can be made to trust the CA certificate by using a group policy

Define Certificate Policies and Practices

A Certificate Practice Statement (CPS) describes the policies and practices that your certificate authority follows when providing certificate services, including:

- Certificate types and usage
- Management of the certificate life-cycle
- Procedures and policies governing end-users
- Technical specifications for certificates

The range of information contained in a CPS includes, but is not limited to:

- General information
  - legal issues, liabilities and financial obligations
  - Public Key Infrastructure knowledge requirements
- Authentication and identification concerns
  - positive identification of the CA, including CA name, server name, and DNS address
  - how users are authenticated to the CA
  - the intended purpose of the certificate
  - what certificate policies are implemented by the CA and what certificate types are issued
  - policies, procedures, and processes for issuing, renewing, and recovering certificates
- Physical and personnel security
  - physical, network, and procedural security for the CA
  - requirements for certificate enrollment and renewal
  - requirements for certificate users, including what users must do in the event that their private keys are lost or compromised
  - policies for revoking certificates, including conditions for certificate revocation, such as employee termination and misuse of user rights
  - warning and caution notes about using certificates
- Technical security requirements
  - cryptographic algorithms, Cryptographic Service Provider (CSP), and key length used for the CA certificate
  - minimum length for the public key and private key pairs
  - certificate key strengths and related security consequences
  - lifetime of the CA certificate
  - certificate life cycle details
  - standards or protocols used

  - private key management requirements, such as storage on local disk, smart cards, or other hardware devices
- Certificate profile
  - certificate types and usage
  - extensions supported and used, constraints
  - certificate limitations
- CRL policies
  - where to locate CRL distribution points
  - how often CRLs are published

You can add or alter your certificate practice statement by editing the `$ORACLE_HOME/j2ee/oca/applications/ocaapp/oca/helpsets/oca_practice_stmt/ocaadmin_cs_practicestmt.html` file.

After OracleAS Certificate Authority is restarted, your changes will appear on the Practice Statement page when a user clicks the Practice Statement icon which appears on every page.

> **Note:** The Certificate Practice Statement created by the OracleAS Certificate Authority administrator using this procedure is not globalization (i18n) compliant. This means that it is rendered in the locale in which it is edited by the administrator, regardless of the client locale; clients will see it only in the language in which it was created.

Define CRL Policies

Before using a certificate, applications in your trust environment should be able to verify the status of the certificate so that a revoked certificate is not used for authentication. The certificate revocation list (CRL) accomplishes this goal, with OracleAS Certificate Authority periodically generating and storing the list of revoked certificates.

> **Note:** OracleAS Certificate Authority's Certificate Revocation List resides in Oracle Internet Directory with a CRL Distribution Point (CRLDP) extension, and also resides in the database. To prevent a revoked certificate from being misused, applications extract the CRLDP from the certificate and use it to retrieve the CRL from Oracle Internet Directory. End users, on the other hand, point their browser to OracleAS Certificate Authority, which retrieves the CRL from the database.

There are two ways to generate the CRL:

- By default, OracleAS Certificate Authority automatically generates the CRL. The administrator can configure a notification parameter for e-mail notification in the event that CRL generation fails.
- The administrator can also generate an updated CRL manually.

Some recommendations for administering the CRL follow:

- You should identify an appropriate validity period for the CRL, as well as the interval (before and after expiry) in which a new CRL should be generated. This interval should be less than the validity period, and chosen so that applications have some grace period to retrieve the latest CRL. This also allows some time (the range of time between the end of CRL validity and the next CRL generation) to recover from a certificate authority failure before the CRLs become invalid.
- Configure automatic CRL generation. Reserve the use of manual CRL generation for extreme cases, for instance when a high-value certificate must be revoked.

See Also:
- "Updating the Certificate Revocation List (CRL)" in Chapter 4 for details about generating the CRL
- "Retrieving the Certificate Revocation List" in Chapter 4 for details about programmatic access to the CRL using `ldapsearch`.
- "Handling Certificate Revocation Lists (CRLs)" in Chapter 8 to see how to obtain the CRL from the OracleAS Certificate Authority User home page

Define Alerts and Notifications

OracleAS Certificate Authority provides a range of parameters to alert end-users and the administrator when key events occur, and to schedule administrative jobs:

- receive an alert when the number of pending certificate requests exceeds a specified threshold
- schedule a job to automatically generate the CRL at specified intervals
- receive an alert whenever automatic generation of the CRL fails
- schedule a job to synchronize certificate information with Oracle Internet Directory (this is useful in the event that the directory goes down, since OracleAS Certificate Authority can queue up certificates and synchronize with the directory once it comes back up)
- customize the e-mails automatically sent to OracleAS Certificate Authority users to notify them of administrative actions like certificate approval and rejection

For details on this topic, see "Notification Sub-tab" in Chapter 5.

Planning your OracleAS Certificate Authority Architecture

It is important to think about certain system architecture issues before you begin implementing OracleAS Certificate Authority. You should consider what kind of trust hierarchy you need, whether to configure offline CAs, and the security of your OracleAS Certificate Authority installation. Proper planning will enable you to implement a reliable and scalable certificate authority.

This section contains the following topics:

- CA Trust Hierarchy
- Securing the CA

CA Trust Hierarchy

In a CA trust hierarchy, the root certificate authority for a security domain is the basic CA that is ultimately trusted by all users in the organization. The root CA can in turn

issue a certificate to another CA, thus creating a subordinate CA (sometimes known as a Sub CA).

You can install a subordinate CA at any time, create a certificate request and save it to a file, and submit it to the root CA. Once approved, the certificate can then be imported into the subordinate CA for use as a Sub CA signing certificate. Sub CAs can, in turn, issue certificates to lower levels of CAs.

By appropriate use of these techniques, which OracleAS Certificate Authority fully supports, you can set up a trust hierarchy in which end users (or applications acting on behalf of users) can trace the CA certificate chain back to the root CA - the ultimate source of trust for the hierarchy - providing secure, reliable, and efficient digital communication.

Online and Offline CAs

In the simplest scenario, there is a single certificate authority serving the entire organization. This CA serves multiple roles: as the root CA, and also as certificate issuer. Figure 3–1 illustrates this scenario.

Figure 3–1 The Root CA

However, this configuration is not advisable even for relatively small sites; it is preferable to create a hierarchy of CAs to provide important benefits in security, reliability, availability, and cost savings. For example, CAs in the hierarchy can be dedicated to specific roles to protect critical CAs against attack.

Figure 3–2 demonstrates how the root CA can be protected. It shows a simple certificate authority hierarchy serving an organization which consists of two geographically distinct sites. Each site is served by a SubCA. The two SubCAs are subordinate to the root CA, which is offline. The SubCAs are both online, and service certificate requests for the end users at their respective sites.

Figure 3–2 A Simple CA Hierarchy

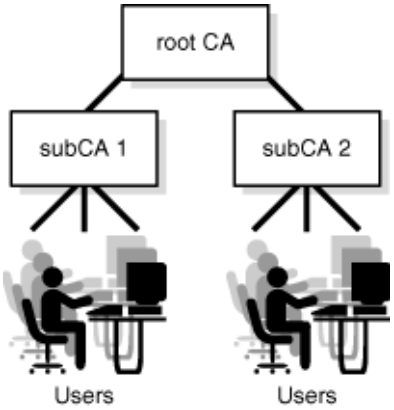
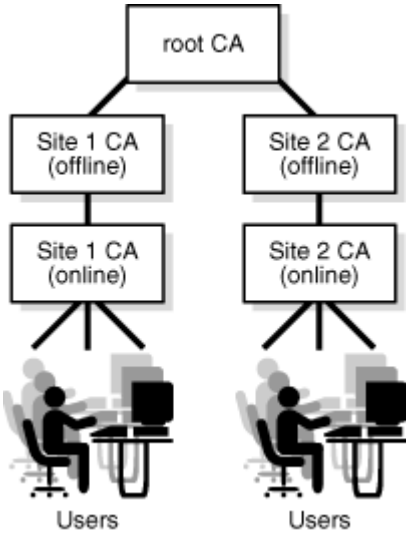


Figure 3–3 shows an extension of the previous certificate authority hierarchy for the same organization. Each site is now served by a pair of Sub CAs; one Sub CA is online and functions as the issuing certificate authority, while the other, higher Sub CA is offline for security. Both pairs of Sub CAs are, in turn, subordinate to the root CA, which itself is offline.

With this arrangement, if the issuing Sub CA is somehow compromised, the offline Sub CA higher up in the hierarchy can revoke its CA certificate and create a new Sub CA to replace it.

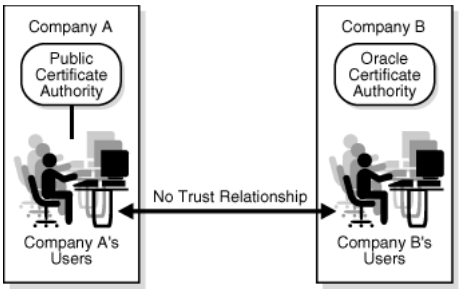
Figure 3–3 A CA Hierarchy with Two Levels of SubCA



Another advantage of a trust hierarchy is the ability to set up trust relationships across companies or organizations that are served by independent certificate authorities.

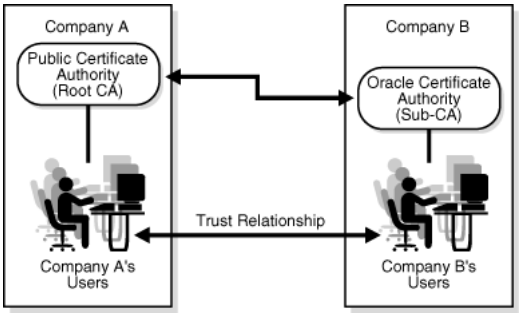
In Figure 3–4, companies A and B have separate CAs; company A uses a third-party CA while company B uses OracleAS Certificate Authority. In this scenario there is no trust relationship between the organizations since the CAs are independent.

Figure 3–4 Organizations with no trust relationship



In Figure 3–5, the two companies have entered into a trust relationship by establishing company B's certificate authority, OracleAS Certificate Authority, as a Sub CA to company A's third-party CA. Users in the two companies can now communicate securely based on the trust hierarchy that has been set up between the respective CAs.

Figure 3–5 Organizations related by a trust hierarchy



Depending on your security and availability requirements, you will need to decide upon a trust hierarchy that suits your needs. For configuration details, refer to "Configuring an OracleAS Certificate Authority Instance to Be a Subordinate CA of Another CA" in Appendix B.

Securing the CA

Securing the entire CA installation, and particularly the root CA, is of the utmost importance. Given the effort involved in redeploying CAs and reissuing certificates, a compromised root CA can prove much more costly than a compromised intermediate CA or issuing CA.

To protect the root CA, it is recommended that you:

- install OracleAS Certificate Authority on a dedicated machine
- secure it against intruder attack by maintaining the server in a physically secure location
- limit access to trusted administrators
- consider using hardware storage for administrator keys
- offline the root CA (and subCAs, whenever possible) for additional protection. See "Online and Offline CAs" for more information.
- set up an online Sub CA for day-to-day operations like issuing certificates
- follow standard guidelines for hardening the host and for the Oracle Application Server installation

- remove unnecessary services, and limit users who have access to the host machine

See Figure 3–3, "A CA Hierarchy with Two Levels of SubCA" and the related discussion for details.

Deployment Considerations and Base Scenarios

Depending on your site's requirements, you can choose from among several different deployment strategies for OracleAS Certificate Authority. This section describes the different scenarios and contains the following topics:

- Required Components for OracleAS Certificate Authority
- Default Deployment
- Production Deployment
- DMZ Deployment
- High Availability Deployment Options

Required Components for OracleAS Certificate Authority

OracleAS Certificate Authority needs the following components for its operation:

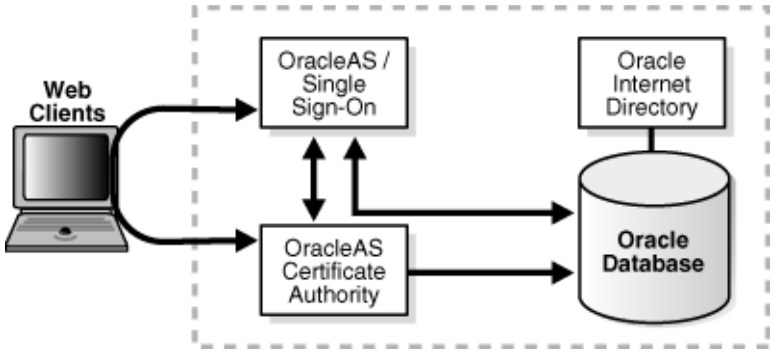
- Oracle HTTP Server (OHS) (must be on the same machine as OracleAS Certificate Authority)
- OC4J for OracleAS Certificate Authority (must be on the same machine as OracleAS Certificate Authority)
- Infrastructure metadata repository
- Oracle Internet Directory
- Single Sign-On (optional)

Note: OCA is not automatically selected for installation - that is, as a default selected choice. To install OCA, you must explicitly select it for installation.

Default Deployment

In the *default* deployment, all the required components are installed on the same machine and in the same Oracle home, as shown in Figure 3–6. This configuration is ideal for development and non-production environments, and is the default configuration if you do not make other choices during installation.

Figure 3–6 Oracle Application Server Certificate Authority Default Installation

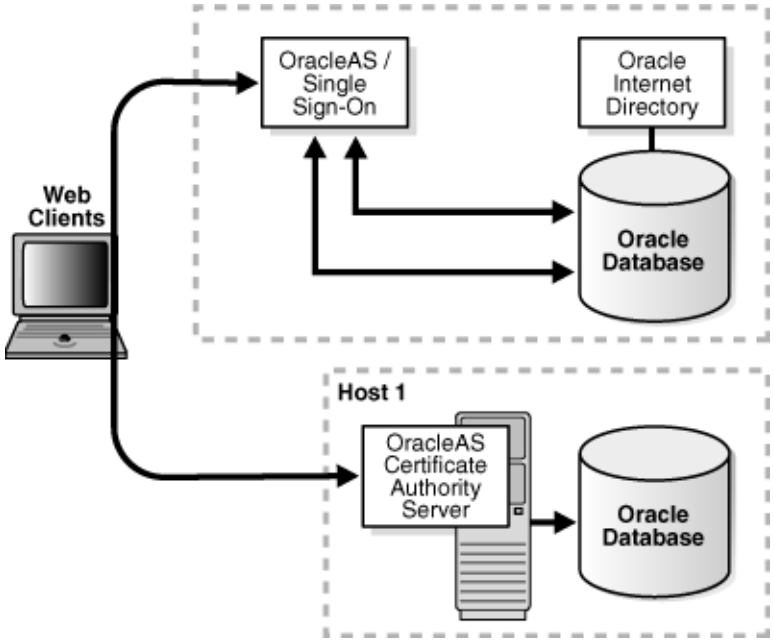


The installation instructions for this default deployment of OracleAS Certificate Authority appear in Section 6.19 of the Oracle Application Server Installation Guide.

Production Deployment

In the *recommended* production deployment, OHS, OC4J, OracleAS Certificate Authority, and the infrastructure metadata repository reside on one machine, in one Oracle home. Remaining components like OracleAS Single Sign-On and Oracle Internet Directory are on a different machine, in a different Oracle home. This physical separation makes it possible to harden the security of that separate location, to protect OracleAS Certificate Authority in a very secure location. Since OracleAS Certificate Authority is at the top of the trust chain for certificates, these additional protections are prudent in a production environment, as illustrated in Figure 3–7. Similarly, it is better for OracleAS Certificate Authority security reasons not to use remote administration mechanisms to start or stop these components.

Figure 3–7 OracleAS Certificate Authority Recommended Production Installation

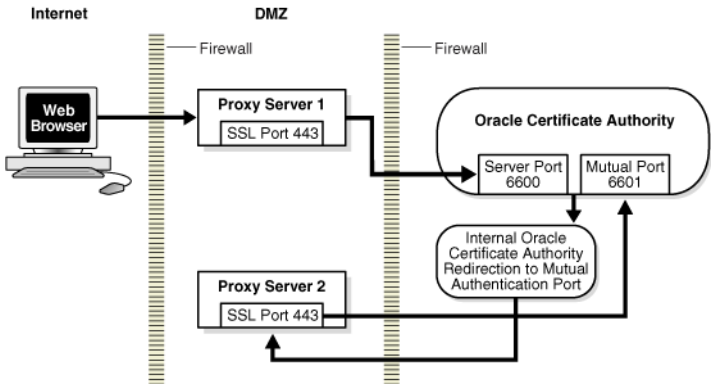


Installation instructions for this recommended deployment appear in Section 6.20 of the *Oracle Application Server Installation Guide*.

DMZ Deployment

A DMZ configuration exposes OracleAS Certificate Authority to an external network and may be necessary if, for example, you wish to allow access to the certificate authority by internet users.

Figure 3–8 OracleAS Certificate Authority DMZ Installation



In this scenario, which is shown in Figure 3–8, you configure proxy servers in the DMZ to handle OracleAS Certificate Authority requests.

OracleAS Certificate Authority requires two ports for its functioning. External users access OracleAS Certificate Authority on either port 443 (SSL users - this is the recommended setup) or port 80 (non-SSL users). All requests are routed through proxy server 1 utilizing, for example, port 6600 (or another port in the available range) for server authentication. If - and only if - it is determined that a request requires mutual authentication, OracleAS Certificate Authority internally redirects the request to proxy server 2, which then routes it to mutual authentication utilizing, for example, port 6601.

If only port 443 is allowed in the proxy server, then two proxy servers should be used. Also, the proxy server information needs to be updated in the OracleAS Certificate Authority repository.

For details about setting up this scenario, see Appendix F, "External Access to Protected OracleAS Certificate Authority".

Note: This is not a recommended topology for OracleAS Certificate Authority deployment. However, if it becomes necessary to deploy in the DMZ, it is recommended that you:

1. have a SubCA for external users
2. allow only manual authentication to OracleAS Certificate Authority, and have the administrator issue these certificates

---

High Availability Deployment Options

Broadly speaking, there are two types of high availability options for protecting a certificate authority deployment:

- problems such as human errors, machine and media failures can be resolved with localized, high availability protection at a single data center
- natural disasters and regional network outage can be addressed with geographically distributed disaster recovery deployments

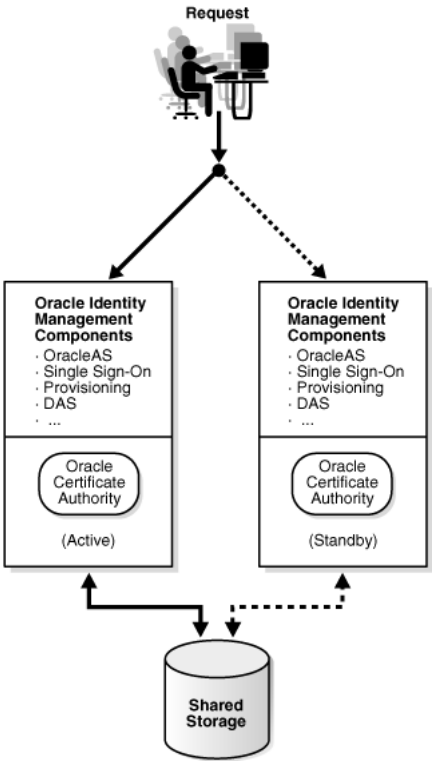
This section presents three high availability options for your OracleAS Certificate Authority deployment: a cold failover cluster for localized protection, a disaster recovery solution, and a combination of these to protect against both types of failures.

> **See Also:** The deployment options shown here are described in depth in the *Oracle Application Server High Availability Guide*.

Cold Failover Cluster

Cold failover cluster, which is shown in Figure 3–9, operates in an active-passive mode to provide localized high availability. The active node runs an Oracle Application Server instance including OracleAS Certificate Authority as well as other Oracle Identity Management components. The passive node, which is not started unless and until the active node goes down, likewise hosts an Oracle Application Server instance consisting of OracleAS Certificate Authority and related components. In this cluster environment, the *ORACLE_HOME* for the OracleAS Infrastructure resides on a shared storage system.

Figure 3–9 Cold Failover Cluster



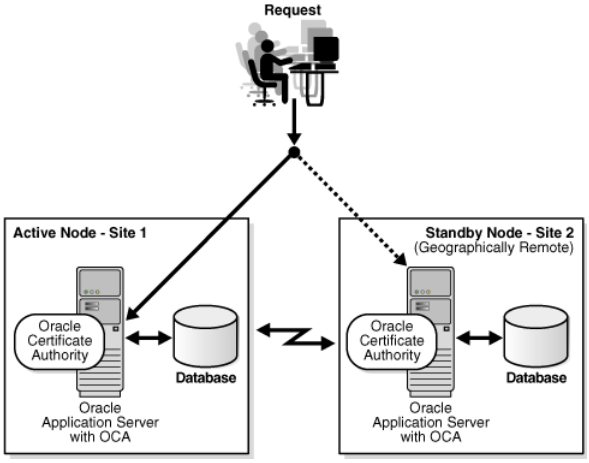
As the name implies, the active node actively services certificate requests, while the passive node is inactive until there is a component failure in the active node; when this occurs, the user requests are redirected to the secondary node, which mounts the file system and assumes control over the shared storage to continue processing.

> **Note:** The virtual host provides a network-addressable host name that maps to one or more physical machines through a load balancer or a hardware cluster. The virtual host can be standalone or it can reside within the OracleAS Infrastructure.

Disaster Recovery

Figure 3–10 shows a disaster recovery configuration utilizing active and standby nodes which are synchronized but are geographically dispersed. Each node hosts an Oracle Application Server instance, including OracleAS Certificate Authority as well as other Oracle Identity Management components, each instance residing on its own `ORACLE_HOME`. In this scenario, the active site is the production site and actively services client requests, while the standby site mirrors the applications (including OracleAS Certificate Authority) and data repository of the production site. Site synchronization is provided by Oracle Data Guard, with OracleAS Guard providing the procedures necessary for backing up and restoring the necessary configuration files.

Figure 3–10 Disaster Recovery



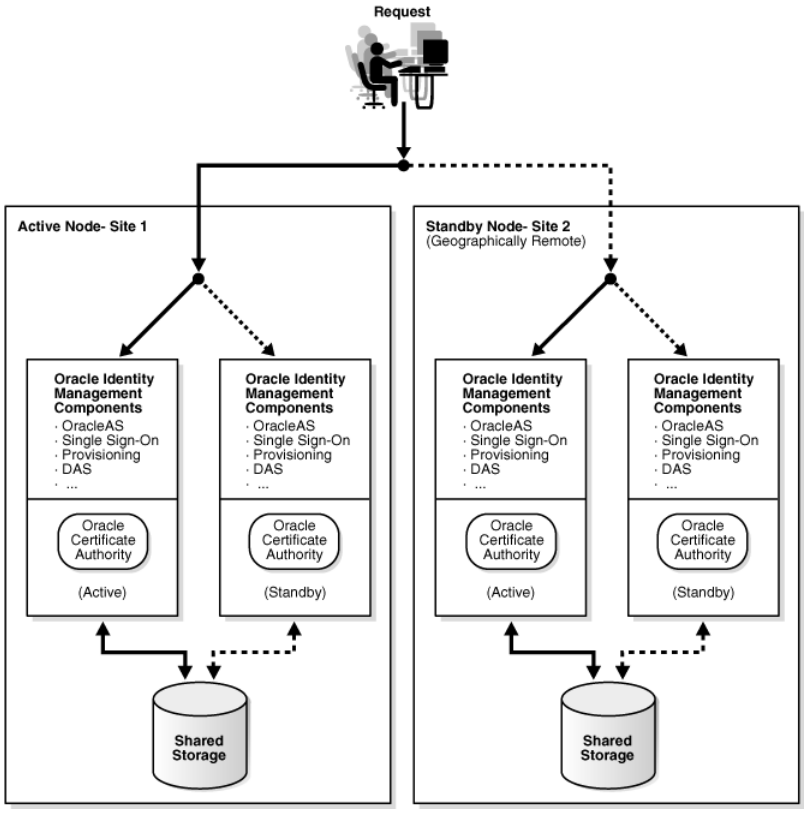
In the event of catastrophic failure at the active site, OracleAS Guard performs failover tasks to switch the standby site over to active mode, and user requests are redirected to that node for continued operation.

Cold Failover Cluster and Disaster Recovery

This configuration combines the "Cold Failover Cluster" and "Disaster Recovery" solutions into a unified system that provides protection from isolated problems as well as from catastrophic failures affecting an entire site.

In this arrangement, shown in Figure 3–11, the active and standby sites each provide independent cold failover cluster services. The site-specific virtual hosts can handle local failures entirely within the site by redirecting requests to the passive Oracle Application Server instance. The overall "global" virtual host, on the other hand, provides a higher level of load balancing by failing over requests to the standby site in the event of a catastrophic failure.

Figure 3–11 Combined Cold Failover Cluster and Disaster Recovery



OracleAS Certificate Authority Implementation and Use Case

This section brings together the logical and physical concepts underlying certificate authority implementation to help you define the characteristics and decision points critical for your OracleAS Certificate Authority deployment.

Implementation Checklist

The following checklist summarizes the key planning items for a deployment of OracleAS Certificate Authority and provides the essential starting point for deployment.

Table 3–2 Implementation Checklist

Planning Item	Recommended / Proposed Value	Notes
Certificate policies and implementation details		
CA DN		
CA certificate lifetime		See Table 4–4 for default value
CA wallet type		Enter *signing*, *SSL*, or *S/MIME*
CA wallet key size		See Table 4–4 for default value

Table 3–2 (Cont.) Implementation Checklist

Planning Item	Recommended / Proposed Value	Notes
CA certificate validity period		See Table 4–4 for default value
SubCA certificate validity period		
Total number of users		
Number of authentication certificates		
Number of encryption certificates		
Number of signing certificates		
Number of code signing certificates		
Validity period for new certificates		See "User Certificates Tab" in Chapter 8 for defaults and other details.
Validity period for renewed certificates		See Table 6–7 for defaults.
Renewal window		Enter number of days before and after expiration that certificates can be renewed. See Table 6–7 for defaults.
RSA public/private key lengths		See "User Certificates Tab" in Chapter 8 for defaults and other details.
Allow multiple certificates for same usage to same name?		This is the UniqueCertificateConstraint policy
CRL distribution point		For details, see: ▪ Updating the Certificate Revocation List (CRL) ▪ Handling Certificate Revocation Lists (CRLs)
CRL publication frequency		
How will certificates be stored?		Enter storage requirement, such as file or token
Architecture		
Number of levels in CA hierarchy		
Role of root CA		Enter online/offline, whether issues certificates
Sub CA type		Enter *online* or *offline*
Additional SubCA data		
Deployment		

Table 3–2 (Cont.) Implementation Checklist

Planning Item	Recommended / Proposed Value	Notes
Separate repository for OracleAS Certificate Authority?		
DMZ Deployment?		
High Availability		
Cold Failover?		
Disaster Recovery?		

Use Case: MyPKIsite.com

This use case describes a fictional deployment scenario, MyPKIsite.com, to demonstrate PKI-enabled enterprise deployment using OracleAS Portal and OracleAS Certificate Authority. Rather than provide an exhaustive listing of tasks, it identifies the major factors that should be taken into account when deploying OracleAS Certificate Authority.

Scenario

MyPKIsite.com is a fictional online travel service which enables travelers to plan itineraries, make reservations, and securely exchange messages with travel advisors on two continents. The company needs to secure its website using PKI certificates.

> **Note:** MyPKIsite.com is a fictitious company, used only for the purposes of illustration.

Here are some basic facts about the company and its security needs:

- The company has offices in the US and UK
- In addition to employees at both locations, the site must support external users, including customers and partners
- The company's security requirements include
  - enabling SSL authentication for servers
  - the ability to implement single sign-on for a number of applications, using client certificates for authentication
  - the ability to sign and encrypt e-mail messages with S/MIME

Administrative Roles

Based on current needs, it is decided to appoint one person to serve as both the OracleAS Certificate Authority administrator and the web administrator.

CA Hierarchy for MyPKIsite.com

The implementation team decides to implement a basic two-level hierarchy:

1. Offline root CA
2. Four issuing SubCAs:
   - Two issuing SubCAs in the US, one for internal users and another for external users. These issuing SubCAs are subordinate to the root CA

- Two issuing SubCAs in the UK, one for internal users and another for external users. These issuing SubCAs are subordinate to the root CA.

Alternative Hierarchy Option

During planning, the team had also considered an alternative option consisting of a three-level hierarchy:

1. Offline root CA
2. Offline SubCA for the US, and another offline SubCA for the UK
3. Four issuing SubCAs:
   - Two issuing SubCAs in the US, one for internal users and another for external users. These issuing SubCAs are subordinate to the offline US SubCA.
   - Two issuing SubCAs in the UK, one for internal users and another for external users. These issuing SubCAs are subordinate to the offline UK SubCA.

If security warrants the need for additional protection at a later time, they can later move to this three-level hierarchy by adding another SubCA layer consistent with this option.

User Entries in Oracle Internet Directory

User entries, which will contain users' certificates, must be added in Oracle Internet Directory. Given the large number of users who require certificates, the team decides to use Oracle Internet Directory's bulk loading tool to define and maintain user entries in the directory.

`bulkload` loads and appends large numbers of entries to Oracle Internet Directory through LDIF files. `bulkload` handles various steps of the procedure, including:

- Checking the input files for schema and data consistency
- Converting the LDIF data into a format appropriate for the loader
- Loading and indexing the data

By way of reference, there are other tools the team could have utilized for creating user entries:

- Delegated Administration Services
- Command-line tools
- Oracle Directory Manager

Component Instances

The site will make use of shared Oracle Internet Directory and Oracle Application Server Single Sign-On components.

> **Note:** Multiple *physical* instances of Oracle Internet Directory, one in the US and one in UK, can be set up in multi-master mode.

The team must install a total of five instances of OracleAS Certificate Authority, including one root CA and four issuing SubCAs. Each instance has its own metadata repository, on its own machine.

Certificate Requirements for MyPKIsite.com

The team makes the following decisions about certificate provisioning and usage:

Provisioning

End users will be authenticated to the site using single sign-on certificates. Manual certificates will be issued for servers.

> **See Also:** For more information, see "Certificate Provisioning".

Certificate Lifetime

Certificates issued by MyPKIsite.com are valid for the following periods:

Table 3–3 Certificate Lifetimes

Certificate	Initial Validity	Renewal Window	Renewed For
User	Two years	30 days before/after expiration	One year
Server	Ten years	30 days before/after expiration	Five years

Once certificates expire, they will not be renewed. (However, company practice allows certificates to be renewed up to 30 days after expiration. To implement this, the administrator can set a parameter which provides a "grace period" for renewal. Regardless, certificates cannot be renewed if they have been expired longer than 30 days.)

Certificate Key Sizes

MyPKIsite.com will use these key sizes:

- 1024 bytes for user certificates
- 1024 bytes for server certificates
- 2048 bytes for CA certificates

Certificate Usage

The site will use two types of certificates:

- One certificate for authentication and signing
- Another certificate for encryption

The unique certificate constraint will be imposed, so that a user can only have one authentication and signing certificate, and one encryption certificate.

> **See Also:** See "Certificate Types" for more information.

Certificate Distribution

The team considers two options for distributing the root CA and SubCA certificates:

1. The user imports these certificates manually
2. The certificates reside in the base image of each user's machine

The team decides that users should import their certificates manually so that base images do not need to be rebuilt. Users will go to the OracleAS Certificate Authority home page to download and input the CA certificate into their browsers.

Certificate Storage

MyPKIsite.com users have smart cards on their desktop machines. Users will store their private keys on the smart cards.

Revocation

Certificates will be revoked for the standard revocation reasons:

- Key Compromise
- Affiliation Change
- CA Compromise
- Certificate Hold
- Cessation of Operation
- Remove From CRL
- Superseded

CRL Location and Publication

The CRL is to be published every three days at midnight, and the CRL validity period is set at 7 days.

The root CA's CRL is valid for 6 months.

> **See Also:** See "Updating the Certificate Revocation List (CRL)" in Chapter 4 for more information.

Security Considerations

The team decides to take the following measures for securing the certificate authority:

- Install the root OracleAS Certificate Authority instance, and its associated metadata repository, on a dedicated machine
- Maintain the CA server in a physically secure location
- Offline all non-issuing CAs, as outlined in "CA Hierarchy for MyPKIsite.com".

High Availability Considerations

Since they will handle all user requests, the four certificate issuing SubCAs must be highly available. They will be supported with a combination of Cold Failover Cluster and Disaster Recovery using Dataguard.

All CA instances, offline as well as online, need to be backed up on a regular schedule.

Detailed Implementation Checklist for MyPKIsite.com

MyPKIsite.com's implementation team has now finalized decisions about the design and operation of their OracleAS Certificate Authority installation. The checklist in Table 3–4 summarizes these decisions, and provides links which explain how the various setup tasks are performed.

Although derived from the checklist in Table 3–2, this list is more detailed and refers to specific OracleAS Certificate Authority tasks and functions.

> **Note:** Some of the items in the list, such as user certificate requests, are not immediate tasks but rather future activities. They are listed here for completeness.

Table 3–4 Implementation Checklist for MyPKIsite.com

Task/Item	Value	References
Install components	-	Platform-specific *Oracle Application Server Installation Guide*
Create user entries	-	*Oracle Internet Directory Administrator's Guide*
Set up SubCAs	4 SubCA instances	Appendix B, "Setting up a CA Hierarchy", particularly "Configuring an OracleAS Certificate Authority Instance to Be a Subordinate CA of Another CA".
Set up passwords	-	"Changing Privileged Passwords" in Appendix A
Request administrator certificate	-	"Requesting the Administrator Certificate" in Chapter 4
Request server certificates	-	"Server/SubCA Certificates Tab" in Chapter 8
Request user certificates (ongoing)	-	■ "End-User Tabs and Processes" in Chapter 8 ■ "Certificate Request Form" in Appendix I
Revoke expired certificates?	Yes, allowing for grace period (see "Set certificate renewal window")	"RevocationConstraints" in Chapter 6
Set certificate renewal window	30 days before/after expiration	"RenewalRequestConstraint" in Chapter 6
User certificate unique?	Yes	"UniqueCertificateConstraint" in Chapter 6
Set certificate validity	Ten year server certificate Two year user certificate	"ValidityRule" in Chapter 6
Set certificate key sizes	2048 (CA) 1024 (server, user)	"RSAKeyConstraints" in Chapter 6
Set key storage mechanism	smart cards	-
Define e-mail parameters	-	"Mail Details" and "Email Templates" in Chapter 5
Set up notifications	daily	"Alerts" in Chapter 5
Configure CRL generation frequency	every seven days	"Scheduled Jobs" in Chapter 5
High Availability		"OracleAS Certificate Authority and High-Availability Features" in Chapter 7
Write CPS		■ "Define CRL Policies" ■ "Define Certificate Policies and Practices"

4

Introduction to Administration and Certificate Management

The Oracle Application Server Certificate Authority web administrative interface covers the following three broad areas, each accessible from a tab on the home page:

- Managing certificate issues: requests for certificate issuance, revocation, or renewal; certificates already issued; and certificate revocation lists (CRLs)
- Managing configuration issues: parameters for OracleAS Certificate Authority actions and for implementation of certificate security policies
- Viewing logs of OracleAS Certificate Authority activity

This chapter describes the first of those three areas: certificate management. The other two are described in Chapter 5, "Configuring Oracle Application Server Certificate Authority".

Some administrative operations require the command-line interface described in Appendix A, "Command-Line Administration". Two of these operations are starting and stopping OracleAS Certificate Authority, as explained in later sections, along with requesting or replacing the administrator's certificate.

For end-user interactions with OracleAS Certificate Authority, a separate web interface presents forms enabling personal certificate-related operations: see Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority".

The present chapter contains the following sections:

Starting and Stopping Oracle Application Server Certificate Authority

For security reasons, OracleAS Certificate Authority's start and stop operations can only be done using the command-line tool `ocactl`, which requires the administrator's password. An example of using these operations appears in Replacing the

Administrator Certificate. This tool is fully described in Appendix A, "Command-Line Administration".

Before OracleAS Certificate Authority can be started, the following five components must be operating or available:

- Infrastructure metadata repository
- Oracle Internet Directory
- and optionally OracleAS Single Sign-On
- Oracle HTTP Server (OHS)
- OC4J for OracleAS Certificate Authority

If OracleAS Certificate Authority is installed in a different `$ORACLE_HOME` from the other infrastructure components, then OHS and OracleAS Certificate Authority's OC4J must be started separately, after the repository. Use this command in OracleAS Certificate Authority's `$ORACLE_HOME`:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

If a single `$ORACLE_HOME` contains all the infrastructure components, including OracleAS Certificate Authority, then OHS and OC4J will already have been started, as in Section 4.3 earlier.

To start, stop, or restart OracleAS Certificate Authority, enter the corresponding command on the command line:

1. To stop OracleAS Certificate Authority, use this command:

   ```
   $ORACLE_HOME/oca/bin/ocactl stop
   ```

2. To start OracleAS Certificate Authority, use this command:

   ```
   $ORACLE_HOME/oca/bin/ocactl start
   ```

3. To restart OracleAS Certificate Authority, use the stop command listed in Step 1, followed by the start command listed in Step 2.
4. To get the status of Oracle Application Server Certificate Authority, use this command:

   ```
   $ORACLE_HOME/oca/bin/ocactl status
   ```

Requesting the Administrator Certificate

You must have the administrator certificate before you can use any of the Oracle Application Server Certificate Authority administrative options and controls in the web interface. If you have the administrator password created during installation, this certificate is easy to get, and is the first step you must do before any other task.

In other systems, requesting, acquiring, and installing your administrator PKI certificate required a whole set of command-line, floppy disk, and cut-and-paste operations.

With OracleAS Certificate Authority, however, the process is simple and easy:

To request the administrator certificate for your authentication, you simply fill in and submit a brief form that appears after OracleAS Certificate Authority is started for the first time. You must be accessing OracleAS Certificate Authority from the computer you intend to use as the administrator. Clicking the Certificate Management tab displays a Welcome page, followed by a form requesting your identifying data.

The form requires your common name, organization, and the OracleAS Certificate Authority administrator password created during installation. You can also supply other DN information: your email address, organizational unit, locality, state, and country.

You can select the certificate key size (default: 2048) and the validity period (default: 1 year).

When the administrator certificate is issued, you install it into your browser. With this certificate in your browser, you can access the OracleAS Certificate Authority facilities in the administration and configuration interfaces to manage certificate requests, certificate revocation or renewal, and policies.

This simple process — easy installation after filling in a simple request-form — replaces all the operations formerly required (before OracleAS Certificate Authority) for PKI certificate acquisition and use.

To request your certificate, perform the following steps:

1. Access the OracleAS Certificate Authority administration interface.

   Launch your web browser and enter the URL and port number of the administration server as they were displayed at the end of installation. For example:

   ```
   https://Oracle_HTTP_host:ssl_port/oca/admin
   ```

   where `Oracle_HTTP_host` is the host on which OracleAS Certificate Authority is installed, and `ssl_port` is listed in `$ORACLE_HOME/install/portlist.ini` under "Oracle Certificate Authority SSL Server Authentication port". For Windows, the path is `%ORACLE_HOME\install\portlist.ini`.

   > **Note:** If port changes have occurred since installation, then the most current information is not in `portlist.ini`. Instead, sign on to the Oracle Enterprise Manager Control and click the instance on which OracleAS Certificate Authority was installed. Then click the Ports link, find the entry in the Type column that says "OracleAS Certificate Authority Server Authentication (SSL)", and use the number in the adjacent column, headed "Port In Use".

   The screen displays a welcome page. Clicking the link provided there displays the form to request the administrator certificate.

2. Enter into that form the DN, password, and certificate information to request your certificate:

   - **DN Information:** Enter the data for the distinguished name (DN) that will identify the administrator as the certified owner of the certificate.

Table 4–1 DN Information for the Administrator's Certificate

Field Name	Information to Enter
Common name	The name that you want on the certificate
Email address	Email address of the administrator
Organization unit	Name of the organization unit or division to which the administrator belongs
Organization	Name of the company or organization to which the administrator belongs

Table 4–1 (Cont.) DN Information for the Administrator's Certificate

Field Name	Information to Enter
Location	The city location of the administrator
State	The state or province of the administrator
Country	Two-letter code for the administrator's country

Note:

For a DN, the DC and EMAIL components must use only printable (ASCII) characters.

This restriction means that even in a locale that uses a multibyte character set, the DC and EMAIL components for Distinguished Names must still use ASCII characters.

- **Certificate Authority Administrator Password:** Only the OracleAS Certificate Authority administrator can do certificate and configuration management. This person is initially authenticated by entering here the password as entered during OracleAS Certificate Authority installation, in the screen named "Specify OCA Administrator Password".

  Passwords must

  – Begin with an alphabetic character from your database character set
  – Be at least eight characters long
  – Contain at least one alphabetic character and at least one non-alphabetic character, that is, a numeric or special character
  – Use only characters in the ASCII character set
  – Be different from all Oracle reserved words; and
  – Contain only alphanumeric characters from your database character set. If needed, the underscore (_), dollar sign ($), or pound sign (#) can also be used, although Oracle Corporation strongly discourages you from using the characters $ and #.

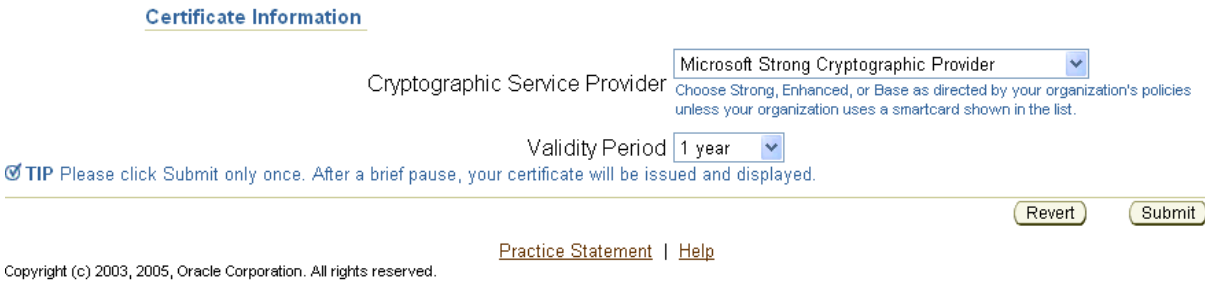
  Thus during installation, the password you choose for the OracleAS Certificate Authority administrator must accommodate these restrictions.

  If your database will be using Oracle's password complexity verification routine (specified using the PL/SQL script UTLPWDMG.SQL), then the password must also meet the following requirements (or additional requirements that you add to that script):

  – Be at least four characters long
  – Differ from the username
  – Have at least one alpha, one numeric, and one punctuation mark character
  – Be different from simple or obvious words, such as welcome, account, database, or user
  – Subsequent changes to this password must also differ from the previous password by at least 3 characters.

- **Certificate Information:** The two vital data for creating a new certificate are the size of its keys and the period of its validity (or its expiration date). In this section of the form, you choose these parameters.
  * In **Netscape**, the phrase **Certificate Key Size** appears, referring to the size in bits of the key-pair to be generated: 512-bit, 1024-bit, or 2048-bit. Choose the size appropriate to your site: 2048-bit is OracleAS Certificate Authority's default, providing excellent security. Higher numbers improve the security at some price in performance.
  * In **Internet Explorer,** the phrase **Cryptographic Service Provider** appears, referring to a choice of providers for cryptography service. Standard choices include key sizes of 512-bit (Microsoft Basic Crypto Provider), 1024-bit (Microsoft Enhanced Crypto Provider), and 2048-bit (Microsoft Strong Cryptographic Provider). OracleAS Certificate Authority's default is the "Strong" choice, if available, followed by Enhanced, if available, and then by Basic. Other choices may also be present, such as Gemplus for smartcard usage. Select the size according to your requirements.

    This section of the form will look like this:



    OracleAS Certificate Authority recommends using Microsoft Strong Cryptographic Provider for the Administrator Certificate. However, if readers for smartcards like Gemplus or Schlumberger are available, they should be used; if no reader is installed, selecting smartcard suppliers causes an error.

- **Validity Period:** The duration of the certificate's validity. The standard default of 1 year is shown, but you can choose your desired period.

3. If you need to start over, click the **Reset** button.
4. To send your request for the Administrator certificate, click **Submit**. (You may have to supply your browser security password.)
5. Follow the instructions that your browser presents as it generates a key-pair. This process can take a few minutes, depending on keysize chosen and processor/memory limitations.
6. Click **Install in Browser**. (You may have to supply your browser security password.)

   Now you have a client authentication certificate in the common name you specified.

   At this point, you can perform any of the tasks available through the web interface of OracleAS Certificate Authority, as described in Chapter 5, "Configuring Oracle Application Server Certificate Authority".

7. Click **Administration Home** to access the welcome page for OracleAS Certificate Authority.

Replacing the Administrator Certificate

You may in future need to replace the administrator's certificate. Reasons could include the password to your private key being lost, the private key somehow being compromised or stolen, or the administrator role being given to someone new.

To replace the administrator certificate, you must stop the server, revoke the current administrator's certificate, and restart the server. These tasks are performed by using the command-line tool `ocactl`, which requires the OracleAS Certificate Authority Administrator password. For security reasons, these commands are only enabled on the command line and not through the graphical user interface (GUI).

The administrator then navigates to the Oracle Application Server Certificate Authority web page and fills in the form presented for Web Administrator Enrollment, as described earlier in "Requesting the Administrator Certificate".

Here are the three relevant command-line tasks:

1. To stop the OracleAS Certificate Authority server, enter the following command on the command line:

   ```
   $ORACLE_HOME/oca/bin/ocactl stop
   ```

2. To revoke the administrator's certificate, enter the following command:

   ```
   $ORACLE_HOME/oca/bin/ocactl revokecert -type WEBADMIN -reason REASON_CODE
   ```

   Note: You may choose any one of the following reason codes (separated by |):

   ```
   {KEY_COMPROMISE | CA_COMPROMISE | AFFILIATION_CHANGE | SUPERSEDED | CESSATION_
   OF_OPERATION | CERTIFICATE_HOLD | REMOVE_FROM_CRL | UNSPECIFIED}
   ```

3. You may want to change the administrative password as well. See "Changing Privileged Passwords" in Appendix A, "Command-Line Administration".

4. On the command line, start OracleAS Certificate Authority services by entering one of the following commands:

   For UNIX, enter `$ORACLE_HOME/oca/bin/ocactl start`

   For Windows, enter `%ORACLE_HOME%\oca\bin\ocactl start`.

At this point, follow the instructions at "Requesting the Administrator Certificate" to obtain that certificate, enabling all administrative capabilities.

Overview of the OracleAS Certificate Authority Administration Interface

To perform administrative tasks you must have a valid administrator certificate. If your initial sign-in is as a regular user, rather than as administrator, you may get the error message described in Appendix C, "Troubleshooting OracleAS Certificate Authority", in section "Prerequisite Issues and Warnings", item "Key Pair Generation Fails during Certificate Requests on Windows"

To access the OracleAS Certificate Authority administration interface, launch your web browser. Enter the URL and port number of the administration server as they were displayed at the end of installation:

```
https://Oracle_HTTP_host:ssl_port/oca/admin
```

For information about the host and port number in the URL, see "Requesting the Administrator Certificate", Step 1.

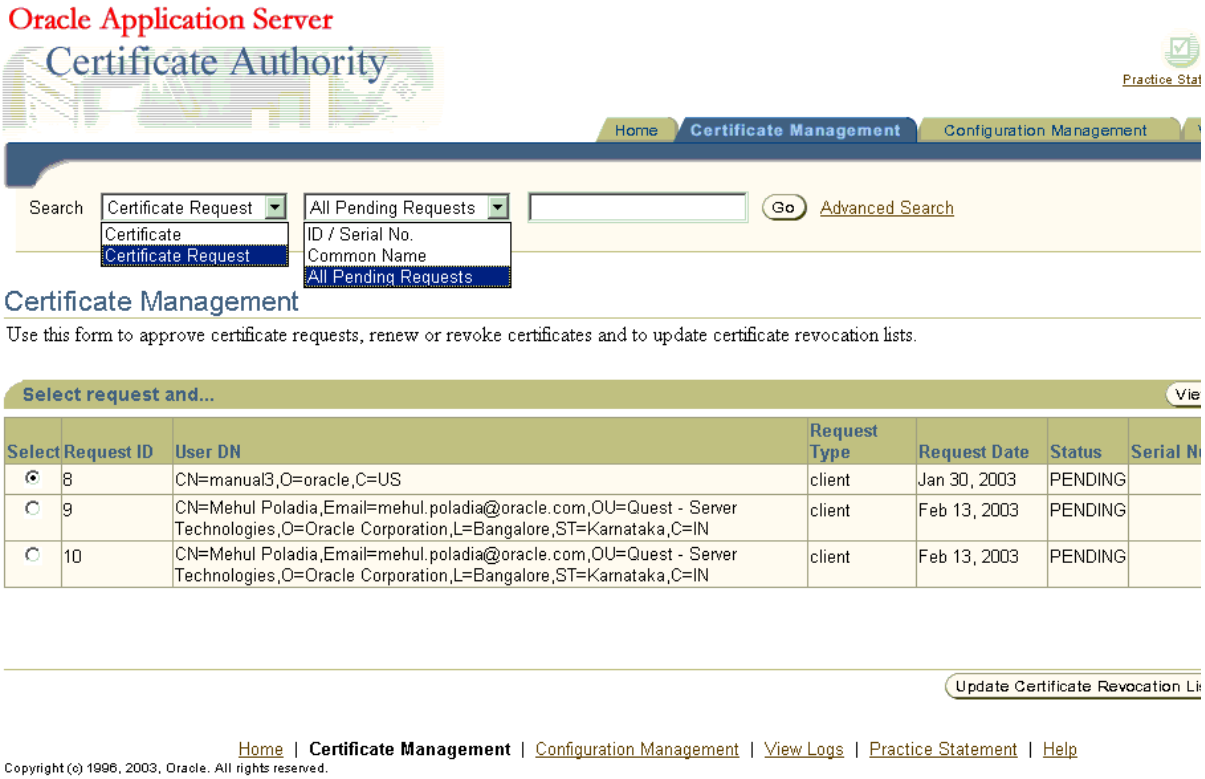
After issuing the command to start OracleAS Certificate Authority, the OracleAS Certificate Authority home page appears, presenting three additional subtabs, as the following figure shows:

These three subtabs enable you to address specific tasks in managing certificates or the Certificate Authority configuration:

- "Certificate Management Tab", described in this chapter
- "Configuration Management Tab", described in Chapter 5, "Configuring Oracle Application Server Certificate Authority"
- "View Logs Tab", described in Chapter 5

Certificate Management Tab

The Certificate Management tab shows all the pending certificate requests, displaying a page that looks like the following:



This page enables the administrator to choose among the following tasks:

Managing Certificates

Oracle Application Server Certificate Authority maintains a master list of all certificate requests and their current status: pending, rejected, or certified. Upon entering the Certificate Management tab, all certificate requests needing action (pending) are displayed. The administrator is responsible for approving or rejecting such requests, for revoking or renewing certificates as needed, and for managing the Certificate Revocation List (CRL) generation.

In performing these tasks as the administrator, you can search the master lists of certificates or certificate requests by name or number, and then examine specific certificates or requests of interest.

You can then

- approve or reject any individual certificate request,
- revoke specific issued certificates, if they have been compromised or are no longer appropriate, such as being owned by someone who has left the company, or renew any existing certificate during a brief period just before or after it expires.

  See Also: You can specify this renewal-period window: see Chapter 6, "Managing Policies in Oracle Application Server Certificate Authority", in the following sections:

  - the "Certificate Renewal Policy as Shipped" section under "Policy Sub-tab of Oracle Application Server Certificate Authority" and
  - the "Edit" section under "Policy Actions".

All of these certificate management tasks are described in the sections that follow:

- Approving or Rejecting Certificate Requests
- Viewing Details of Certificates
- Revoking Certificates
- Renewing Certificates
- Listing a Single Certificate Request or Issued Certificate
- Using Advanced Search

Approving or Rejecting Certificate Requests

The starting screen of the Certificate Management tab displays a list of all pending certificate requests. To approve or reject a certificate, follow the steps in the corresponding section.

To Approve a Certificate Request

1. Select the desired certificate request by clicking the radio button next to it.
2. Click **View Details**.
   The **Certificate Request Details** screen appears, displaying information about the selected certificate. The contact information of the requestor is displayed. You should follow the organization's practice of authenticating the user, such as sending him email or calling him.
3. Check the validity period, and change it if necessary.
4. For Sub CA certificate issuance, a default path length (for listing trusted certificate authorities) is displayed as 2. (You can change this if required.)
5. Click **Approve**.
   A message appears indicating that the certificate request is approved.
   Please inform the owner of the certificate request so that he can install the certificate.

To Reject a Certificate Request

1. Select the desired certificate request by clicking the radio button next to it. You should reject the certificate request when the requestor cannot be verified, or when the certificate properties are not correct.
2. Click **View Details**.
   The **Certificate Request Details** screen appears, displaying information about the selected certificate.
3. Click **Reject**.
   A message appears indicating that the selected certificate request is rejected.
   Please notify the requestor about the rejection.

Viewing Details of Certificates

From the **Certificate Management** tab, you can select a certificate and view its details.

To select a single certificate, see "Listing a Single Certificate Request or Issued Certificate".

To display a list of certificates, see "Using Advanced Search".

From your search results, select the certificate you wish to review, and click **View Details**. The **Certificate** page appears, showing the certificate's detailed contents. (This page's buttons also enable you to revoke, renew, or install the selected certificate.)

Revoking Certificates

As the administrator, you can revoke certificates before their specified lifetime, and should do so if one of the following situations occurs:

- The owner of the certificate has changed status and no longer has the right to use the certificate.
- The private key of a certificate owner has been compromised.

For a complete list of revocation codes, see "Reasons for Revocation".

To find the target certificate, follow the instructions in "Listing a Single Certificate Request or Issued Certificate" on page 4-11 or "Using Advanced Search" on page 4-12. Once you have selected the correct certificate, you can choose to review its detailed contents by clicking **View Details**, or revoke it with the following steps:

1. To submit the revocation request, click the **Revoke** button.
   The **Revocation Confirmation** screen will appear, where you must choose a revocation reason from these eight choices: Key Compromise, Affiliation Change, CA Compromise, Certificate Hold, Cessation of Operation, Remove From CRL, Superseded, or Unspecified.
2. You can then click **Cancel** to leave the certificate in force, or click **OK** to revoke it, in which case a message appears indicating that the revocation is successful.

See also: End-users who are using OracleAS Single Sign-On or SSL authentication can also revoke their own certificates, as described in "Certificate Revocation" in Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority".

Notes:

- The certificates for the administrator and for the root CA cannot be revoked through the web interface, but rather only by means of the `ocactl` command-line tool.
- Revoking a root CA certificate is a very drastic operation, which will make your OracleAS Certificate Authority installation non-functional and will invalidate the certificates already issued. This operation should only be done when the CA key is compromised, as described in "Revoking a Root CA Certificate" in Command-Line Administration.
- Revoking the administrator's certificate can be required by a key being compromised or stolen, or the administrator role passing to someone new: see "Revoking the OracleAS Certificate Authority Web Administrator's Certificate" in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics".

Reasons for Revocation

An administrator can specify one of the following reasons when revoking a certificate:

- affiliationChange: the certificate holder's relationship with the organization has been terminated
- cACompromise: the private key of the certificate authority who signed the certificate has been compromised
- certificateHold: the certificate has been placed on hold at this time (this amounts to a temporary revocation and it is the only reason code that allows the certificate to be assigned a different status subsequently, either to "unrevoke" the certificate for use or to revoke it with another reason code)
- cessationOfOperation: the organization to whom the certificate was issued has ceased operations, and the CA's certificate is revoked using this code
- keyCompromise: the private key of this certificate has been compromised
- removeFromCRL: the certificate was placed on certificateHold, and is now being "unrevoked"
- superseded: a new certificate has been issued in place of the existing one
- unspecified: the certificate is revoked without a specific reason code; using this revocation reason is not recommended practice, however, since it makes it difficult to understand why a certificate was revoked

Renewing Certificates

The administrator can renew a user certificate 10 days (default policy) before or after it expires, enabling it to continue to be used without interruption. (The administrator can alter the number of days allowed before and after expiration.) Expired certificates can be renewed during the number of days specified for the period before and after the expiration date. Once a certificate expires and is not renewed during this permitted period, it becomes unusable and must be replaced by submitting a new certificate request and having it approved.

To renew a certificate, the administrator selects it (see the sections on listing and searching), clicks **View Details** to display the **Certificate** page, and then clicks **Renew**. If the date is within the established window around the certificate's expiration date (default: 10 days before or after), the certificate can be renewed. Otherwise, an error message appears, regarding the established window.

For OracleAS Single Sign-On-authenticated or SSL-authenticated renewal requests, the same policy governing user certificate renewals (RenewalCertificateRequestConstraints) is applied automatically. When Oracle Application Server Certificate Authority processes renewal requests from end entities, this policy sets the new validity period for the renewed certificate.

Listing a Single Certificate Request or Issued Certificate

From the first page of the user web interface, the Oracle Application Server Certificate Authority administration interface lets you display a specific certificate or certificate request. (To generate a list of certificates or requests that meet criteria you specify, see "Using Advanced Search".)

To find a specific certificate or certificate request, do the following steps:

1. Use the Search pull-down menus:
   - To see all pending certificate requests, select **All Pending Requests**.
   - To display a specific issued certificate, select **Certificate**.

- To display a specific certificate request, select **Certificate Request**.
- To search for a specific Request ID or serial number, select **ID/Serial**.
- To search for a specific Common Name, select **Common Name**.

2. Fill in the Search criteria field with the value appropriate to your search request:
   - For **All Pending Requests**, no further specification is needed.
   - For **ID/Serial**, enter the serial number or the Request ID of the desired certificate or request.
   - For **Common Name**, enter the desired Common Name.
3. Click Go. (Pressing **Enter** instead of clicking **Go** will not work.)
   - A successful search for *a single* certificate request displays a line representing that certificate request. When you click **View Details**, information is displayed regarding the request, including contact, requestor, and validity period, along with buttons labeled **Approve** and **Reject**. Whichever button you click will attach the corresponding status to that request. This status will then appear with this certificate request whenever it is listed as the result of a future search.
   - A successful search for *all* pending certificate requests displays them in a list. If there are more than 25, they are displayed 25 at a time. Clicking the number identifying a request displays its details and permits you to approve or reject it.
   - A successful search for *a single* issued certificate displays a line representing that certificate, along with the **View Details** button. Clicking View Details shows you the data on the certificate along with buttons to **Revoke**, **Renew**, or **Install in Browser**. The Revoke button invalidates that certificate and tags it as Revoked in the database. At some future time, when you choose the Update CRL (Certificate Revocation List) button or when the CRL is automatically regenerated, the latest list of revoked certificates is uploaded to Oracle Internet Directory. Applications in your trust environment can use the CRL to prevent entities with revoked certificates from being authenticated.

     See also: "Updating the Certificate Revocation List (CRL)" on page 4-14 for details.

Using Advanced Search

The **Advanced Search** feature enables you to use more complex search criteria to find and list multiple certificates or certificate requests, as follows:

- For certificate requests, separate searches can list all pending, rejected, or certified requests.
- For requests or issued certificates, you can search by email address, by an advanced DN, by a serial number or range, or by specific entries in the DN, such as name, organization, state, country, and so on. These components must be presented as a contiguous string. For example, certificates owned by cn=lakshmi, ou=st, o=oracle will not be selected or found if you specify cn=lakshmi, o=oracle as the search criteria. In that specification, the search string is not contiguous because ou=st is missing.

From the results listed for a search, the administrator can select

- any single certificate found in a certificate search and, after viewing its details, renew it or revoke it (or install it into the browser), or

- any single request found in a certificate request search, view its details, and either approve or reject issuing a certificate.

In each type of search, after you specify your search parameters, click the **Go** button. OracleAS Certificate Authority displays 25 records at a time.

To perform an advanced search for certificate requests or issued certificates:

1. Click **Advanced Search** on the Certificate Management page.

   The resulting page is structured in sections, each described as follows, so that you can choose the particular type of search you want, from the following choices:

   - Search Certificate Requests using Request Status (Pending, Rejected, or Certified)
   - Search Using DN (Distinguished Name) (certificates or certificate requests)
   - Search Using Advanced DN (certificates or certificate requests)
   - Search Using Serial Number Range or Request ID Range (certificates or certificate requests)
   - Search Using Certificate Status (Valid, Revoked, or Expired certificates)

2. After specifying your search, click the Go button to see a list of the results.

   For all search results, OracleAS Certificate Authority displays 25 records at a time. To see more, use the **Previous** and **Next** buttons to navigate.

Search Certificate Requests using Request Status

Use this section of the Advanced Search page to list certificate requests by status. From the drop down menu, select Pending, Rejected or Certified, and click **Go**. The list of certificate requests matching your status selection will display, 25 records at a time.

Search Using DN (Distinguished Name)

Use this section of the Advanced Search page to list certificates by a particular owner, which can be a server or an end-user. You can search by issued certificates or by requested certificates.

Table 4–2 Elements on Which You Can Search

Element to Search on	Meaning/Content of that Element
Common name	The name on the certificate that you want to find
Email address	Email address that is part of the DN
Organization unit	Name of the unit within the company or organization to which the owner belongs
Organization	Name of the company or organization to which the owner belongs
City/Locality	The city location of the owner
State/Province	The state or province of the owner
Country	Two-letter code for the owner's country

Note: Regarding searches using DN and Advanced DN:

Please note that searches using DN and Advanced DN require a contiguous search. When selecting multiple fields or using advanced DN, please make sure that a contiguous string is formed. For example, for a valid certificate of cn=johnDoe, ou=st, o=oracle, c=us, ou=st, your entering a search string of o=oracle is valid, but ou=st, c=us would not be valid.

Search Using Advanced DN

Use this section of the Advanced Search page to search for issued certificates (**Certificate**) or requested certificate (**Certificate Request**) by the distinguished name of the owner. You can enter the complete DN string instead of entering a value for each RDN string.

See Also: **domain component attribute**.

Search Using Serial Number Range

Use this section of the Advanced Search page to find all issued or requested certificates within a range of serial numbers. You can search by issued certificates or by requested certificates. Select one of those two choices, specify the lowest and highest serial number of interest, and click **Go**.

Table 4–3 Elements Specifying Certificate Serial Number Range for Searches

Element Specifying Range	Meaning/Content of that Element
Lowest Serial Number	Enter the lowest serial number of the range
Highest Serial Number	Enter the highest serial number of the range

Search Using Certificate Status

Use this section of the Advanced Search page to find all valid, revoked, or expired certificates. Select one of those three choices and click **Go**.

Updating the Certificate Revocation List (CRL)

Revoking a certificate should make it unusable in your environment. Making the fact of revocation publicly available ensures that revoked certificates are not misused. Publishing the list of revoked certificates, called the certificate revocation list (CRL), accomplishes this goal because entities granting authentication can first check this list. For example, all the applications in your trust environment can use the CRL to prevent authentication of a revoked certificate.

Automatic CRL generation is enabled by default when OracleAS Certificate Authority is installed. Once you have provided the necessary email information, any failure of CRL generation causes an email to be sent to you automatically.

See also: "Mail Details" on page 5-4.

The first CRL is generated at midnight with a validity period (and regeneration interval) of one day. These values (and auto-generation) are configurable in the Scheduled Jobs section of the Notification subtab within the Configuration Management tab of the Administrator's web interface.

You can generate an updated CRL manually by performing the following steps:

1. From the main **Certificate Management** page, click the **Update Certificate Revocation List (CRL)** button.
   The **Update Certificate Revocation List** form appears.
2. For the **CRL Validity**, specify a number, representing how many days until the next update.
3. For the **Signature Algorithm**, choose from the drop-down menu, such as MD5 with RSA or SHA1 with RSA. (Oracle recommends using SHA-1 because it generates a larger digest, which is inherently more secure against known attacks, such as inversion and brute-force collision attacks.)

After filling in the form, click the **Submit** button. This action generates the CRL.

You can retrieve it for review or saving by choosing **Save CRL** then **Install in Browser** or **Save to Disk**.

> **See also:** Handling Certificate Revocation Lists (CRLs) in Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority".

The Oracle HTTP Server uses this list to check the validity of the SSL certificates it receives, rejecting an SSL connection with any end-entity whose certificate is on the CRL. If your system uses multiple such servers, you will need to copy the CRL to the appropriate path and filename used by those servers as their CRL. Follow the steps established for each server in setting up its CRL.

Similarly, browser and email clients can verify servers they are connecting to, verifying incoming S/MIME email using these CRLs.

Oracle Internet Directory Integration

OracleAS Certificate Authority publishes the following to Oracle Internet Directory:

- Certificates are published to the user's directory entry in the attributes userCertificate and userSMIMECertificate
- Certificate Revocation Lists (CRL) are published to the location cn=oca1,cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext

> **Note:**
> - You must have certificate publishing enabled in order to publish certificates to Oracle Internet Directory. See "Certificate Publishing" in Chapter 5.
> - You can enable the Synchronize Directory option so that, in the event that the directory is temporarily unavailable, certificates will be queue up and published when the directory again becomes available. See "Scheduled Jobs" in Chapter 5.

This section addresses the following topic related to directory integration:

- Retrieving the Certificate Revocation List

Retrieving the Certificate Revocation List

OracleAS Certificate Authority publishes the Certificate Revocation List (CRL), containing the list of revoked certificates, to Oracle Internet Directory. Other applications or users may need to work with the CRL from time to time.

You can obtain the CRL directly from the OracleAS Certificate Authority User home page, as explained in "Handling Certificate Revocation Lists (CRLs)" in Chapter 8.

Alternatively, for programmatic access, you can obtain OracleAS Certificate Authority's CRL using the **ldapsearch** command, which finds specific entries in the directory:

```
ldapsearch -p port -h ldaphost -b
  "cn=oca1,cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext"
  –s scope -L "objectclass=*" certificaterevocationlist
```

where:

- -p connects to the directory at a specified port
- -h specifies the ldap host machine
- -b specifies the DN location
- -s specifies the search scope
- -L prints the entries in LDIF format
- "objectclass=*" indicates the search filter
- certificaterevocationlist is the attribute to retrieve

For example:

```
ldapsearch -p 3060 -h rjackson-sol -b
  "cn=oca1,cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext"
  –s base -L "objectclass=*" certificaterevocationlist
```

which produces the CRL output:

```
dn: cn=oca1,cn=CRLValidation,cn=Validation,cn=PKI,cn=Products,cn=OracleContext
certificaterevocationlist:: MIICADCB6QIBATANBgkqhkiG9w0BAQUFADA8MQswCQYDVQQGE
 wJVUzEPMA0GA1UEChMGb3JhY2xlMRwwGgYDVQQDExNDQS1sa2V0aGFuYS1zdW4tOTA0Fw0wNTAxM
 DQyMjA2MjZaFw0wNTAxMDkyMjA2MjZaMCIwIAIBBRcNMDUwMTA0MjIwNTQzWjAMMAoGA1UdFQQDC
 gEBoFUwUzBRBgNVHSMBAf8ERzBFoUCkPjA8MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGb3JhY2xlM
 RwwGgYDVQQDExNDQS1sa2V0aGFuYS1zdW4tOTA0ggEBMA0GCSqGSIb3DQEBBQUAA4IBAQAwBRgih
 GOB08sWRg2sIaelqLFlUYNvnbtOe4QjdyTPaAy6k31+15jGi1vA7UBw7c0HqLv9r9iHLn7x9MtBj
 Ei8GKj+OJ5GGvrVVnj7ngoSAfpMMhg805m+sgZu0UoBbBkuh9tyAGFzUbxqMCadwakUgEwi7OVsn
 2jaDJilPD/1Lcp975hhlO0JH5hAwpERttSzaZcLqNEPGc9GMiAEUkTVCEa9rPwaw+C42msTZg38N
 7hChaqVf6gj/NpwTOZw98tVyOfU/Iy5tndh5ghbx4PMQ8HoxjXuw0xh6VHTvjmV6q51eTfiAFD3e
 M+IWjxO7fdgL8zUTZ/6HA8fNxZgaJen
```

You can parse this output into a format suitable for your applications. If your applications require access to the CRL on a regular basis, you can set up an automated script to periodically copy the CRL to the file system.

Single Sign-on and OracleAS Certificate Authority

OracleAS Certificate Authority and OracleAS Single Sign-On complement each other in simplifying the provisioning of user certificates and using them to enable PKI authentication to all applications that use OracleAS Single Sign-On. The two configuration choices described in this section can make this collaboration even easier:

- Broadcasting the OracleAS Certificate Authority Certificate Request URL to SSO-Authenticated Users
- Bringing SSO-Authenticated Users to the OracleAS Certificate Authority Certificate Request URL

The first configuration choice, broadcasting, makes it even easier for an OracleAS Single Sign-On user to file a certificate request than it is using the default OracleAS Certificate Authority configuration. OracleAS Certificate Authority's default is to provide certificates when an OracleAS Single Sign-On-authenticated user files a certificate request, a process that takes several steps. That process is described in the "Single Sign-on Authentication (SSO)" section of Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority".

Broadcasting makes it even easier by providing a link that can be sent to all users, enabling them to request an OracleAS Single Sign-On/OracleAS Certificate Authority certificate directly.

The second configuration choice is described in the section following that, Bringing SSO-Authenticated Users to the OracleAS Certificate Authority Certificate Request URL. It explains an OracleAS Certificate Authority configuration command that shortens that process considerably, by simplifying OracleAS Single Sign-On configuration. OracleAS Single Sign-On's default deployment does not automatically use SSL, which PKI authentication requires. So for OracleAS Single Sign-On to leverage OracleAS Certificate Authority-provided user certificates at run-time, OracleAS Single Sign-On server needs to be configured to use SSL and certificates. This second configuration choice, described in "User Certificates and SSO Usage", details how this process can be further simplified, leveraging the usual configuration defaults.

The last two subsections are

- Enabling PKI Authentication with SSO and OracleAS Certificate Authority
- User Certificates and SSO Usage

They describe all the steps required for PKI authentication with OracleAS Certificate Authority and OracleAS Single Sign-On server, and the process Single Sign-On uses for authentication.

Broadcasting the OracleAS Certificate Authority Certificate Request URL to SSO-Authenticated Users

The URL at which OracleAS Single Sign-On users can get an OracleAS Certificate Authority Certificate can be sent by email, as an embedded HTML link, or published as a link in the enterprise portal. These methods give you flexibility in publishing this capability to users who may need it.

This URL, for the SSO Certificate Request, is

```
https://<Oracle_HTTP_host>:<oca_ssl_port>/oca/sso_oca_link
```

in which the sender of such an email should of course replace `<Oracle_HTTP_host>` by the web or IP address of the host, and replace `<oca_ssl_port>` by the Oracle Certificate Authority SSL Server Authentication port number.

For information about the host and port number in the URL, see "Requesting the Administrator Certificate", Step 1.

Users can then click this link and do the same steps detailed in the next section, Bringing SSO-Authenticated Users to the OracleAS Certificate Authority Certificate Request URL.

> **Note:** If port changes have occurred since installation, then the most current information is not in portlist.ini. Instead, sign on to the Oracle Enterprise Manager Control and click the instance on which OracleAS Certificate Authority was installed. Then click the Ports link, find the entry in the Type column that says "OracleAS Certificate Authority Server Authentication (SSL)", and use the number in the adjacent column, headed "Port In Use".

Bringing SSO-Authenticated Users to the OracleAS Certificate Authority Certificate Request URL

Although Oracle Application Server Certificate Authority is configured by default to act on OracleAS Single Sign-On authentication, there are several steps. Users would still need to go to the OracleAS Certificate Authority user interface, select SSO authentication, and then request the certificate. (See Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority", in the Single Sign-on Authentication (SSO) subsection.) Some users might find this process a bit difficult.

Therefore, OracleAS Certificate Authority has a mechanism to simplify the user experience, by sending users directly to the OracleAS Certificate Authority Certificate Request URL after authentication by the OracleAS Single Sign-On server.

Oracle Application Server Certificate Authority can be configured to provide this URL to OracleAS Single Sign-On, for display whenever OracleAS Single Sign-On is not using a certificate to authenticate a user. After OracleAS Single Sign-On authenticates such a user, it then displays the OracleAS Certificate Authority screen enabling that user to request a certificate. After that certificate is created and installed into the user's browser, future authentication can simply use that certificate automatically. (It should be noted, however, that this pop-up screen is shown to all users whether they are interested or not, and to some it could seem an inconvenience.)

> **Note:** To see the pop-up, users must have pop-up-blocking turned off in their browsers.

To configure OracleAS Certificate Authority in this way, the administrator uses the `ocactl` command-line tool (with the administrator password) to issue the following command:

```
ocactl linksso
```

The administrator can also use the `ocactl` command-line tool (with the administrator password) to cancel the use of this URL through OracleAS Single Sign-On, by issuing the following command:

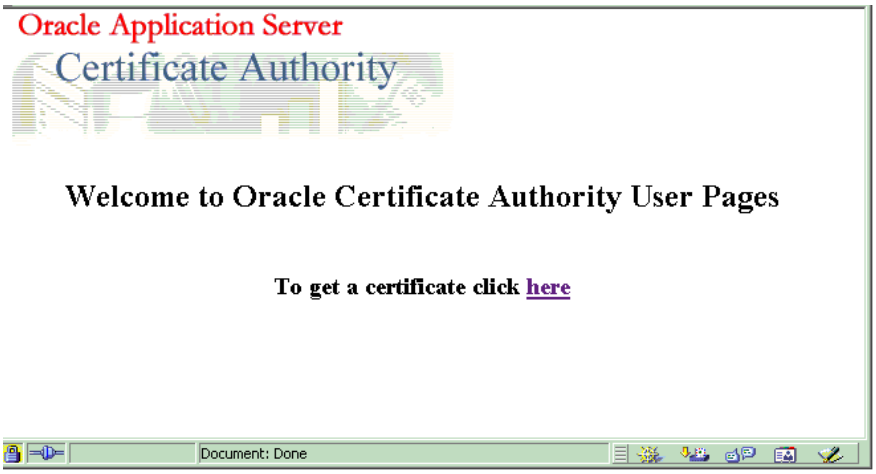
```
ocactl unlinksso
```

Please note that these commands do not require OracleAS Certificate Authority service to be shut down. However, the SSO server needs to be restarted for them to take effect, by using the following commands in the OracleAS Single Sign-On server ORACLE_HOME:

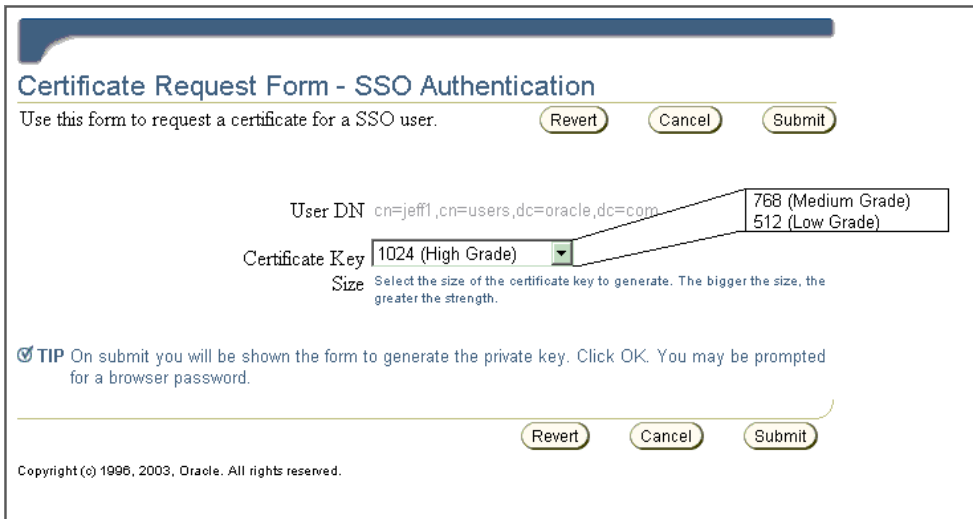
```
$ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=oca
```

```
$ORACLE_HOME/opmn/bin/opmnctl startproc type=oc4j instancename=oca
```

After the `ocactl linksso` command is executed and the OracleAS Single Sign-On server is restarted, the OracleAS Certificate Authority welcome page will be displayed whenever OracleAS Single Sign-On is not using a certificate to authenticate a user. That page looks like the following illustration:



When the OracleAS Single Sign-On user clicks that **here** link, the OracleAS Certificate Authority certificate request page appears:



This composite illustration shows that SSO users must choose a key size and then click **Submit** once their choice is set as desired. (Clicking **Revert** changes the choice back to the default.) After the request is submitted, the key for this certificate is automatically generated (which can take a few minutes). Then the certificate is imported into Oracle Internet Directory and displayed to the user. After the user views the certificate information and clicks **Install in Browser**, the certificate is installed into the user's browser for automatic use.

User Certificates and SSO Usage

After OracleAS Certificate Authority is re-registered with the Single Sign-On server, users who have already authenticated to OracleAS Certificate Authority using Single Sign-On can use their certificates as before.

New users can provision their certificates by using the OracleAS Certificate Authority Certificate Request URL for OracleAS Single Sign-On, as described in the sections referenced earlier.

Once OracleAS Single Sign-On can recognize a user by means of a certificate, she can access applications, including OracleAS Certificate Authority, either by username/password log-in or by certificate.

Thus, after a user logs in with username/password, follows the steps to create a certificate, and installs it into the browser, she can thereafter authenticate herself to the OracleAS Single Sign-On server through PKI.

When the browser of a user presents a certificate to OracleAS Single Sign-On, wanting authentication to use some application, OracleAS Single Sign-On checks that certificate against the directory. If the certificate stored under the user's nickname (and optionally his subscriber name) matches the one presented by the browser, the authentication is successful.

Note: Matching rules in Oracle Internet Directory control how certificates offered are matched to certificates in the directory. See the following references:

- *Oracle Internet Directory Administrator's Guide*, Authentication in Oracle Internet Directory
- *Oracle Internet Directory Administrator's Guide*, Searching the Directory for User Certificates

---

The single sign-on server then supplies the application with a URLC token containing user information, enabling the application to redirect the user to the requested URL. The requested content can then be delivered.

Default Install Values for OracleAS Certificate Authority

Table 4–4 lists the installation default values and other information, including default locations and validity periods for several important wallets.

If you want to change the depth of Sub CA's, that is, the path length, then the CA signing wallet should be regenerated using the command line. Use `ocactl` as described in Appendix A, "Command-Line Administration", in the section entitled "Generating a Sub CA Signing Wallet from OracleAS Certificate Authority".

However, once the CA is regenerated, all previously issued certificates would be invalid. So if you want to change the path length value, the CA signing wallet should be regenerated immediately after the install, as should all dependent wallets such as the SSL wallet.

Note: The OracleAS Certificate Authority schema in one repository can only be used with one OCA.

When installing another OracleAS Certificate Authority, you must not choose a repository that has been used to install an earlier OracleAS Certificate Authority: the OracleAS Certificate Authority configuration tool will fail.

This failure will force you to exit and restart the whole installation.

Table 4–4 Installation Values for Wallets, CRL, and OHS Port (See Note 1.)

Type of Wallet or Value	Default DN	Default Key Size	Default Validity Period	Other Values	Location for This Wallet or Value
CA signing wallet	This DN is entered during installation	2048 (See Notes 2 and 3.)	3560 days	Default Path Length = 3	Database
CA SSL wallet	cn=<hostname> + CA's DN (except CA's CN)	1024 (See Note 4.)	730 days	--	*$OH/oca/wallet/ssl* (See Note 5)
OHS Port for OracleAS Certificate Authority virtual host	--	--	--	6600 and 6601 (See Note 6.)	*$OH/Apache/Apache/conf /ocm_apache.conf* (See Note 7)
Certificate Revocation List	--	--	One day	--	--

Notes to Table 4–4:

1. To set different properties, use ocactl.
2. For the CA signing wallet, used to sign the certificates, only the DN and Key Size can be changed during installation.

   Note:

   For a DN, the DC and EMAIL components must use only printable (ASCII) characters.

   This restriction means that even in a locale that uses a multibyte character set, the DC and EMAIL components for Distinguished Names must still use ASCII characters.

3. For the CA signing wallet, after installation all elements can be changed by running `ocactl generatewallet -type CA` to regenerate the CA signing wallet. You can also change the validity period by renewing this certificate with the desired validity period.
4. Used for the HTTP Server hosting the Certificate Authority. All CA SSL wallet values can be changed by running `ocactl generatewallet -type CASSL`. It can be regenerated at any time, such as expiration, with a commandline option, or replaced with an SSL wallet from a different CA, such as Verisign. This replacement can be done to avoid the warning "CA certificate not trusted" when first connecting to OracleAS Certificate Authority. Possible key sizes are 512, 768, 1024, and 2048, with 1024 the default.
5. $OH stands for `$ORACLE_HOME`, so the full location is `$ORACLE_HOME/oca/wallet/ssl`.
6. Other ports available for use with multiple installs, such as another OracleAS Certificate Authority, include 6602 through 6619.
7. $OH stands for `$ORACLE_HOME`, so the full location is `$ORACLE_HOME/Apache/conf/ocm_apache.conf`.

Note: Two listener ports are defined for OracleAS Certificate Authority in the `ocm_apache.conf` file.

The reason two are needed is that there is a part of the functionality that does not need certificates and a part of the functionality that does need certificates.

Using two listener ports is preferable to using the ClientCertificate optional directive in Apache, which would display a certificate-related dialog for all cases.

Enabling PKI Authentication with SSO and OracleAS Certificate Authority

You need to do certain steps to configure OracleAS Single Sign-On to use certificates. The full procedure appears in Appendix E, but without the detailed context and explanations provided by the *Oracle Application Server Single Sign-On Administrator's Guide*, which you should also read.

Here is an overview to the general steps you will perform:

1. Enable SSL as described in the *Oracle Application Server Single Sign-On Administrator's Guide* in Chapter 7, Enabling SSL.
2. Configure OracleAS Single Sign-On for certificates, as described in the *Oracle Application Server Single Sign-On Administrator's Guide*.
3. Re-register OracleAS Certificate Authority's virtual host to the Single Sign-On Server, as explained in the "Re-registering the Virtual Host with the SSL-Enabled SSO" section of Appendix E, "Enabling SSL and PKI on SSO".

After being PKI-enabled, the OracleAS Single Sign-On server can use certificates to authenticate users for applications rather than requesting username and password. When a user of an application partnering with OracleAS Single Sign-On chooses OracleAS Single Sign-On authentication, the browser asks her to choose a certificate to log in to those applications. The certificate she wants will be one previously installed into the browser. After she selects the appropriate certificate, the OracleAS Single Sign-On server will use that certificate to authenticate her and then redirect her to the partner application she originally requested.

This requirement presents the following issue:

- Users need to log on to OracleAS Certificate Authority to get their certificates.
- Since OracleAS Certificate Authority also uses the OracleAS Single Sign-On authentication service, users without certificates cannot log on to OracleAS Certificate Authority.

This issue is resolved by using multiple authentication levels in the OracleAS Single Sign-On server. Once PKI is enabled, all partner applications will have "medium high" security level (using certificates for authentication), even though OracleAS Certificate Authority can have "medium" security level by using username/password or Windows Native Authentication. This allows OracleAS Certificate Authority to use passwords to authenticate a user before issuing a certificate, but forces other OracleAS Single Sign-On server-enabled applications to use certificates for authentication.

See Appendix E for the full procedure, including those steps needed to configure OracleAS Certificate Authority to have "medium" security level using username/password. The steps specific to the security level are in the "Enabling PKI on SSO" section of Appendix E.

Similarly, OracleAS Certificate Authority can be configured to use other authentication mechanisms like Windows Native Authentication. Assign a security level to the plugin implementing the authentication mechanism and then assign the OracleAS Certificate Authority URL to use that security level as in Step 3 there (in "Enabling PKI on SSO").

> **See Also:** For more detail, see Chapter 6, Multiple Authentication, in the Oracle Application Server Single Sign-On Administrator's Guide.

5

Configuring Oracle Application Server Certificate Authority

The Oracle Application Server Certificate Authority administrative web interface covers the following three broad areas, each accessible from a tab on the home page:

- Certificate issues, regarding issued certificates; requests for certificate issuance, revocation, or renewal: and certificate revocation lists (CRLs)
- Configuration issues, regarding parameters for OracleAS Certificate Authority actions and for implementation of certificate security policies
- Viewing logs of OracleAS Certificate Authority activity

This chapter describes the second and third of those areas: configuration management and viewing logs. It contains the following sections:

> **Note:** For an overview of certificate configuration issues and the certificate policy statement, read "Certificate Requirements and Policies" in Chapter 3 before proceeding.

Structure of the Administration Interface

The home page of the graphical user interface (GUI) for Oracle Application Server Certificate Authority presents three additional tabs, as the following figure shows:

These three subtabs enable you to address specific tasks in managing certificates or configuring the Certificate Authority:

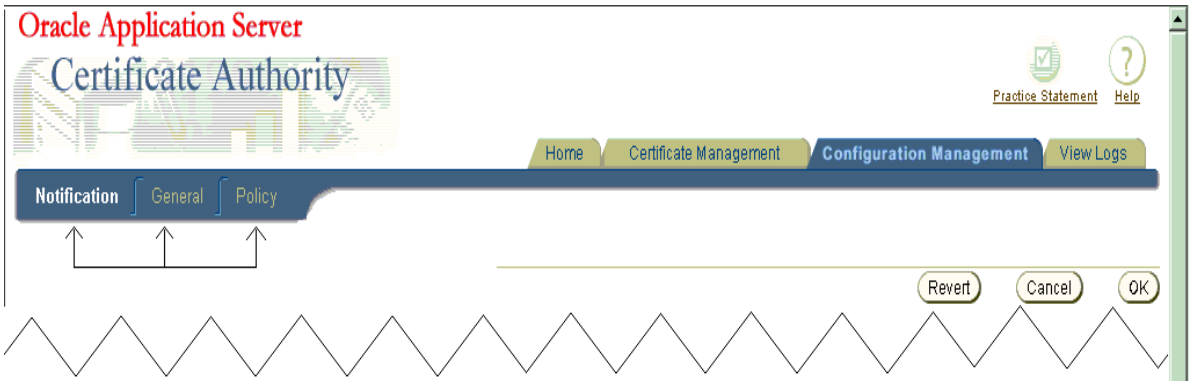
- "Certificate Management Tab", described in Chapter 4, particularly in the section entitled "Managing Certificates"
- "Configuration Management Tab", described in this chapter
- "View Logs Tab", described in this chapter

Configuration Management Tab

The Configuration management tab is one of the four choices available when you first enter the OracleAS Certificate Authority web environment. Clicking the Configuration Management tab on the home page displays the first of the three subtabs, each representing a grouping of the OracleAS Certificate Authority configuration management facilities.

The content and use of those subtabs are explained in the following sections:

- Summary of Configuration Tasks
- Notification Sub-tab
- General Sub-tab
- The "Policy Sub-tab of Oracle Application Server Certificate Authority" and "Policy Actions" are discussed in Chapter 6, "Managing Policies in Oracle Application Server Certificate Authority"



Summary of Configuration Tasks

Table 5–1, Table 5–2, and Table 5–3 list the tasks encompassed by the Notification, General, and Policy sub-tabs of Configuration Management and provide links to discussions of those tasks.

Table 5–1 Notification Sub-tab Tasks and Discussions in Configuration Management

Notification Sub-tab Tasks and Data	**Links to Task Discussions**
Specify server name and email contacts for alerts and notifications.	▪ Mail Details
Specify desired types of alerts.	▪ Alerts
Enable auto-generation of CRLs and specify its start time and the interval between generating CRLs, and the start time and interval for directory synchronizations	▪ Scheduled Jobs

Table 5–2 General Sub-tab Tasks and Discussions in Configuration Management

General Sub-tab Tasks and Data	**Links to Task Discussions**
Specify that certificate publishing uses SSL or non-SSL communication channel with Oracle Internet Directory.	▪ Certificate Publishing
Specify that end-users can use SSL and OracleAS Single Sign-On authentication for certificate management.	▪ SSL and SSO Authentication
Specify default usage for client certificates	▪ Default usage for client certificates
Specify subject alternative name extension	▪ Subject Alternate Name Extension
Specify logging, tracing, both, or neither.	▪ Logging and Tracing
Specify default values for DN components shown in enrollment.	▪ Default Base DN Components
See configuration parameters for the database and directory.	▪ Database Settings, Directory Settings

Table 5–3 Policy Sub-tab Tasks and Discussions in Configuration Management

Policy Sub-tab of Oracle Application Server Certificate Authority Tasks and Data (in Chapter 6)	Links to Task Discussions
See the policies applicable to available operations, such as certificate requests, revocations, or renewals.	▪ Default Certificate Request Policies ▪ Default Certificate Revocation Policy ▪ Certificate Renewal Policy as Shipped
Edit, enable, disable, delete, add, or reorder policies.	▪ Policy Actions

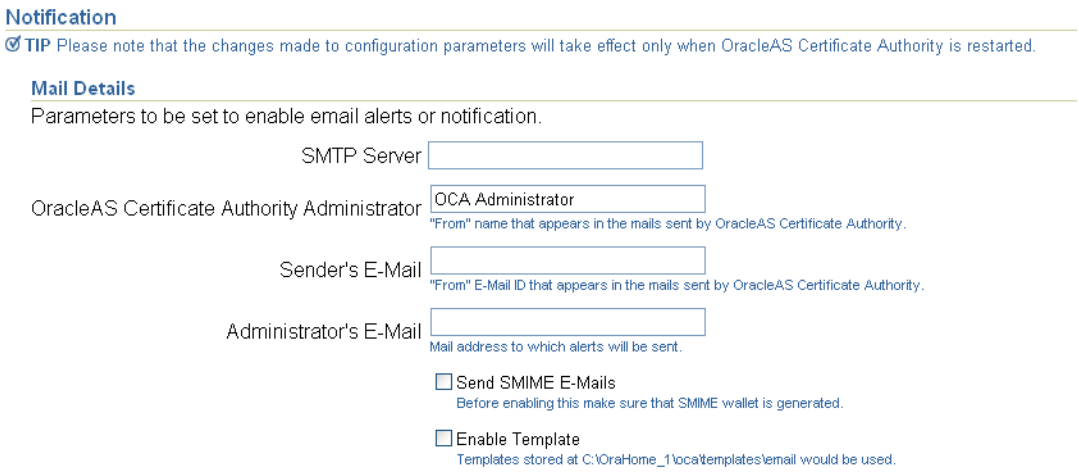
Notification Sub-tab

Notification parameters control what events trigger notification emails to the administrator, how those emails are generated, and how often checking is done to reveal such events.

Changes you make to **Notification** configuration parameters will take effect only after OracleAS Certificate Authority is restarted.

Mail Details

Mail parameters enable email notifications to be sent to the email address you specify for the administrator and to OracleAS Certificate Authority users when appropriate. (Before selecting encrypted (S/MIME) email, you must first create an S/MIME certificate and wallet.) Notification emails use your specified server, sender, and template. You specify your choices in the following portion of the Notification subtab screen:



Note that the hint following **Enable Template** will, after installation, display the exact path to the template directory. For example, if `$Oracle_Home` is defined during installation as `/private/sitename/username`, then this hint will display as "Templates stored at /private/sitename/username/oca/templates/email."

See Also:

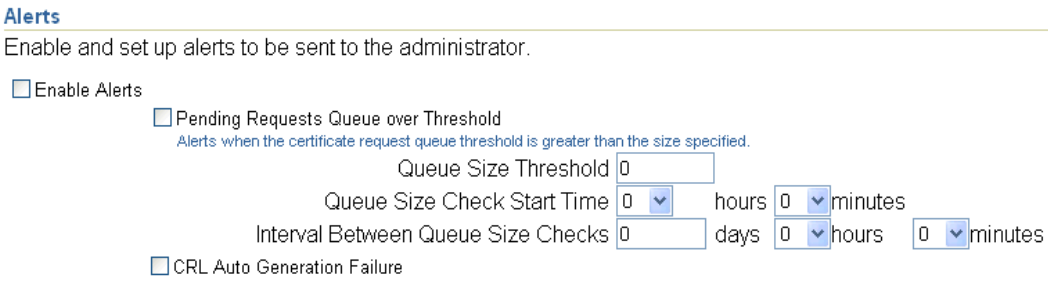
- "Regenerating the CA SSL and CA S/MIME Wallets" in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics"
- Appendix G, "S/MIME with OracleAS Certificate Authority"

Alerts

Alerts parameters enable you to specify whether you are to receive alerts in the following circumstances (if you have specified the email information):

- When the number of pending certificate requests exceeds the queue threshold you specify here, to be checked on the schedule you specify here (start time and repeat interval). The start time refers to the server's time zone and is specified in a 24-hour format. For example, a start time of 14 hours 30 minutes starts the first check at 2:30 in the afternoon, server time. The interval (default one day) is added to that time to specify the time of the next check; it must be nonzero. Changes survive restarts.
- Whenever automatic generation of the CRL fails. Such failure could occur, for example, if the database or Oracle Internet Directory were temporarily unavailable. Other rare possibilities include unpredictable runtime or configuration errors related to memory, I/O, or connectivity issues.

You specify your choices in the following portion of the Notification subtab screen:

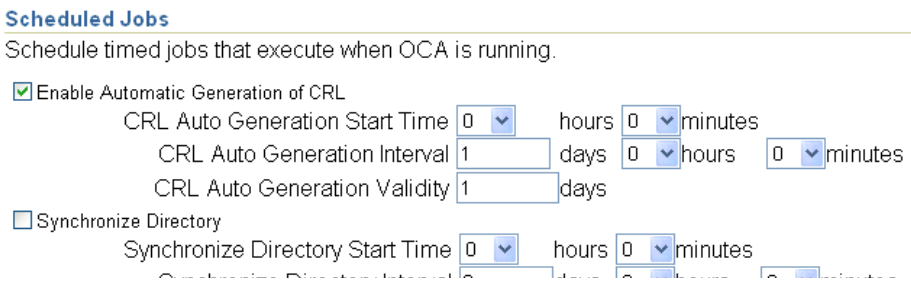


Scheduled Jobs

Scheduled Jobs parameters enable you to make the following choices about automatic jobs:

- Whether a CRL is to be generated automatically, starting when, and how often thereafter. This feature, enabled by default when OracleAS Certificate Authority is installed, establishes a reliable, timely, and regular process supporting applications that depend on the CRL to detect revoked or expired certificates. The start time refers to the server's time zone and is specified in a 24-hour format. For example, a start time of 14 hours 30 minutes starts the first job at 2:30 in the afternoon, server time. The interval you specify is added to that start time to specify the time of the next CRL generation (default one day); it must be nonzero. Changes survive restarts.
- Whether directories are to be synchronized, starting when, and how often thereafter. This feature ensures timely, regular updates to the certificate information in the Oracle Internet Directory. Even certificates issued (or revoked or expired) during any temporary directory downtime will be published (or removed) during synchronization. The start time refers to the server's time zone and is specified in a 24-hour format. For example, a start time of 14 hours 30 minutes starts the first job at 2:30 in the afternoon, server time. The interval you specify (default one day) is added to that start time to specify the time of the next synchronization; it must be nonzero. Changes survive restarts.

You specify your choices in the following portion of the Notification subtab screen:



Email Templates

You can specify and customize the body of e-mail alerts and notifications using templates, which are stored in the following directory:

```
$ORACLE_HOME/oca/templates/email
```

> **Note:** Templates are turned off by default and must be explicitly enabled.

You can use tokens to format the e-mail to provide specific information. These tokens are replaced before the e-mail is sent. Table 5–4 lists the notifications, filenames for e-mail format and the supported tokens.

Table 5–4 Tokens for Customizing E-mail

Notifications	Template File Name	Supported Tokens
CertificateRequestNotify	`reqacc.txt`	#NAME#, #REQUESTID#, #SUBJECTDN#, #PHONE#, #EMAIL#
RequestApprovalNotify	`reqapp.txt`	#NAME#, #REQUESTID#, #SUBJECTDN#, #SERIALNUM#, #OCAURL#, #PHONE#, #EMAIL#, #VALIDITY#
RequestRejectionNotify	reqrej.txt	#NAME#, #REQUESTID#, #SUBJECTDN#, #PHONE#, #EMAIL#
PendingRequestsAlert	`pendreq.txt`	#NAME#, #NUMBERREQUESTS#
CRLAutoGenFailureAlert	`crlfail.txt`	#NAME#

> **Note:** If you do not check the box for **Use Template in Configuration Management** in the Notification screen, then templates are not used. All alerts and notifications would be predefined text that cannot be changed.

Values for the tokens

Table 5–5 describes the values that will replace each of the listed tokens before the alert or notification is sent:

Table 5–5 Supported Token Values

Notifications and Template File Names	Supported Tokens and the Data to Replace Them
CertificateRequestNotify Template = *reqacc.txt*	#NAME#: Replace with the contact data Name specified in the certificate request. #REQUESTID#: Replace with the request ID issued by OracleAS Certificate Authority to this request. #SUBJECTDN#: Replace with the DN in the certificate request. #PHONE#: Replace with the contact data phone number in the certificate request. #EMAIL#: Replace with the contact data email address in the certificate request.
RequestApprovalNotify Template = *reqapp.txt*	#NAME#: Replace with the contact data Name specified in the certificate request. #REQUESTID#: Replace with the request ID issued by OracleAS Certificate Authority to this request. #SUBJECTDN#: Replace with the DN in the certificate request. #SERIALNUM#: Replace with the serial number of the certificate #OCAURL#: Replace with the URL of the user home page #PHONE#: Replace with the contact data phone number in the certificate request. #EMAIL#: Replace with the contact data email address in the certificate request. #VALIDITY#: Replace with the validity period for which the certificate request is approved by the administrator.
RequestRejectionNotify Template = *reqrej.txt*	#NAME#: Replace with the contact data Name in the certificate request. #REQUESTID#: Replace with the request ID issued by OracleAS Certificate Authority to this request. #SUBJECTDN#: Replace with the DN in the certificate request #PHONE#: Replace with the contact data phone number in the certificate request. #EMAIL#: Replace with the contact data email address in the certificate request.
PendingRequestsAlert Template = *pendreq.txt*	#NAME#: Replace with the value specified in the OracleAS Certificate Authority Administrator field under Configuration Management in the Notification screen. #NUMBERREQUESTS#: Replace with the number of pending requests in the OracleAS Certificate Authority repository
CRLAutoGenFailureAlert Template = *crlfail.txt*	#NAME#: Replace with the value specified in the OCA Administrator field under Configuration Management in the Notification screen.

Note:

The language in which you edit these templates is used in the final results, so it is best to use the language of the server, because the message body is encoded in the language of the server locale.

If you do not use templates, then all alerts and notifications will appear in the language of the server locale.

General Sub-tab

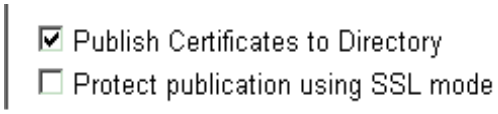
This sub-tab enables you to set parameters controlling the following tasks:

- Certificate Publishing
- SSL and SSO Authentication
- Default usage for client certificates
- Subject Alternate Name Extension
- Logging and Tracing
- Default Base DN Components
- Database Settings
- Directory Settings

Changes you make to **General** configuration parameters will take effect only after OracleAS Certificate Authority is restarted.

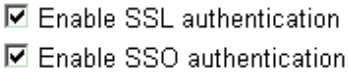
Certificate Publishing

The choices in this section enable you to publish certificates to the directory. Since OracleAS Certificate Authority always connects to Oracle Internet Directory by using the SSL port, the second checkbox shown here is no longer needed ("Protect publication using SSL mode"). The direct Diffie Hellman SSL connection does not require authentication, and OracleAS Certificate Authority then authenticates itself to the directory server by sending its username/password over the now-secured SSL connection.



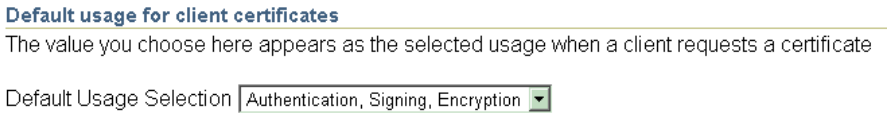
SSL and SSO Authentication

The choices in this section let you specify that SSL or OracleAS Single Sign-On users can be recognized automatically, meaning that their existing certificates (or OracleAS Single Sign-On authentication) are accepted as authenticating their identities. Enabled by default, such acceptance means OracleAS Certificate Authority will issue them a new certificate without administrator intervention.



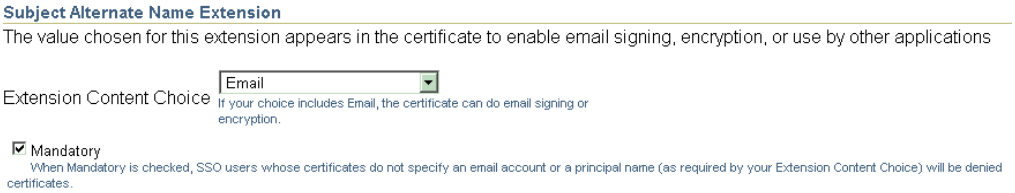
Default usage for client certificates

The value you choose here appears as the selected usage when a client requests a certificate. This does not prevent the user from selecting a different usage from the drop-down list, which includes authentication, encryption, signing, and combinations of these, plus CA signing, and code signing.



Subject Alternate Name Extension

For SSO users, the value chosen for this extension appears in the certificate to enable email encryption, signing, or use by other applications. Your choices are shown in Extension Content Choice.

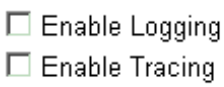


Extension Content Choice Choose from None, Email, Principal Name (UID), or Email/Principal Name (UID). The choice made here appears in the certificate as the subject alternate name, enabling email encryption, signing, or use by other applications. (UID means user identifier or unique identifier.) Choosing "Email/Principal Name (UID)" causes both to be listed in the certificate.

Mandatory If this box is checked, the Subject Alternate Name Extension is required for all SSO-authenticated certificates. If an email address or Principal Name cannot be found in Oracle Internet Directory for the user named in an SSO-authenticated certificate request, that request will be denied. An error message will state that an SSO-authenticated certificate could not be issued because an email account was not found in the Oracle Internet Directory, and that the requestor should contact the administrator.

Logging and Tracing

The choices in this section let you specify whether to create a log file of all user activities, a tracing file of all details for every error, or both.



Logs are stored in the OracleAS Certificate Authority repository; you can view them from the View Logs tab. Trace is stored on the file system, in the file at *$ORACLE_HOME/oca/logs/oca.trc*.

Default Base DN Components

The values you fill in here will be used to pre-fill some of the Distinguished Name elements on the manual enrollment request form used to submit certificate requests.

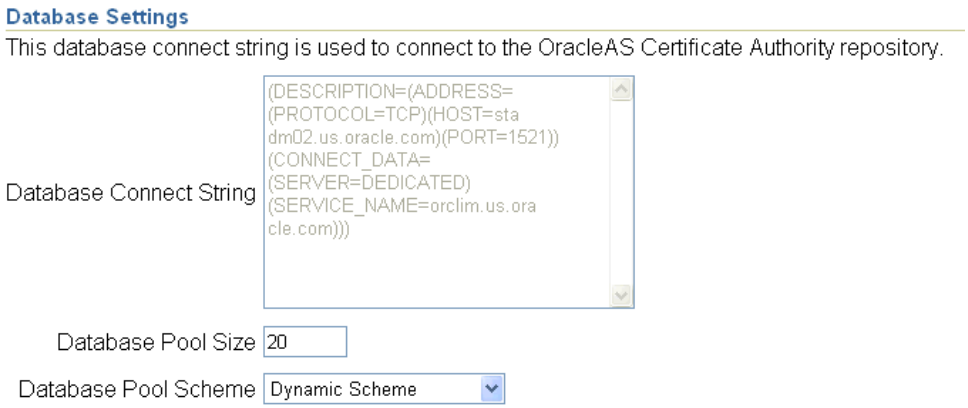
This facility is simply for the users' convenience, supplying common fields. The values you fill in here can be overridden as needed.

Database Settings

The settings shown here tell you the database connect string, the database pool size, and the database pool scheme. The connect string is the one being used to connect to the OracleAS Certificate Authority repository.

You enter, in the **Database Pool Size** text box, the number of connections to the database (default: 20) that represents how many users you expect to access OracleAS Certificate Authority concurrently. Specify a number slightly larger than what you expect; for example, if you expect about 25 concurrent users, specify 27 or 28 as the **Database Pool Size**. When a user in that pool of connections exits OracleAS Certificate Authority, the connection becomes available to the next new user. For each user beyond that number, a new connection will be opened, to be closed as soon as that user has exited OracleAS Certificate Authority.

In **Database Pool Scheme**, you choose how you want to treat connection requests that come in after all the connections you specified in **Database Pool Size** are in use. The default, "dynamic," means that a new connection is opened immediately for the new user, and after that user exits OracleAS Certificate Authority, that connection is closed. If you choose "Fixed wait scheme", then after 20 users (or the number you specify) are connected to OracleAS Certificate Authority, every subsequent user attempting to connect simply waits until one of the original 20 users exits. If you choose "Fixed Return Null," then after the original pool size limit is reached, each new user attempting to connect simply gets an error message. No new user can connect until an existing OracleAS Certificate Authority user exits.



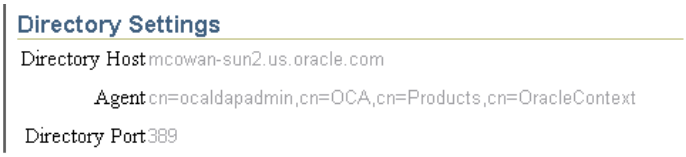
The database connect strings only change if OracleAS Certificate Authority's repository moves to a new location (or if a change is made directly to the connection string). Examples include changing the nodes or the port used for connection. In these cases, you can use the `ocactl updateconnection` command to update the repository connection settings, and then restart OracleAS Certificate Authority to use the new connection information.

See Also:

- "Performance Tuning for OracleAS Certificate Authority" in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics" on page 7-8
- `updateconnection` in Table A–2, " Operations and Parameters of the OracleAS Certificate Authority (OCA) ocactl Tool" of Appendix A, "Command-Line Administration".

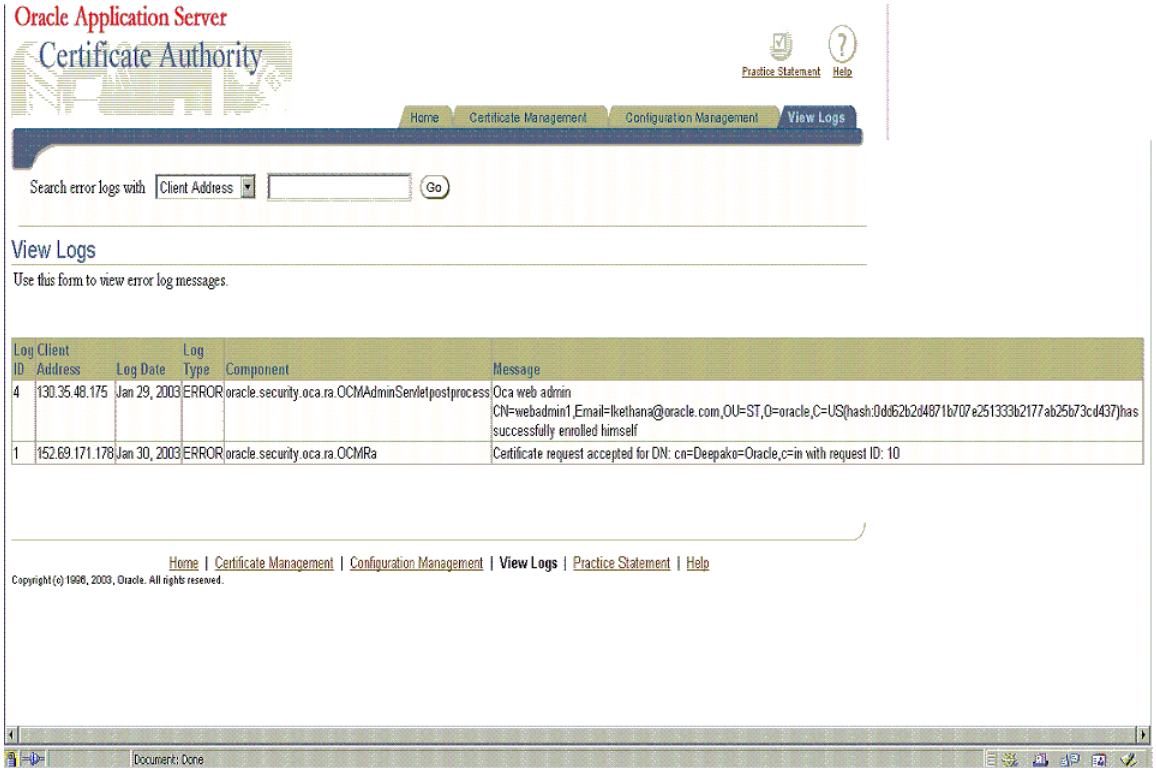
Directory Settings

The settings shown here simply tell you the host, agent, and port being used to connect with Oracle Internet Directory. If a change is made to the connection string, you can use the `ocactl updateconnection` command to update the repository connection settings, and then restart OracleAS Certificate Authority to use the new connection information.



View Logs Tab

The View Logs page enables you to view logs that record messages regarding transactions or errors occurring during use of OracleAS Certificate Authority. Such a screen would look like this:



Each line of such a log contains six elements, beginning with a log id number, the IP address that initiated the client activity, and the date of the action. Each line also includes the log entry type, the component of OracleAS Certificate Authority generating the entry, and the component's message about the activity.

These logs can be searched, for example by client (IP) address or message content. The logs enable an administrator to learn where requests originated and what messages were issued for those requests. Searching enables review of specific message types, such as pertaining to rejections, and specific source IP addresses that may have initiated the actions that caused such messages.

> **See Also:** Clearing Log or Trace Information for OracleAS Certificate Authority on page 7-13 in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics"

6

Managing Policies in Oracle Application Server Certificate Authority

Oracle Application Server Certificate Authority automatically enforces the policies specified by the organization to apply to requests for certificate issuance, revocation, or renewal. The policy rules supplied with OracleAS Certificate Authority support standard needs. They are, however, configurable by the administrator, using the Configuration Management tab of the OracleAS Certificate Authority web interface, or by adding custom policy plug-ins to meet the site's needs. The administrator can also bypass policies by disabling them, if needed.

This chapter explains the policy management component of OracleAS Certificate Authority, including the tools for developing custom policy plug-ins.

Topics in this chapter include:

Definitions

Table 6–1 Policy Concepts, Terms, and Definitions in OracleAS Certificate Authority

Concept or Term	Definition
Policy Rule or Policy	In Oracle Application Server Certificate Authority, a policy rule is a set of defaults and ranges for the values of parameters that apply to certificates, requests, and so on. For example, a policy rule for validity period can specify 365 days as the minimum validity, 730 days as the default, and 3650 days as the maximum. Policy rules can also contain predicates, which limit or alter the application of the rule. Without predicates, a policy rule for a particular operation, such as renewal, applies to all such requests.

Table 6–1 (Cont.) Policy Concepts, Terms, and Definitions in OracleAS Certificate

Concept or Term	Definition
Predicate	A predicate, in OracleAS Certificate Authority, is an expression you create to identify a type of certificate or certificate request, plus corresponding values. When the type of certificate or certificate request matches the predicate expression, these corresponding values are used to evaluate the request's validity, instead of the policy's default values. Predicates are available only for OracleAS Certificate Authority default policies; they cannot be used with custom policies, which are discussed at section "Developing a Custom Policy Plug-in". Examples: Type=="client", Type=="server", or Type=="*"
Plug-in	A Java class that implements a policy rule.

Overview of Policy Management

Policy management means formulating and applying policies (sets of rules) chosen by the Oracle Application Server Certificate Authority administrator to enforce organizational constraints. Constraint examples include the choices offered to the user for selecting key size and validity period.

As the administrator, you can use policies shipped with OracleAS Certificate Authority to define the following operations:

- How OracleAS Certificate Authority is to evaluate incoming requests for certificate issuance, revocation, and renewal
- What restrictions the CA is to impose on certificate parameters, such as validity length and key length, or on the issuance of multiple certificates with the same subject name and intended usage

You can enable, disable, or modify policy rules using the edit capability of the Configuration Management tab in OracleAS Certificate Authority's web interface. See the section entitled "Policy Sub-tab of Oracle Application Server Certificate Authority".

You can also create new rules and develop policy plug-ins to embody them. Each rule is embodied in a policy plug-in, that is, a Java class that implements the evaluations or restrictions chosen by the administrator. There is a one-to-one mapping between a policy rule and a policy plug-in. OracleAS Certificate Authority's default plug-ins cover most of the common policy configuration needs. To write a policy plug-in, the administrator must follow good programming practices and use the APIs provided by the OracleAS Certificate Authority package, as described in "Developing a Custom Policy Plug-in" on page 6-22.

After developing a new plug-in defining a site-specific policy, you can use that same Policy subtab to name and describe it to OracleAS Certificate Authority. If you also enable it, OracleAS Certificate Authority will enforce the new rules as it does its own.

Policy rules are enforced by the policy processor module in the OracleAS Certificate Authority engine. This processor module applies all enabled rules sequentially; rules that are not enabled, or are disabled, are not enforced. The order used is the order in which they are listed on the Policy Rules page for each operation, in the Policy subtab. That is, the processor module calls the policy plug-ins in the order specified on the Policy Rules page for each operation. Every incoming request is subjected to all applicable enabled policy rules for that type of operation, that is, request, renewal, or revocation. If a rule is enabled and its terms are not met by an incoming request, that request is rejected.

Each policy rule relates to one or more attributes of a request for certificate issuance, revocation, or renewal. For example, one such attribute relates to minimum and maximum key sizes used in an RSA algorithm. The relevant default policy checks that all such attributes are within the algorithm's valid ranges.

Policies are administered through the web interface using the Policy subtab of the administrative interface.

Further details of policy processing involving predicates are discussed in the section titled "Predicates in Policy Rules".

Oracle Application Server Certificate Authority Policies

Oracle Application Server Certificate Authority supplies constraint-specific policy rules that the policy processor uses to evaluate an incoming certificate enrollment, revocation, or renewal request. Within each rule, you can configure OracleAS Certificate Authority to check an incoming request for particular attributes, and either accept these attributes, alter them, or reject the request.

If a policy rule is enabled, the OracleAS Certificate Authority server applies the rule to the certificate request being processed

Table 6–2 lists the default constraint-specific policy rules; the first column contains links to the discussion of each policy rule.

Table 6–2 Default Constraint-specific Policy Rules

Policy Rule Name	Function	Default State
RSAKeyConstraints	Enforces constraints on key lengths	Enabled
ValidityRule	Enforces a specified validity period on certificates	Enabled
UniqueCertificateConstraint	Prohibits multiple certificates being issued to the same name for the same usage	Enabled
RevocationConstraints	Allows or rejects requests for revocation of expired certificates	Enabled
RenewalRequestConstraint	Allows or rejects requests for renewal of expired certificates	Enabled

RSAKeyConstraints

The RSAKeyConstraints policy rule imposes constraints on the minimum and maximum key sizes used for RSA public/private keys.

Table 6–3 describes the parameters of the RSA key constraints module.

Table 6–3 Parameters in the RSA Key Constraints Policy Rule

Parameter	Description
Status (Enabled/Disabled) Default=Enabled	Specifies (on the Policy Rules page) whether the rule is enabled or disabled. If you enable the rule and set the remaining parameters correctly, Oracle Certificate Manager applies the rule to certificates specified by the predicate expressions. If you disable the rule, the Oracle Certificate Manager allows the RSA key size to be any multiple of 16 between 512 and 4096 bits.

Table 6–3 (Cont.) Parameters in the RSA Key Constraints Policy Rule

Parameter	Description
predicate Default: "*"	Specifies the predicate expression for this rule, to limit the types of certificate to which this rule will apply. If you want the rule to be applied to certificate requests, type * in this field. Examples: Type=="client" Type=="*" See "Predicates in Policy Rules".
minSize Default=1024	Specifies the minimum length (bits), for the RSA key (the length of the modulus in bits). The value must be smaller than or equal to the one specified by the maxSize parameter. Valid values: 512, 1024, 2048, or 4096 bits.
maxSize Default=2048	Specifies the minimum length (bits), for the key (the length of the modulus in bits). The value must be greater than or equal to the one specified by the minSize parameter. Valid values: 512, 1024, 2048, or 4096 bits.

An administrator can specify multiple sets of predicates, minSize, and maxSize using complex predicate expressions.

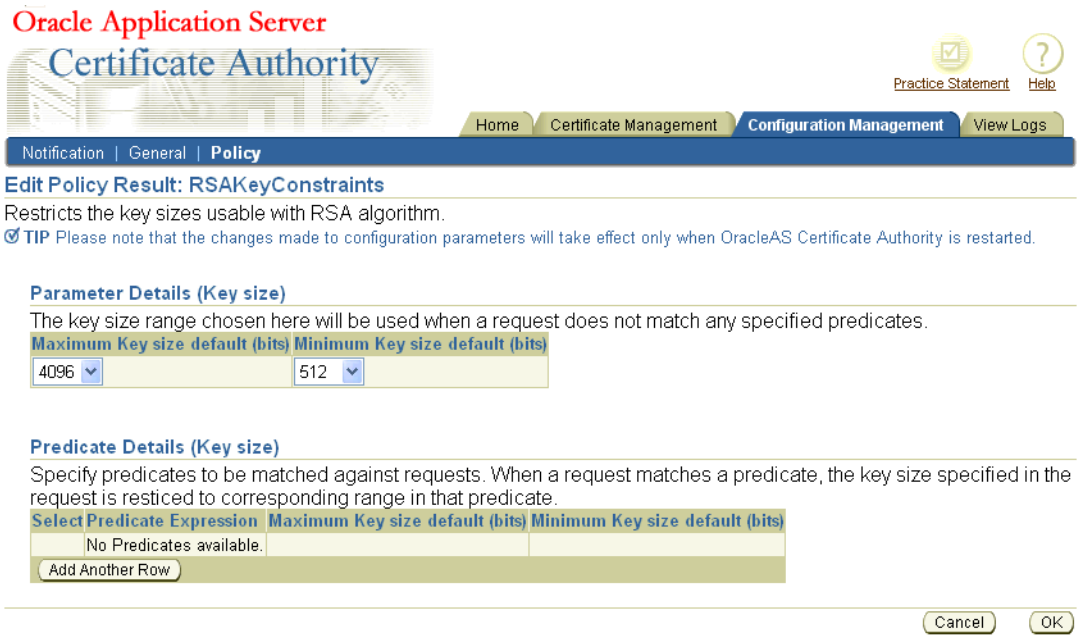
For example, an organization might need (minsize, maxsize) for Sales and Finance departments to be (512,1024) and (1024,2048), respectively. Multiple predicate expressions and value sets can be used to specify this requirement:

Predicate 1: dn=="ou=Sales"

minSize, maxSize are specified as 512,1024

Predicate 2: dn=="ou=Marketing"

minSize, maxSize are specified as 1024,2048



ValidityRule

The ValidityRule policy rule determines if the validity period in the certificate request is acceptable and enforces the minimum and maximum validity dates as follows:

- For automatic-user certificate requests (OracleAS Single Sign-On Server or SSL authentication), this rule sets the validity period.
- If a request for a manual user certificate or a server certificate does not meet the policy, that request is rejected.
- Irrespective of the validity set by this rule, the expiration date of the certificate cannot be later than that of the CA certificate. This check cannot be disabled.

Table 6–4 describes the parameters for the issuance validity constraints module. The illustration at the end of this section shows how they appear in the web interface.

Table 6–4 Parameters in the ValidityRule Policy

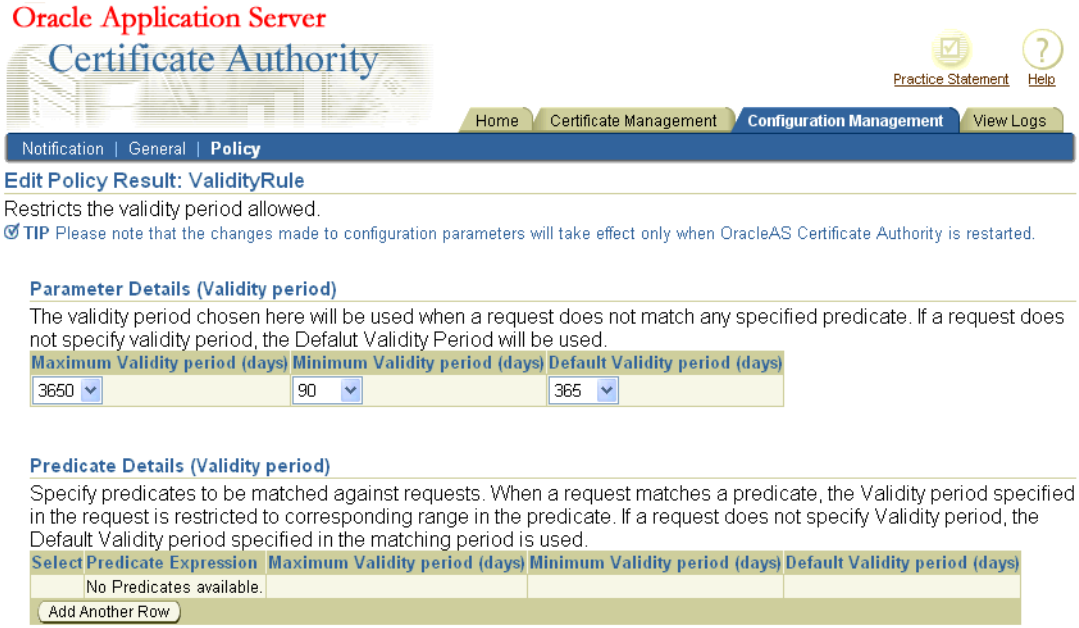
Parameter	Description
Status (Enabled/Disabled) Default=Enabled	Specifies (on the Policy Rules page) whether the rule is enabled or disabled. If you enable the rule and set the other parameters correctly, Oracle Application Server Certificate Authority checks the configured validity period in certificates specified by the predicate parameter. If you disable the rule, Oracle Application Server Certificate Authority does not use the period specified in the rule to check the configured validity period in certificates. Instead, it uses the validity period specified in the request.
Minimum Validity Default Minimum=90 days	Specifies the minimum validity period (days) for certificates. Valid values: an integer greater than zero and less than the value specified by the Maximum Validity parameter.
Maximum Validity Default Maximum=3650 days	Specifies the maximum validity period (days) for certificates. Valid values: an integer greater than zero and greater than the value specified by the Minimum Validity parameter. Default validity period is the Default Maximum: 3650 days.
validityPeriod Default = 365 days	Specifies the validity period for OracleAS Single Sign-On / SSL Users. Must be between minimum and maximum validity period. Value set to 365 days.
predicate	Specifies the predicate expression for this rule, to limit the types of certificate to which this rule will apply. If you want the rule to be applied to all certificate requests, type * in the field. Examples: Type=="client" Type=="*" See "Predicates in Policy Rules".

If this rule is disabled, OracleAS Certificate Authority issues certificates with the validity specified in the certificate request, as long as that period is less than or equal to the validity period of the certificate of the CA.

For the automatic client users (that is, OracleAS Single Sign-On server- and SSL-authenticated users), the validity is automatically set by using the "Default Validity period" in the matching predicate specified in the policy. For all other users, validity is expected as part of the certificate request. This capability enables an

administrator to specify the exact validity period that automatic users will get, eliminating the need for such users to enter this value.

The validity period that applies to the Certificate Authority can be 5 years or even 10 years. The longer the validity period is for the CA, the longer its issued certificates remain valid without the need for renewal or replacement. The installation process for OracleAS Certificate Authority uses a default of 10 years for the root CA. The following illustration shows the validity-rule parameters.



UniqueCertificateConstraint

The UniqueCertificateConstraint policy rule prevents OracleAS Certificate Authority from issuing multiple certificates to the same subject name for the same usage. When enabled, this policy can reject such a request, if the parameter in the policy is set to prohibit multiple such certificates.

The policy checks the incoming request against the Oracle Application Server Certificate Authority repository for any certificates matching the subject DN of the incoming certificate request. If an existing certificate is found for the subject DN, then the certificate usages (encryption, signing, and so on.) are checked. If there is an existing certificate for the requesting DN that also specifies the same usage as is being requested, the request is rejected if the policy is set to reject multiples.

Table 6–5 describes the parameters for the UniqueCertificateConstraint module.

Table 6–5 Parameters in the UniqueCertificateConstraint Policy Rule

Parameter	Description
Status (Enabled/Disabled) Default=Enabled	Specifies (on the Policy Rules page) whether the rule is enabled or disabled. When enabled, the rule uses the checkbox allowing multiple certificates with the same usage. If it prohibits multiple certificates for the same subject name and the same usage, the request is rejected. If you disable the rule, OracleAS Certificate Authority will approve multiple certificate requests for the same subject name and the same usage.
Checkbox allowing multiple certificates to have the same DN and the same usage Default: checked	When checked, this box allows OracleAS Certificate Authority to issue a new certificate for a DN that already has a certificate even if the usages are same. When unchecked, this box prevents OracleAS Certificate Authority from issuing a new certificate to a DN that already has a certificate if the new and old certificate usages would be the same.

RevocationConstraints

The OracleAS Certificate Authority Administrator can restrict revocation of expired certificates by applying this policy to user certificate revocation requests. If this policy is enabled, revocation of an expired certificate is allowed after its expiration date. If you don't want to allow revocation of expired certificates in your PKI setup, you can use the policy to configure Oracle Application Server Certificate Authority accordingly.

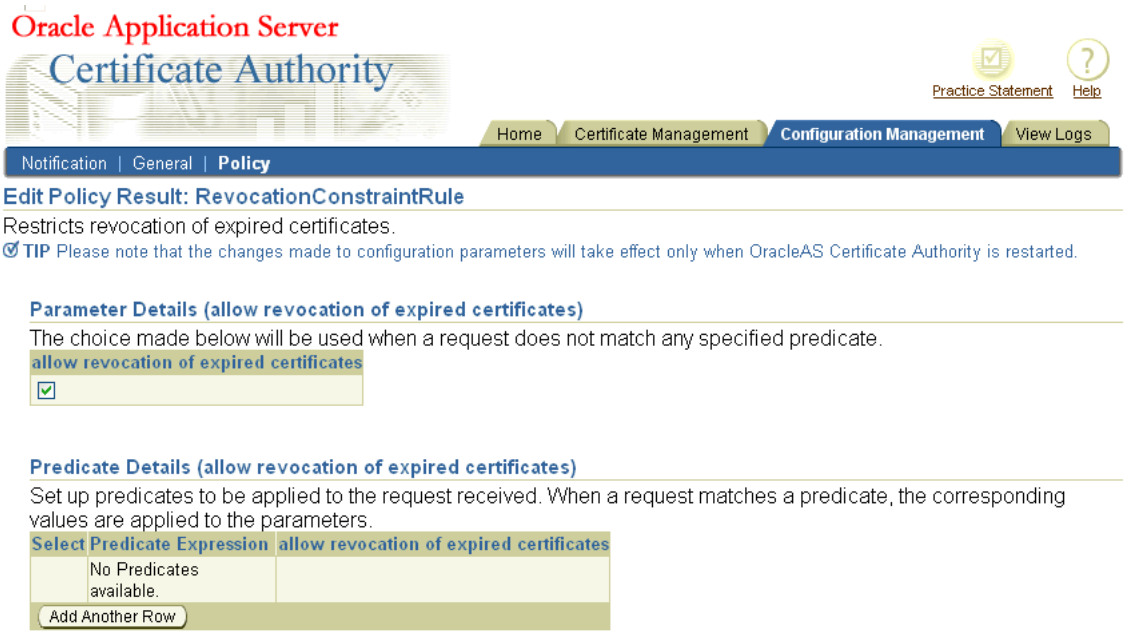


Table 6–6 describes the parameters of the revocation constraints module.

Table 6–6 Parameters in the Revocation Constraints Policy Rule

Parameter	Description
Status (Enabled/Disabled)	Specifies (on the Policy Rules page) whether the rule is enabled or disabled.
Default=Enabled	If you enable the rule and set the other parameters correctly, OracleAS Certificate Authority verifies the validity period of the certificate being revoked, checks the value assigned to the allowExpiredCerts parameter, and accordingly allows or denies the revocation request. If you disable the rule, OracleAS Certificate Authority does not verify the validity period of the certificate being revoked, nor whether it is expired. The certificate is simply revoked.
allowExpiredCerts	Specifies whether to allow (True) or prevent (False) revocation of expired certificates.
Default: True	The default is True (allow).

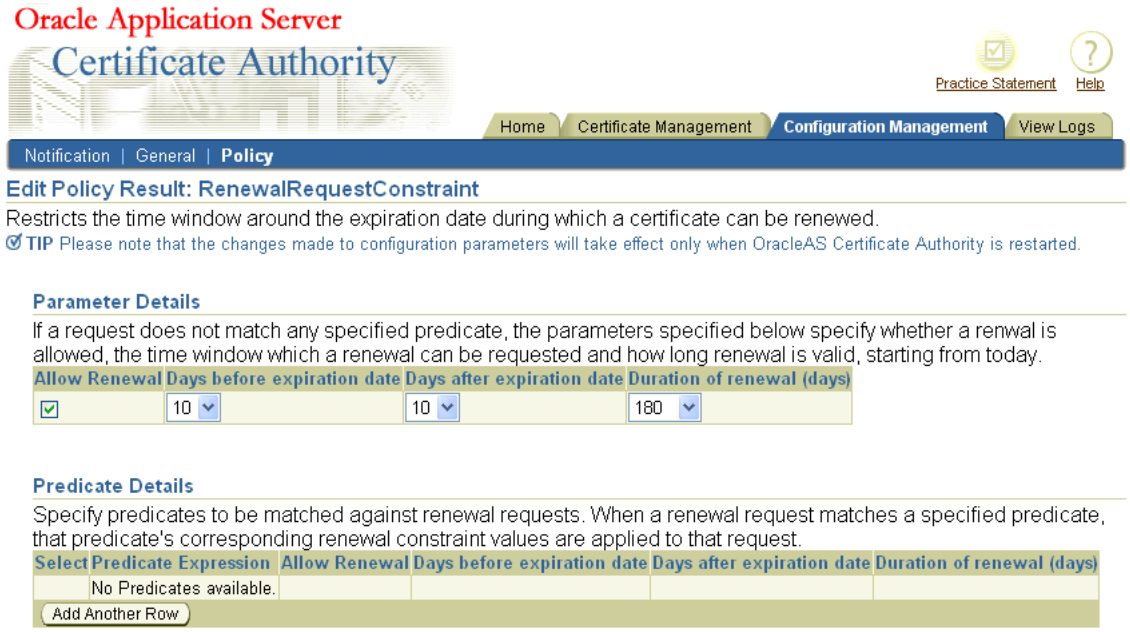
RenewalRequestConstraint

The OracleAS Certificate Authority Administrator can restrict the time window during which renewal of certificates is allowed by applying this policy to certificate renewal requests (including the administrator certificate renewal). If this policy is enabled, a user cannot renew a certificate outside the range of days specified around its expiration date. You can exclude or constrain renewal of expired certificates in your PKI setup by configuring this policy accordingly.

Table 6–7 describes the parameters of the renewal constraints policy rule.

Table 6–7 Parameters in the Renewal Constraints Policy Rule

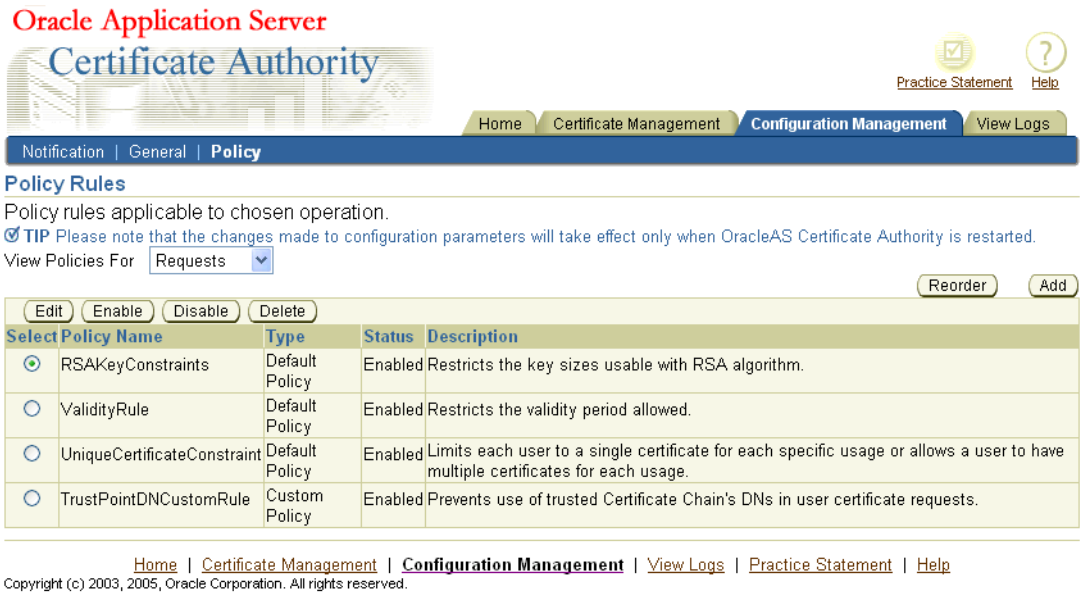
Parameter	Description
Status (Enabled/Disabled) Default=Enabled	Specifies (on the Policy Rules page) whether the rule is enabled or disabled. If you enable the rule and set the other parameters correctly, Oracle Application Server Certificate Authority verifies whether the request is made with in the specified number of days before or after its expiry by checking against the parameters renewalNotBefore and renewalNotAfter. If it succeeds, it will set the validity period to the value specified in the validityPeriod parameter. If you disable the rule, the OracleAS Certificate Authority does not verify the requested date of the certificate being renewed; it simply renews the certificate and sets the validity period to 365 days.
predicate (No defaults)	Specifies the predicate expression for this rule. If you want the rule to be applied to all certificate requests, specify '*' in this field (default). Since auto users are always of type client, ocmcert, Type predicate expression need not be used, for example, DN=="ou=ST,o=Oracle,c=US". (DN entries must be contiguous, and must be complete down to the "C=" entry, but need not necessarily start with CN. A comma must be used to separate each DN field from the next.) See "Predicates in Policy Rules".
allowRenewal Default: TRUE	Specifies whether to allow (value set to TRUE) or prevent (FALSE) renewal of certificates.
renewalNotBefore Default: 10	Specifies how many days before its expiration that a certificate can be renewed. Permissible values are 10, 15, 20, 25, or 30.
renewalNotAfter Default: 10	Specifies how many days after the expiration of a certificate it can be renewed. Permissible values are 10, 15, 20, 25, or 30
validityPeriod Default: 365 days	Specifies the validity period, in days, for renewed certificates. Permissible values: Numeric, for whatever period is desired.



All OracleAS Certificate Authority policies are managed by the OracleAS Certificate Authority administrator using the policy subtab of the administrative web interface.

Policy Sub-tab of Oracle Application Server Certificate Authority

When you first select the Policy sub-tab, Oracle Application Server Certificate Authority displays all the policy rules that can apply to certificate requests.



You can change the display to show the policy rules applicable to revocations or renewals by selecting either "Revocations" or "Renewals" from the drop-down box labeled "View Policies For". Oracle Application Server Certificate Authority then displays those policies. The policies shipped with OracleAS Certificate Authority, and the actions available to the administrator, are summarized in the following sections:

- Default Certificate Request Policies
- Default Certificate Revocation Policy
- Certificate Renewal Policy as Shipped
- Policy Actions

Policies specify the rules by which certificate requests are evaluated and by which issued certificates are renewed or revoked. You can add a policy for requests, revocations, or renewals and, if more than one policy exists, reorder the policies to alter the sequence in which they are applied. For each policy of a given type, you can view and edit its parameters and predicates, enable or disable it. Deletion of OracleAS Certificate Authority Default Policies are not allowed, but you can delete custom policies.

To add a policy, you must specify its name and description, and specify a class that you have previously added as a jar in the `$ORACLE_HOME/oca/policy` directory. (For Windows, `%ORACLE_HOME\oca\policy`.)

See Also: "Developing a Custom Policy Plug-in"

The administrator can disable any policy. Disabling a policy does not remove it from the possibility of future use, but rather resets an entry in the OracleAS Certificate Authority repository that can later be re-enabled. Deleting a policy makes it permanently unavailable (unless you later add it as if new).

Policies are enabled by an entry in the OracleAS Certificate Authority repository. Enabling a disabled policy (or one that was specified in the OracleAS Certificate Authority repository but not enabled) makes its parameters and predicates effective once again.

Policy parameters usually specify default limits or ranges that a certificate request must not violate or it will be rejected automatically. Some parameters simply enable or disable a capability or a constraint. Parameters apply to all circumstances except those specified in predicates.

Policy predicates identify specific types of certificates or requests for which the policy parameter limits, ranges, or constraints are specified to be different from the defaults for all other certificates or requests.

Changes you make to any Policy configuration parameters will take effect only after OracleAS Certificate Authority is restarted, as described in the section titled "Starting and Stopping Oracle Application Server Certificate Authority" in Chapter 4.

OracleAS Certificate Authority ships with policies that apply to certificate requests, revocations, and renewals, as discussed in the sections that follow.

The administrator can override the policy by unchecking the "apply policy" checkbox when issuing a certificate.

Default Certificate Request Policies

Certificate requests must satisfy the parameters and predicates of the policies that restrict four factors important to security. You can adjust the parameters and predicates affecting the following factors in the default policies:

- Narrow or widen the range of key sizes, and set the defaults for RSA public/private keys
- Narrow or widen the range of validity periods, and set the defaults

- Allow or disallow a user to have multiple certificates for each type of usage, that is, for signing, key encipherment, or data/email encipherment, respectively designated signing, certificate signing, or encryption (SMIME, "Secure Multipurpose Internet Mail Extensions")
- Allow or disallow the use of trusted-certificate-DNs as certificate applicants or owners.

Default Certificate Revocation Policy

The RevocationConstraintRule is an OracleAS Certificate Authority default policy shipped with Oracle Application Server Certificate Authority. You can set parameters and predicates on this policy as required, such as to allow or disallow revocation of expired certificates.

Certificate Renewal Policy as Shipped

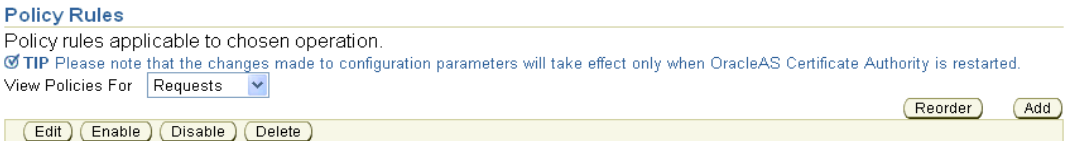
You can set the parameters and defaults for the RenewalRequestConstraint policy, which establish whether and when certificates can be renewed, and for how long. You specify the window within which renewal is to be allowed, by setting a number of days before and after the certificate's established expiration date. The default is 10 days before and after that date. You can also change the default renewal period, which is initially set at 365 days.

TrustPointDNCustomRule as Shipped

The TrustPointDNCustomRule is a custom policy, a sample plugin for an example OracleAS Certificate Authority policy, which is shipped with OracleAS Certificate Authority. This policy prevents user certificate requests from getting certificates in the name of some trusted Certificate Chain's DNs. For such plugins, you set the description, the name of the class implementing the policy, and specify in a checkbox whether this policy should be enabled.

Policy Actions

The buttons you see on the Policy Rules screen represent the actions you can take, which are described in the sections that follow: "Edit", "Enable or Disable", "Delete", "Reordering Policies", and "Adding Policies".



Edit

When you select a policy and click **Edit**, Oracle Application Server Certificate Authority displays the screen for that policy, showing its parameters and predicates as currently set. For example, the screen for the key constraints policy shows the defaults for maximum and minimum key sizes. It also shows the predicates that change those defaults for specific certificate types.

On any such page, you can choose different values for the default parameters or for the specific values associated with the existing predicates. For standard policies, you can also change those predicates by typing in the Expression text box, reorder the

predicates using the Reorder button, or add a predicate using the Add button. (See "Reordering Predicates" and "Adding Predicates".)

The Custom Policy edit screen will appear when you select a custom policy and click edit. The usual edit screen is only for default policies.

Enable or Disable

When you create a policy, you can choose to enable it. If you do, it will apply to the type of operation (request, revocation, or renewal) for which you specified it. If you do not enable it, or if you choose to disable it at some point, its parameters, defaults, and predicates will not apply to any request, revocation, or renewal.

However, a disabled policy remains available in the database. You can then later enable it at your discretion.

A deleted policy, on the other hand, is removed from the database, making it permanently unavailable unless re-entered as a wholly new policy.

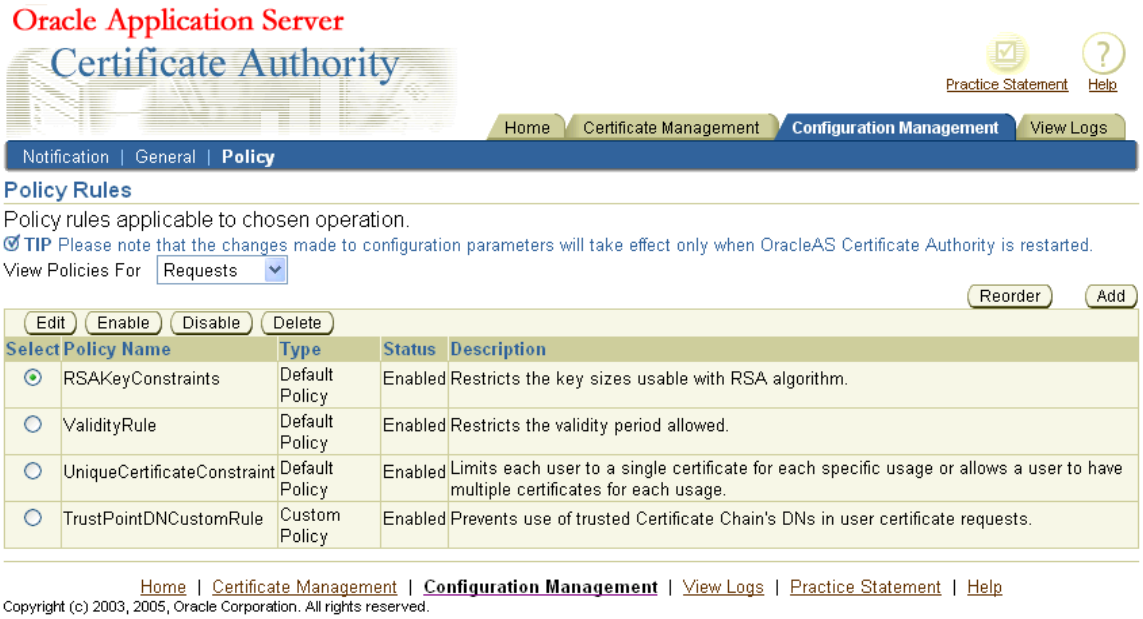
Delete

On the Policy Rules page, the default OracleAS Certificate Authority policies cannot be deleted; only Custom Policies can be deleted. If you had added a custom policy, it would appear in the list, and you could select it and click Delete.

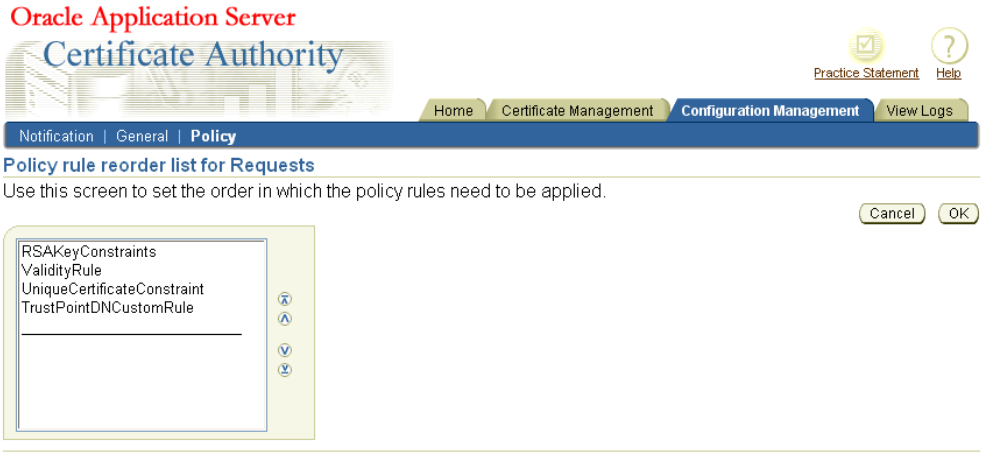
On the Edit page for a particular rule, you can select a predicate and click **Delete**. OracleAS Certificate Authority immediately removes that predicate and displays an informational message saying it has done so.

Reordering Policies

As an administrator, you can change the order in which policies are applied. For example, the default policies for certificate requests appear in the order shown on the following screen:

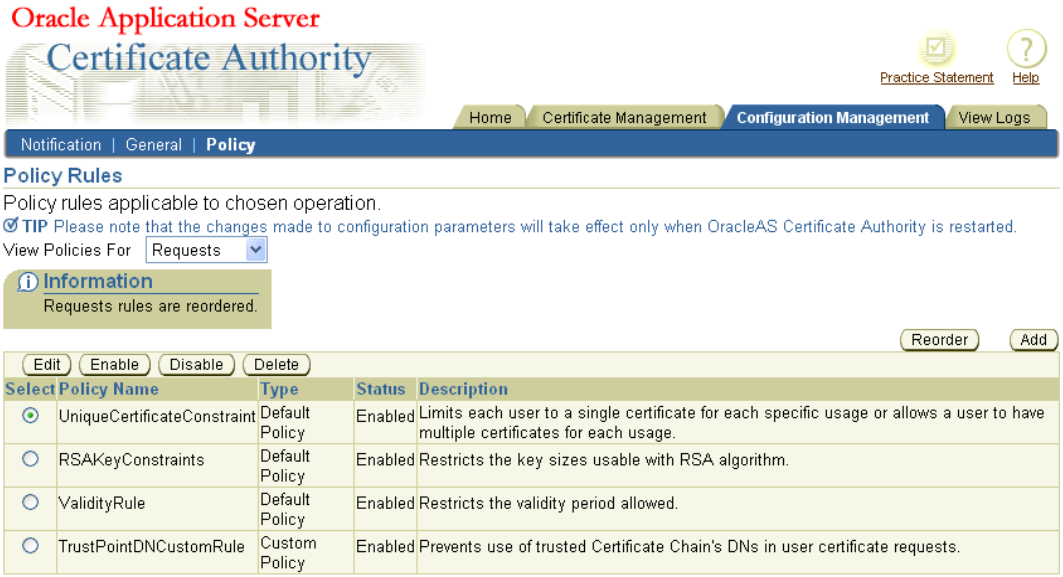


When you click **Reorder**, OracleAS Certificate Authority displays the list of existing policies. You can select and re-position them until you have your desired order, using the following screen:



To move the Unique policy up two positions, click it to select it and then click the upward-pointing button two times, creating the following screen:

When you click OK, you see that policy in the first position instead of where it had been, as shown on the following screen:

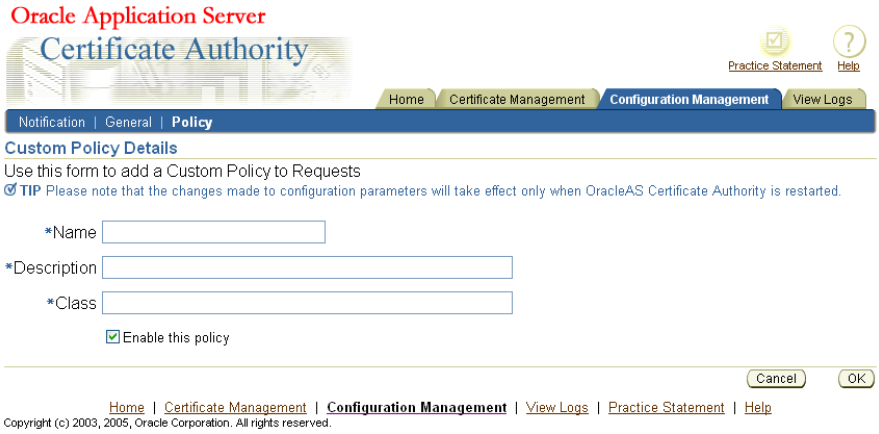


Note that OracleAS Certificate Authority displays an Information message alerting you to the change.

The predicates within a policy rule can also be reordered in a similar way. See the section titled "Reordering Predicates".

Adding Policies

On the Policy Rules page, you can click the **Add** button to add a new policy for the type of operation you were reviewing, that is, for requests, revocations, or renewals. Only custom policies can be added, as embodied in an object class that you have already defined and made available as a jar in the `$ORACLE_HOME\oca\policy` directory. OracleAS Certificate Authority displays a form for you to enter the new policy's name, description, and object class, and to specify whether it should be enabled. For more information on custom policy development, see "Developing a Custom Policy Plug-in".



See "Developing a Custom Policy Plug-in" on page 6-22 for further explanation.

You can also add a predicate, within a policy rule, to any of the default policies displayed on the edit page for the policy. (Predicates cannot be added to custom policies.) See "Adding Predicates".

Predicates in Policy Rules

Policy rules are specified and enforced according to certain conventions, as explained briefly in the section "Overview of Policy Management". This section explains the use of predicates in policy rules and supplies examples, in the following subsection:

- "Multiple Predicate Evaluation", which had the following subsections:
  - Evaluation Example for Multiple Predicates
  - One Further Example of Evaluating Multiple Predicates
  - Reordering Predicates
  - Adding Predicates

Note: Policy rules cannot be shared across request types, that is, requests for certificate issuance, revocation, or renewal.

A predicate specifies certain values and an expression used as a test of incoming certificate requests. The specified values are to be used instead of the policy's defaults if the predicate expression is matched by the corresponding elements of a certificate request. When a match occurs, the values associated with that predicate expression are used to evaluate the request's validity and set its parameters, instead of the policy's default values.

Predicates are optional, and they cannot be used in custom policies.

You can specify predicates in the web interface for a rule within a default policy. Once specified, the predicates are matched with every incoming request for the particular certificate operation the policy applies to, that is, request, revocation, or renewal.

If an incoming certificate or certificate request matches no predicate expression, or if the rule has no predicates, then the default values, ranges, or actions specified for the policy are used to evaluate the request. For example, values in the request are checked to verify they are in the correct default range specified in the policy. If they are, the request will be honored. Values that do not match the specified defaults or are not in the specified ranges cause the request to be rejected with an informational error message.

If an incoming certificate or certificate request does match a type specified in a predicate, then the defaults or ranges in the rule are not applied to that certificate or certificate request. The only values that can be applied to it are those you specify as corresponding to that predicate.

Thus, as an administrator, you can enhance a rule in a default policy and configure it for different user populations. For example, you can set a longer validity period for the "Development" department than for the "Sales" department.

The predicate expression is a logical expression. You form the expression using variables and relational operators. For example, you could set up a predicate to set different validity dates for certificates for users in different groups.

The following are valid sample predicate expressions:

```
Type==client AND DN=="ou=Sales,o=oracle,c=us"
Type==server AND DN=="o=Oracle,c=us"
```

Table 6–8 lists the logical operators used in predicate expressions.

Table 6–8 Logical Operators

Operator	Description
==	Equal to
!=	Not equal to
AND	Logical operator AND

The following rules use the delimiter ":=" to separate the name of the policy expression and its valid syntax. They show what is valid in constructing policy expressions:

```
Predicate expression := Expression | AndExpression

AndExpression := Expression AND Expression

Expression := Attribute op Value

Attribute :=  <attrib_name>

op:    ==  or    !=
```

```
Value :=  a string
```

OracleAS Certificate Authority does not support operators such as OR, <, and >. You can implement the OR logical expression by splitting the predicate into multiple predicates and specifying the same value. (The policy plug-ins and APIs support multiple predicates.) In the predicates, values can be any string enclosed in double quotes. Attribute is always specified as <attrib_name>. All predicate expressions and string values are case-insensitive. A Value in an Expression can be set to "*" to match every "attribute" under consideration, for example, type=="*" matches all the certificate types. However, using "*" with any other string to form partial-pattern string matching is not supported.

Table 6–9 describes the attributes and the values they can have.

Table 6–9 Predicate Attributes

Attributes	Variable Name	Description
type	type	Specifies the certificate type. Allowable values include the following: ■ type=="client" ■ type=="server" ■ type=="ca "
usage	usage	Specifies how the certificate will be used. Allowable values are the integers 1 through 9, in quotes, representing all the capabilities and combinations of encryption, signing, and authentication, plus code signing and certificate signing: ■ usage=="1", meaning encryption ■ usage=="2", meaning signing ■ usage=="3", meaning signing and encryption ■ usage=="4", meaning authentication ■ usage=="5", meaning authentication and encryption ■ usage=="6", meaning authentication and signing ■ usage=="7", meaning authentication, signing, and encryption ■ usage=="8", meaning code signing ■ usage=="9", meaning certificate (CA) signing
DN	DN	Specifies the distinguished name. Valid parameters include any valid partial or complete DN. (DN entries must be contiguous, and must be complete down to the "C=" entry, but need not necessarily start with CN.)

OracleAS Certificate Authority uses DNs as specified in RFC1779, with the most significant component last. For example, in the well-formed DN "cn=user31415,ou=security,ou=ST,o=Oracle,c=US", cn is the least significant component and c is the most significant one. A comma must separate each DN field from the next.

The term RDN stands for "relative distinguished name," meaning the most granular level local entry name that needs no further qualification to address an entry uniquely. If an RDN appears multiple times, then the least significant RDN, specified first, is understood to be a child of the RDN occurring next. In the earlier example, since "ou=security" appears before "ou=ST", "security" is understood as a sub-division under "ST" division.

A DN specified in the predicate can start at any RDN but should complete at the root. For example, "ou=ST,o=Oracle,c=US" is a valid partial DN that can be specified, whereas "ou=ST,o=Oracle" is an invalid partial DN as it stops at "o=Oracle" and does not contain the root (that is, "c=US").

To support the big-endian order, where the most significant component is first, OracleAS Certificate Authority internally converts it to little-endian order before DN matching is done, for policy evaluations only.

When DN components are matched against a DN expression mentioned in a predicate expression, the following rules are applied:

The predicate matches the DN if the whole predicate is a last part of the DN.

For example, if the predicate expression is

```
DN=="ou=ST,o=Oracle,c=US"
```

then it would match all of the following DNs:

```
"cn=user31415,ou=ST,o=Oracle,c=US"
```

```
"cn=quser2787,ou=security,ou=ST,o=Oracle,c=US"
```

```
"cn=kuser987,ou=security, ou=DAS,ou=ST,o=Oracle,c=US"
```

However, the predicate expression fails to match the following DNs:

```
"cn=user31415,ou=DAS,ou=ST,o=Oracle,c=IN"
```

```
 "cn=quser2787,ou=ST, ou=pki, o=Oracle,c=US"
```

```
  "cn=kuser987,ou=ST,o=Oracle, st=CA,c=US"
```

Multiple Predicate Evaluation

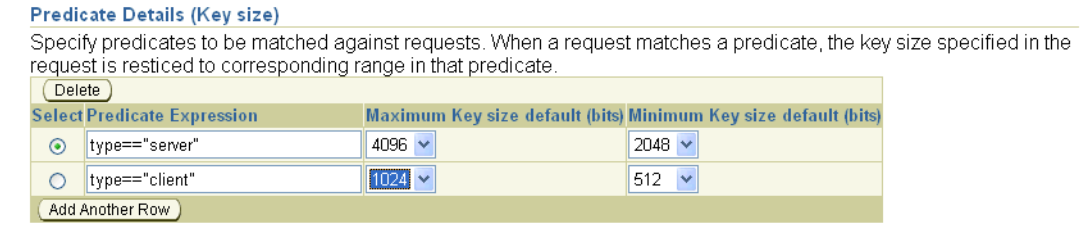
A policy rule can have more than one predicate. When the policy rule has multiple predicates, evaluation begins by comparing the first predicate expression against the incoming certificate request object. If it matches, the rule is applied. If not, then evaluation compares the next predicate expression against that request. This procedure continues until a predicate matches the certificate request object or, if no predicates match, the policy rule's default values are applied.

No attempt is made to find the best match: the first match that occurs is used. The administrator is responsible for specifying the order of predicates in the manner most appropriate to the organization.

One criterion is to place, at the top of the rules, those predicate expressions targeted for specific matches and least-significant RDNs, so they will be evaluated first.

Evaluation Example for Multiple Predicates

The following example demonstrates how a rule evaluates multiple predicates. In this example, the policy rule is about the key sizes used by the RSA rule. The rule has two predicate expressions, about server certificates and client certificates, specifying corresponding minimum and maximum key sizes. If an incoming server or client certificate request specifies a key size outside the range specified for its corresponding predicate, the rule will reject it.



If neither predicate expression matches the incoming certificate request, then the rule compares the requested key size with the minimum and maximum specified as defaults. If the requested key size is outside this range, the request is rejected; otherwise, it is approved. (The factual default range as shipped is 512 to 4096. If the administrator has chosen Microsoft Strong Cryptographic Provider, the default size for the key that is generated is normally 1024. However, in some Windows environments the "Strong" choice creates 4096-bit keys.)

One Further Example of Evaluating Multiple Predicates

Evaluation of multiple predicates can be subtle. They are applied in the top-down order in which they are listed on the Edit page of the Configuration Management tab in the Oracle Application Server Certificate Authority web interface. The sequence is important.

Suppose the first predicate listed in a policy specifies `Type=="client"` and `OU=="Oracle"` and `CN=="Clay"`, and then sets the keylength to 2048.

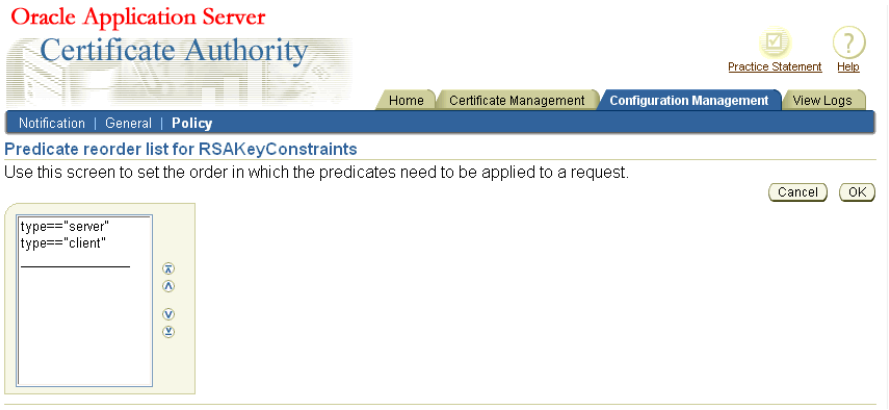
Then, suppose a later predicate in that same policy specifies `Type=="client"` and `OU=="Oracle"`, setting the keylength to 512.

Then only client requests from Clay will have the keylength set to 2048; all other Oracle client requests will have it set to 512.

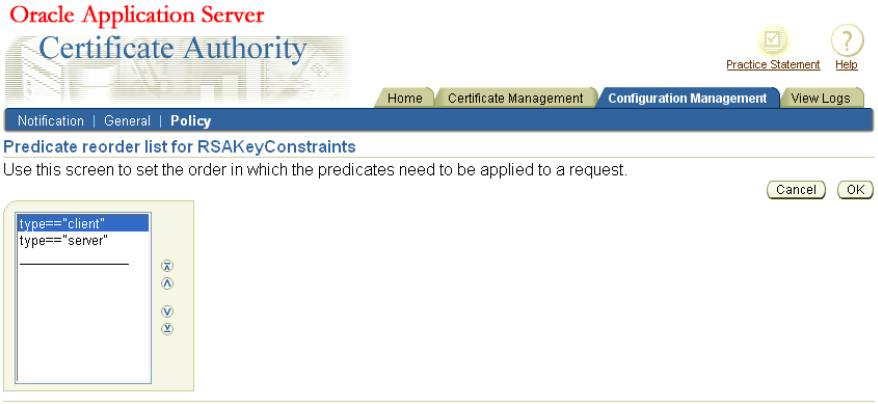
However, if the order is reversed, so that the more general predicate is earlier in the policy, even Clay will have a keylength of 512. The more specific predicate will never be encountered, since only the one earlier in the sequence, the more general one, will be acted on for this policy.

Reordering Predicates

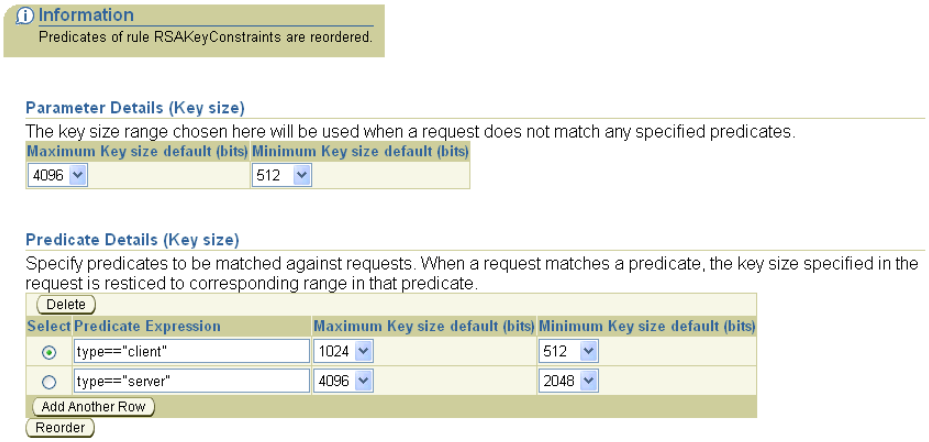
You can reorder predicates in a manner similar to reordering policies, as described in "Reordering Policies" on page 6-13. If you click **Reorder** on a page displaying predicates, like the one shown in the "Evaluation Example for Multiple Predicates" section, Oracle Application Server Certificate Authority displays a screen of the following type:



Select the predicate you wish to move, by clicking it, and then click one of the reordering arrowhead buttons: the predicate will move in the direction you chose. For example, if you reversed the order of the predicates in the "Evaluation Example for Multiple Predicates", you would see the following screen:



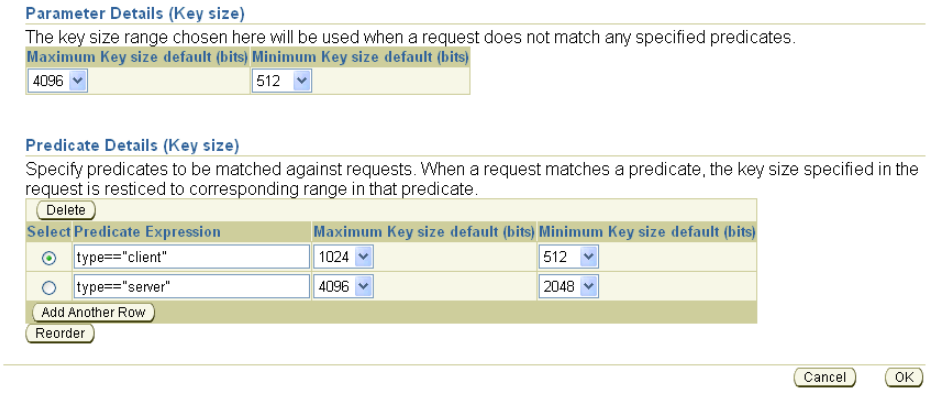
Clicking **OK** would then make that the predicate order for the rule, as shown here:



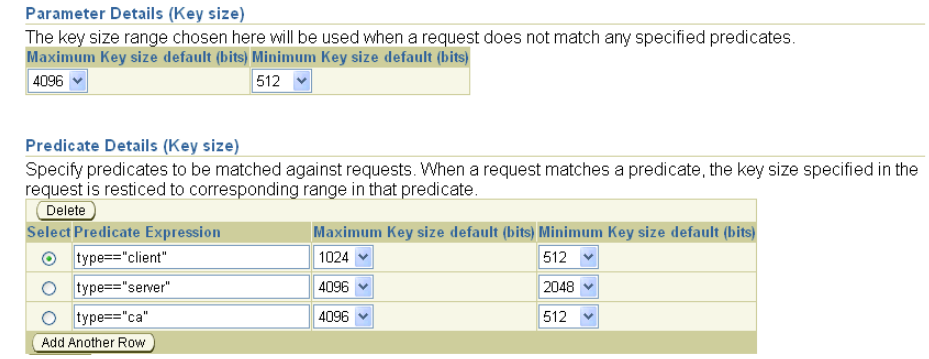
Notice that OracleAS Certificate Authority also displays an Information message acknowledging the changed order.

Adding Predicates

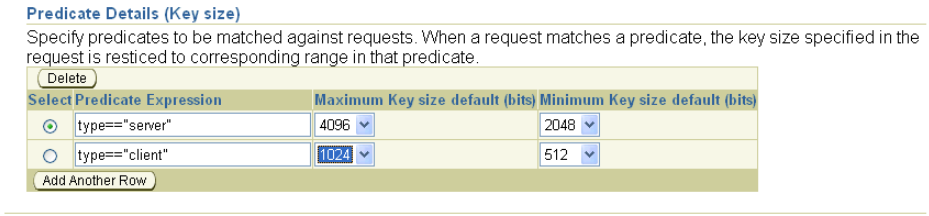
You can add a predicate by clicking **Add Another Row** on a page displaying predicates. OracleAS Certificate Authority displays a blank row for you to fill:



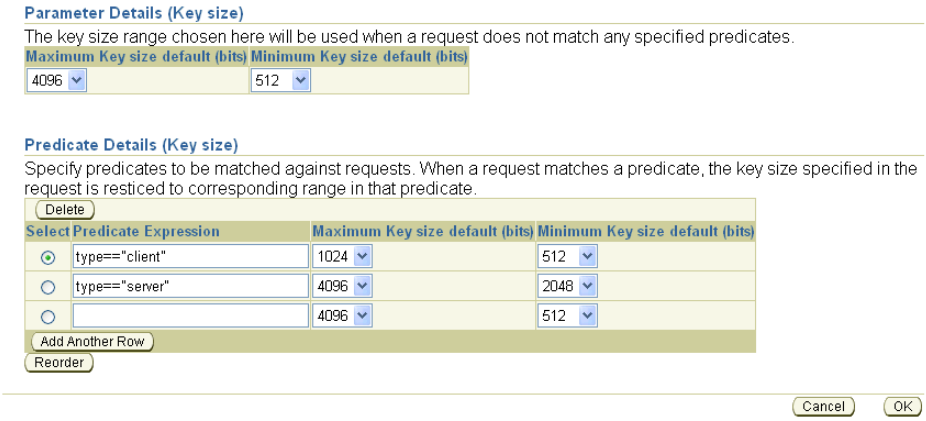
If you fill in the blank row with a valid predicate, as shown in this screen, it will be accepted when you press **OK** (which also returns you to the main policy page).



On the Edit page for a specific policy, you can add a predicate by clicking **Add Another Row**. An example of a predicate is requiring that requests for a server certificate use a higher range of key lengths than required of end-user certificate requests, as in this screen:



When you click **Add Another Row**, OracleAS Certificate Authority displays an empty additional predicate row, where you can type your new predicate into the Predicate Expression box. You also specify the capability or default parameter range to be used when the predicate is matched:



If you specify a predicate that is invalid or already present for this rule, an error message appears.

When you have specified the predicate to your satisfaction, click **OK**. OCA displays the original page for this rule with your new predicate row added at the bottom.

Developing a Custom Policy Plug-in

The default policy plug-ins shipped with OracleAS Certificate Authority are generic. To enhance the policy structure for specific organizational requirements, an administrator can write a custom plug-in using the framework OracleAS Certificate Authority provides. This framework includes APIs to get information about certificates and certificate requests, and a few generic functions. To implement a custom plug-in, an administrator must write a Java class and register it with OracleAS Certificate Authority, which is also called "adding a policy."

The following situations are appropriate examples of goals for developing custom plug-ins to handle:

- To use an additional corporate account repository to validate user requests
- To set additional fields based on other user repositories

The APIs provided by OracleAS Certificate Authority enable the administrator's custom plug-in to acquire request parameters and the attributes of certificates and certificate requests.

> **See Also:** The Javadoc accompanying Oracle Application Server Certificate Authority

The following subsections describe tools and examples to aid an administrator in developing custom plug-ins if the organization requires them:

- What Processing Does a Policy Do?
- Steps in Creating a New Policy Plug-in
- An Example of a Custom Policy Plug-in
- Generic Error Messages

What Processing Does a Policy Do?

A custom plug-in can be written by implementing OCACustomPolicyPlugin interface. OCAPolicyRequest object, which is passed to the 'enforce' method of this interface, has

all the essential attributes (of the certificate or certificate request) and their values set. The custom plug-in can read these objects to get or set attributes of the certificate request or certificate.

The following steps are involved in custom policy plug-in processing:

Table 6–10 Steps in Custom Policy Plug-in Processing

Step	Results
Enforce method of OracleAS Certificate Authority custom plug-in receives OCAPolicyRequest from the policy processor	Automatic retrieval of the objects needed to get the actual parameter values set during the enrollment, renewal, or revocation requests. These parameters are the DN, validity period, serial number, and so on.
The plug-in checks the retrieved parameters from OCAPolicyRequest with the parameter values expected by the plug-in.	If the policy check succeeds, it sets the plug-in result using setplug-inResult method and return TRUE to the policy processor. Otherwise it sets an error using setError() and returns FALSE to the policy processor.

Steps in Creating a New Policy Plug-in

Use these steps to create a new policy plug-in:

1. Write a Java class that implements the `OCACustomPlugin` interface, using the sample implementation shown in the next section as a guide.
2. Save the java class implemented in step 1 and compile after adding `$ORACLE_HOME/oca/lib/oca-1_3.jar` to the java `CLASSPATH` and obtaining the class file.
3. Use the **jar** utility to jar the class file.
   a. For example, the code in the previous section would be `jar`'d and kept in `example.jar`.

      To jar the class, use the `jar` utility available under the `$ORACLE_HOME/jdk/bin` directory.

      * To create `example.jar`, execute:

        ```
        $ORACLE_HOME/jdk/bin/jar cvf example.jar oca
        ```

      * where `example.jar` is the jar file name and OracleAS Certificate Authority is the package directory that contains `custom/policy/plugin/examplePlugin.class`

   b. If you then were to execute '`jar tvf example.jar`', you would see the `examplePlugin.class` file under the directory structure `oca/custom/policy/plugin`.
4. Place this `jar` file into the `$ORACLE_HOME/oca/policy` directory. (For Windows platforms, the slashes become backslashes):

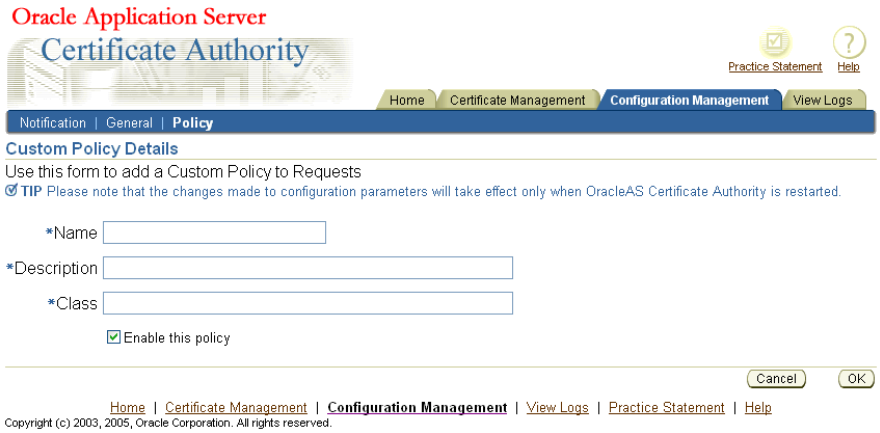
   ```
   $ORACLE_HOME\oca\policy\
   ```

   This directory is pre-created by Oracle Application Server Certificate Authority.
5. Stop OracleAS Certificate Authority, OracleAS Certificate Authority's OC4J, and OHS. Use the following commands in `ORACLE_HOME`:

   ```
   $ORACLE_HOME/oca/bin/ocactl stop
   $ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=oca
   $ORACLE_HOME/opmn/bin/opmnctl stopproc type=ohs
   ```

6. Start OHS, OracleAS Certificate Authority's OC4J, and OracleAS Certificate Authority, in that order. Use these similar commands:

```
$ORACLE_HOME/opmn/bin/opmnctl startproc type=ohs
$ORACLE_HOME/opmn/bin/opmnctl startproc type= oc4j instancename=oca
$ORACLE_HOME/oca/bin/ocactl start
```



7. Use the OracleAS Certificate Authority Administrator's web interface to add your custom policy to define your new rule.

   That is, navigate to the Policy Rules page within the Policy subtab of Configuration Management, click the Add button, and fill in the fields. You supply the name, description, and class for your custom policy, click the enable checkbox to enable it, and then click OK.

8. Restart OracleAS Certificate Authority, as described in the section titled "Starting and Stopping Oracle Application Server Certificate Authority" in Chapter 4, "Introduction to Administration and Certificate Management". Only after OracleAS Certificate Authority is restarted will the new jar be found and recognized, and the rule be able to come into effect. After this step, the custom policy will be applied to requests for certificates, renewals, or revocations, depending on which section was modified by adding this plug-in.

Rules for Custom Policies

While developing custom policy plug-ins, keep the following ground rules in mind:

- There should be no spaces in the policy name
- Make sure the class exists in the specified directory
- The class should implement a particular interface

An Example of a Custom Policy Plug-in

You need to implement OCACustomPolicyplugin interface to write a custom plug-in.

The first step in supplying your own policy plug-in is to create a new Java class.

This section shows you an example of a custom policy plug-in that ensures that the country code in the certificate request is not US.

```
1:    package oca.custom.policy.plugin;
2:    import oracle.security.oca.exception.OCMException;
3:    import oracle.security.oca.policy.custom.OCACustomPolicyplugin;
4:    import oracle.security.oca.policy.OCAPolicyRequest;
5:    import oracle.security.oca.policy.OCMPolicyConstants;
6:    public class PolicyCustomPlugin implements OCACustomPolicyPlugin
7:    {                                 // ends at line 35
8:       public boolean enforce (OCAPolicyRequest policyRequest)
9:       {                          // ends at line 34
10:      // Add the functionality here.
11:      // Assume the plug-in should reject all requests with country code as US.
12:               if (!policyRequest.getCountry().equals("US"))
13:               {
14:                      //Plug-in check succeeded. Country ID in request is not
US.
15:                        //Hence return true.
16:                        return true;
17:               }
18:               else
19:               {                   // ends at line 33
20:                       //Plug-in check failed: Country ID in request is US. Set
error and return false.
21:    try
22:    {
23:
policyRequest.setError("PolicyCustomPlugin",OCMPolicyConstants.POLICY_ERROR,
                                                    "Country ID cannot be US.");
24:                     //The first parameter is the plug-in name.
25:                    //The second parameter is the status, which is an ERROR.
26:                   //The third parameter is the Message to be displayed.
27:     }
28:   catch(OCMException e)
29:  {
30:     //enter exception handling here
31:   }
32:                        return false;
33:               }
34:        }
35:    }
```

In this example, line 1 is the package to which this custom policy plug-in belongs. The custom policy plug-in can belong to any package other than a package that starts with 'oracle.security.oca'.

Lines 2 through 5 import the class files required. The Javadoc API documentation contains the details of these files.

Line 6 implements the OCACustomPolicyPlugin interface. The custom policy interface must be implemented by all custom plug-ins. The interface that OracleAS Certificate Authority gives belongs to the package oracle.security.oca.policy.custom and is available in `$ORACLE_HOME/oca/lib/oca-1_3.jar`.

Line 8 implements the method that will contain the functionality of this plug-in. So when the policy processor invokes this plug-in it will invoke the 'enforce' method.

Line 9 through 28 begins the functionality for this plug-in

Line 12 checks if the country code is not US. The API documentation accompanying OracleAS Certificate Authority contains the details of the methods that can be used on policyRequest.

Line 16 returns success to the policy processor.

Line 18 introduces the handling for the error condition, exercised when the country code in the request is US.

Line 23 sets an error code into policyRequest. This error code is read by the policy processor and is rendered to the screen. You can see something similar when you get a new OracleAS Single Sign-On User certificate and immediately try to renew this certificate. The Renewal plug-in will generate an error.

Line 30 should be replaced by code to handle exceptions.

Line 32 returns an error status to the policy processor, indicating that the request failed the policy check and therefore will not be processed.

Generic Error Messages

Following are the generic error messages, and their associated constants, that can be set if any error is found while applying the policy. These messages are translated into the languages supported by OracleAS Certificate Authority, such as the following three:

- Invalid validity period

  "OCA_POLICY_INVALID_VALIDITY"

- Requested validity period exceeds validity of the CA certificate

  "OCA_POLICY_INVALID_VALIDITY_CA"

- Invalid Distinguished Name

  "OCA_POLICY_INVALID_DN"

For example, in the earlier custom policy example, if line 13 is changed to read

```
13:
policyRequest.setError("examplePlug-in",OCMPolicyConstants.POLIC
Y_ERROR,
OCAPolicyMessage.OCA_POLICY_INVALID_DN);
```

then the output would show the "Invalid Distinguished Name" error.

> **See Also:** The Javadoc provided with the other documentation for descriptions of the classes and methods provided in OracleAS Certificate Authority Custom plug-in, and the constants available for your use.

> **Note:** The generic error messages supported by OracleAS Certificate Authority are translated into the languages supported by OracleAS Certificate Authority, so they are available to use in custom plug-ins as well. By using these constants, these error messages are available to be rendered in any of the languages supported by OracleAS Certificate Authority.
>
> If these messages are not used, then any valid java string can be used. However, these java strings will not have been translated, and so they will be rendered simply as they are supplied.

7

OracleAS Certificate Authority Administration: Advanced Topics

This chapter provides additional context and detail for Oracle Application Server Certificate Authority administrative features, for high-availability features, and for backup and recovery procedures in the following sections:

Wallet Operations for OracleAS Certificate Authority

Wallets are containers for certificates and trusted authorities' certificates. OracleAS Certificate Authority uses wallets for secure storage and access regarding these vital elements. When certificates, trusted authorities, or passwords change, the administrator must take action to enable their use in a consistent and secure manner. The following sections describe such actions:

Regenerating the CA Signing Wallet

> **Warning!:** This operation regenerates the CA signing certificate and replaces the existing CA certificate, invalidating all the certificates issued by the existing CA. For this reason, you must be extremely cautious before attempting the operation. Do not use it unless you are prepared to lose the existing CA certificate and all the certificates the CA issued.

Installation of OracleAS Certificate Authority as a root certificate authority (CA) also creates the CA signing certificate and the CA SSL wallet. The CA SMIME wallet is not generated automatically, but rather must be generated by the administrator. If the CA key is somehow compromised, this certificate and wallet can be regenerated using the administrative command line tool, `ocactl`, as described in the next section.

The new CA certificate and private key will be stored in the OracleAS Certificate Authority repository. The private key is encrypted by the same password that was requested and established during installation of the original CA. The former CA signing certificate entry and all other certificates issued by that former CA signing certificate will become invalid.

Other critical wallets, like CA SSL and CA SMIME, also need to be regenerated, and regenerating these does require a new password. After regeneration of the CA signing wallet, a CRL issued by the old CA will not be useful.

Example of the command to generate the CA signing wallet:

```
ocactl generatewallet -type CA
```

OracleAS Certificate Authority needs to be stopped to execute this command, which can take a few minutes to complete. To restart OCA, see the section titled "Starting and Stopping Oracle Application Server Certificate Authority" in Chapter 4, "Introduction to Administration and Certificate Management".

Regenerating the CA SSL and CA S/MIME Wallets

This section explains how to regenerate the CA SSL and CA S/MIME wallets, respectively.

The CA SSL Wallet

The CA SSL wallet is generated during installation and is used to enable the Oracle Application Server Certificate Authority engine to listen in HTTPS mode. In certain circumstances, you must regenerate the CA SSL and CA S/MIME wallets in order to establish secure communications with the OracleAS Certificate Authority server. These circumstances include a wallet becoming compromised or corrupted, or the CA Signing wallet being regenerated, or a new Sub CA certificate being installed.

Here is an example of the command to generate the CA SSL wallet:

```
ocactl generatewallet -type CASSL
```

OracleAS Certificate Authority, OracleAS Certificate Authority's OC4J, and OHS all need to be stopped to execute this command. After this command executes, restart OHS, OC4J, and OracleAS Certificate Authority, in that order.

This wallet is stored as ewallet.p12 (PKCS#12) under the directory `$ORACLE_HOME/oca/wallet/ssl`, encrypted by the password that was provided during its

generation. This command also generates CA SSL wallet in OracleAS Single Sign-On format and stores it as cwallet.sso at `$ORACLE_HOME/oca/wallet/ssl`.

The advantage to using cwallet.sso is that HTTP Server can be brought up in SSL mode without requiring the Oracle HTTP Server administrator to supply the wallet password. This password is normally requested when HTTP Server starts up in SSL mode, using a PKCS#12 wallet.

The OracleAS Single Sign-On server-format wallet is obfuscated to discourage users from visually opening the file and extracting the keys. However, the operating system file permissions are relied upon to protect it, since it is created with owner-only permissions. The next startup of OracleAS Certificate Authority instance in OPMN will use this wallet for SSL server authentication. (After CA SSL/OracleAS Single Sign-On wallet generation, the OPMN `stopall` and `startall` commands are required.)

The CA S/MIME Wallet

The CA S/MIME wallet enables OracleAS Certificate Authority to sign alerts and notification messages. You must generate it before selecting "Send SMIME E-Mails" in the Notification subtab of Configuration Management in the OracleAS Certificate Authority Administration pages. Once generated, this wallet resides in the OracleAS Certificate Authority database repository.

> **See Also:** "Mail Details" in Chapter 5, "Configuring Oracle Application Server Certificate Authority"

If this S/MIME wallet is compromised or corrupted, or when the CA signing wallet is regenerated, you must regenerate the CA S/MIME wallet. This wallet is encrypted by the administrator's password, which is required to execute the command generating the wallet. You will also be asked for the administrator's Distinguished Name and email address.

Here is an example of the command to generate the CA S/MIME wallet:

```
ocactl generatewallet -type CASMIME
```

The following steps generate and use the CA S/MIME wallet:

1. Go to **Configuration Management** -> **Notification**. Select the **Send SMIME E-Mails** option.
2. Stop OracleAS Certificate Authority using the command

   ```
   $ORACLE_HOME/oca/bin/ocactl stop
   ```

3. Generate the CA S/MIME wallet using the command

   ```
   ocactl generatewallet -type CASMIME
   ```

4. Start OracleAS Certificate Authority using the command:

   ```
   $ORACLE_HOME/oca/bin/ocactl start
   ```

After regeneration of the CA S/MIME wallet, the old CA S/MIME wallet will not be of any use. The new CA S/MIME wallet is used to sign alert and notification messages.

Renewing Critical Wallets

When a certificate is going to expire, renewal will be required. CA Signing, CA SSL, and CA S/MIME wallets can be renewed using `ocactl`, the administrative command line tool. During the execution of the `renewcert` command, `ocactl` will prompt for the new validity period, taking the input as the number of days for which the certificate is to be renewed.

When the CA signing certificate is renewed, a new certificate with new validity period is created and stored in OracleAS Certificate Authority's metadata repository.

When the CA SSL wallet is renewed, the old wallet ewallet.p12 at `$ORACLE_HOME/oca/wallet/ssl/` will be overwritten with the renewed wallet. Renewal of the CA SSL wallet also overwrites the cwallet.sso at `$ORACLE_HOME/oca/wallet/ssl/`.

When the CA S/MIME wallet is renewed, the new wallet overwrites the old CA SMIME wallet in the database repository.

Example to renew CA signing wallet:

```
ocactl renewcert -type CA
```

A renewed CA S/MIME wallet can be used simply by restarting OracleAS Certificate Authority. The renewed CA and CA SSL wallets take effect only after OHS, OracleAS Certificate Authority's OC4J, and OracleAS Certificate Authority are restarted, in that order, as described in the section titled "Starting and Stopping Oracle Application Server Certificate Authority" in Chapter 4, "Introduction to Administration and Certificate Management".

Changing Passwords

After installation, you can change any of the following passwords: CA, CA SSL, CA S/MIME, or DB, by stopping OracleAS Certificate Authority, issuing the `ocactl setpasswd` command, and then restarting OracleAS Certificate Authority.

> **See Also:** Appendix A, "Command-Line Administration" for detailed descriptions of using `ocactl`.

> **Note:** The OracleAS Certificate Authority schema password can be changed only by running this command: `ocactl setpasswd -type DB`. It should not be changed by going directly to the database, for example, by using sqlplus, because OracleAS Certificate Authority will stop working if that is done.

The changes resulting from executing these commands take effect after the next start of OracleAS Certificate Authority. Each use of `ocactl` requires the OracleAS Certificate Authority administrator password. Once this is authenticated, the command requests the new password for the role type specified in the command, which then replaces the one in the password store. The results are again encrypted using the latest OracleAS Certificate Authority administrator password.

Example to change OracleAS Certificate Authority repository password:

```
ocactl setpasswd -type DB
```

Note: If the CA SSL wallet password is changed, you must restart OHS, OracleAS Certificate Authority's OC4J, and OracleAS Certificate Authority, in that order, except in the default install scenario wherein OracleAS Certificate Authority uses cwallet.sso: in that case, OHS need not be restarted.

Configuration Operations for OracleAS Certificate Authority

The OracleAS Certificate Authority administrator configures it to meet the needs of the site using it. Some of these operations are done through the web interface. Others require using command line tools such as `ocactl`, the OracleAS Certificate Authority administrative command line tool, and others that control components on which OracleAS Certificate Authority relies. These configuration operations and the actions the administrator must take are described in the following sections:

- Configuring Oracle HTTP Server to Use a Third Party SSL Wallet
- Revoking a Certificate Authority Certificate
- Revoking the OracleAS Certificate Authority Web Administrator's Certificate
- Configuring Globalization Support for Screens

Configuring Oracle HTTP Server to Use a Third Party SSL Wallet

When OracleAS Certificate Authority is installed, it is automatically configured in SSL mode. Browsers will warn that this site is not trusted until you install the CA certificate. (You can either explicitly install the CA or edit the CA entry in the browser.) To avoid this warning, the OracleAS Certificate Authority administrator can get an SSL certificate for the OracleAS Certificate Authority server from a well-known CA like Verisign.

The **convertwallet** command is used to convert such an SSL Server wallet (`ewallet.p12`, in PKCS#12 format) into a wallet in the OracleAS Single Sign-On format, with file name `cwallet.sso`. The advantage to using `cwallet.sso` is that HTTP Server can be brought up in SSL mode without requiring you to supply the wallet password. This password is usually requested when HTTP Server starts up in SSL mode, using a PKCS#12 wallet. The OracleAS Single Sign-On server-format wallet is encrypted to discourage users from visually opening the file and extracting the keys. However, the operating system file permissions are relied upon to protect it, since it is created with owner-only permissions. Thus the convertwallet command enables the OracleAS Single Sign-On server to bring up the web server in SSL mode automatically, without asking a human for the wallet password.

To install a wallet from a well-known CA, do the following:

1. Shut down OracleAS Certificate Authority, OCA's OC4J, and OHS.
2. Back up wallets in `$ORACLE_HOME/oca/wallet/ssl`.
3. Using Oracle Wallet Manager, create a complete SSL server wallet:
   a. Request an SSL certificate.
   b. Install the certificate of the third-party CA that issued the server certificate.
   c. Install your requested server certificate.
4. Using OWM, import the current OracleAS Certificate Authority CA's certificate as a trust point into this wallet.

See Also: *Oracle Advanced Security Administrator's Guide*

5. Save the wallet at `$ORACLE_HOME/oca/wallet/ssl`.

   Now OCA-issued certificates can be trusted as client certificates against this wallet as the CA SSL server's certificate.

6. Copy the wallet created from the third-party CA (in PKCS#12 format) to `$ORACLE_HOME/oca/wallet/ssl/ewallet.p12`.

7. Run `convertwallet -format SSO`.

8. Start OHS, OracleAS Certificate Authority's OC4J, and OracleAS Certificate Authority, in that order.

Revoking a Certificate Authority Certificate

Revoking a CA signing certificate is a very drastic operation, which will make OracleAS Certificate Authority installation non-functional and invalidate the certificates already issued. This operation, revocation, should only be done when the CA key is compromised, so that you can install a new certificate authority.

Using a sub-CA reduces the risk and cost. Hierarchical CA structure enables normal operations to be conducted by the sub-CA while the root CA is especially protected, perhaps being off-line in a highly secure location. In this way, even if an online subordinate CA is compromised, it can be revoked and a new sub-CA created to replace it. All earlier operations can continue using certificates as issued.

However, if the root CA is compromised, a completely new infrastructure needs to be established, and all applications relying on it need to be updated.

For these reasons, Oracle recommends using a hierarchy of CA's, with special protection for the root CA.

The revokecert command enables you to revoke a root certificate authority certificate or an OracleAS Certificate Authority Administrators certificate. It can only be used when OracleAS Certificate Authority services are not running. Revoking a root certificate authority certificate is required before installing a new CA signing for ongoing OracleAS Certificate Authority operations.

When you intend to install a new CA, first revoke all certificates issued by the existing CA, and update the Certificate Revocation List. This step is necessary because until the new CA signing certificate is generated, all the old certificates signed by the old CA would be marked as Invalid in the OracleAS Certificate Authority repository.

Then use `revokecert` to revoke the old CA signing wallet, giving your reason as a parameter. Once the CA signing certificate is revoked, all certificates issued by that CA would be in an inconsistent state, had you not revoked them already.

Once the OracleAS Certificate Authority administrator certificate is revoked, the administrator cannot access any administrative functions on the web until he gets a new certificate. When he opens the Administration home page, it will require a new enrollment to get a new Administrators certificate.

Example of the command to revoke CA certificate when its key is compromised:

```
ocactl revokecert -type CA -reason KEY_COMPROMISE
```

Steps to be followed to revoke CA certificate and restart OracleAS Certificate Authority:

1. Stop OracleAS Certificate Authority. Use the command

```
$ORACLE_HOME/oca/bin/ocactl stop
```

2. Revoke the CA signing wallet using the command shown earlier.
3. Regenerate the CA SSL wallet.
4. Start OracleAS Certificate Authority. Use the following command:

```
$ORACLE_HOME/oca/bin/ocactl start
```

Revoking the OracleAS Certificate Authority Web Administrator's Certificate

You may in future need to replace the administrator's certificate. Reasons could include the password to your private key being lost, the private key somehow being compromised or stolen, or the administrator role being given to someone new. This operation, revocation, should only be done when the Web administrator key is compromised, so that you can enroll new OracleAS Certificate Authority web administrator.

To replace the administrator certificate, you must stop the server, revoke the current administrator's certificate, and restart the server. These tasks are performed by using the server command-line tool `opmnctl` and the OracleAS Certificate Authority command-line tool `ocactl`, which requires the OracleAS Certificate Authority Administrator password.

Once the OracleAS Certificate Authority administrator certificate is revoked, the administrator cannot access any administrative functions on the web until he gets a new certificate. When he opens the Administration home page, it will require a new enrollment to get a new Administrator's certificate.

The administrator then navigates to the OracleAS Certificate Authority web page and fills in the OracleAS Certificate Authority Web administrator enrollment form.

Example of the command to revoke Web Administrator's wallet when its key is compromised:

```
ocactl revokecert -type WEBADMIN -reason KEY_COMPROMISE
```

Steps to be followed to revoke Web Administrator's certificate and restart OCA:

1. Stop OracleAS Certificate Authority. Use the command

```
$ORACLE_HOME/oca/bin/ocactl stop
```

2. Revoke the Web Administrator's certificate using the command

```
ocactl revokecert -type WEBADMIN -reason <REASON>
```

3. Start OracleAS Certificate Authority. Use the following command:

```
$ORACLE_HOME/oca/bin/ocactl start
```

Note: Having revoked the webadmin certificate, you cannot use it to log in to the web interface for OracleAS Certificate Authority. You must remove it from the browser certificate store before opening the OracleAS Certificate Authority web interface. Then you will be able to display that interface and enroll anew as the administrator.

Configuring Globalization Support for Screens

The administrative and user screens for OracleAS Certificate Authority can appear in the language of the client or of the server, under the following conditions:

1. The Database Character Set must be UTF8.
2. The UI and online help of OracleAS Certificate Authority are rendered in the locale of the client, as are dates. (Times are rendered in the server time zone.) If the client locale is not supported, the screens are rendered in the server locale. If the server locale is also not among the languages supported by OracleAS Certificate Authority, then English is the language used.
3. The practice statement is rendered in the locale in which the Administrator edits the practice statement irrespective of the client locale.
4. ocactl provides globalization support depending on the server locale. If the server locale is anything other than the OracleAS Certificate Authority supported languages, display is in English.
5. In every locale, the actual ocactl commands are themselves in English.
6. Informational messages, such as alerts, notifications, or error messages, are displayed in the language of the server locale, not in the client locale if that is different from the server locale. For example, if OracleAS Certificate Authority were installed on a server whose locale is English, and a Japanese client submits a request, the notification will be in English.

   If you use templates for customizing alerts or notifications, as described in the next section, the language in which you edit those templates is used. It is advisable to edit the templates in the language of the server, because the message body is encoded in the language of the server locale.

   If you do not use templates, then all alerts and notifications will appear in the language of the server locale.

Performance Tuning for OracleAS Certificate Authority

You can enhance the performance of your OracleAS Certificate Authority instance by setting several customizable parameters for the database, Single Sign-On, and memory, as described in these sections:

- Tuning Database Connections
- Tuning Interactions with OracleAS Single Sign-On
- Tuning Maximum Memory

Tuning Database Connections

Table 7–1 Tuning Database Usage

Method	Topic Reference
Make the database pool size the number of concurrent users that you are expecting (default 20).	"Database Settings" on page 5-10
Make the database pool scheme `dynamic`. (default)	"Database Settings" on page 5-10

Each connection in the connection pool uses up a database dedicated process. Therefore, if you adjust these sizes, you need to ensure that the database parameter named PROCESSES is greater than the sum of the following five numbers:

- Size of the OracleAS Certificate Authority pool
- Number assigned for Single Sign-On maximum connections in pool
- Number of mod_plsql connections (could equal the number of users concurrently logging out, because Single Sign-On uses `mod_plsql` for logout)
- Number of Oracle Internet Directory connections, computed as the product of Oracle Internet Directory processes and the number of database connections for each process
- Number of database connections used by other Oracle products

> **See Also:** See the discussion of PROCESSES in *Oracle10i Database Administrator's Guide*.

Tuning Interactions with OracleAS Single Sign-On

For optimum performance, ensure that the Single Sign-On database pool size is not less than the OracleAS Certificate Authority database pool size.

In OracleAS Certificate Authority, the minimum database pool size is hardcoded as 3, so for OracleAS Certificate Authority you can only adjust the maximum. Single Sign-On lets you configure both the minimum and maximum pool size, which are in the `policy.properties` file as the parameters (minConnectionsInPool and maxConnectionsInPool.

> **See Also:** See "policy properties" in *Oracle Application Server Single Sign-On Administrator's Guide*.

Tuning Maximum Memory

OracleAS Certificate Authority is configured to use a maximum memory of 256MB, which should be sufficient for most purposes. However, if you experience OutOfMemory errors, you can change your configuration to use more memory by setting the JVM heap size.

> **See Also:** See the discussion of JVM heap in *Oracle Application Server Performance Guide*.

Tuning Oracle Internet Directory Connections

The number of database connections used by Oracle Internet Directory depends on the number of Oracle Internet Directory processes and the number of database connections for each process. Tune these parameters by adjusting the number of concurrent database connections a single directory server process can have and the number of server processes a single instance can spawn.

> **See Also:** See the discussion of connections and Server Management Fields in Oracle Directory Manager in the appendix on graphical user interfaces in *Oracle Internet Directory Administrator's Guide*.

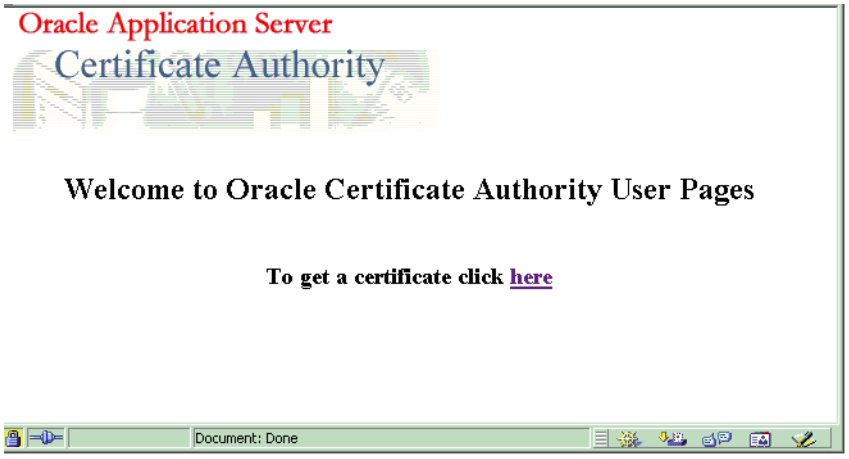
Tuning Other Components

Other Oracle components that you have installed may have additional parameters useful in tuning your system's performance. See the manuals for those components to learn about possible steps you can take toward performance enhancement at your site.

Customization Support

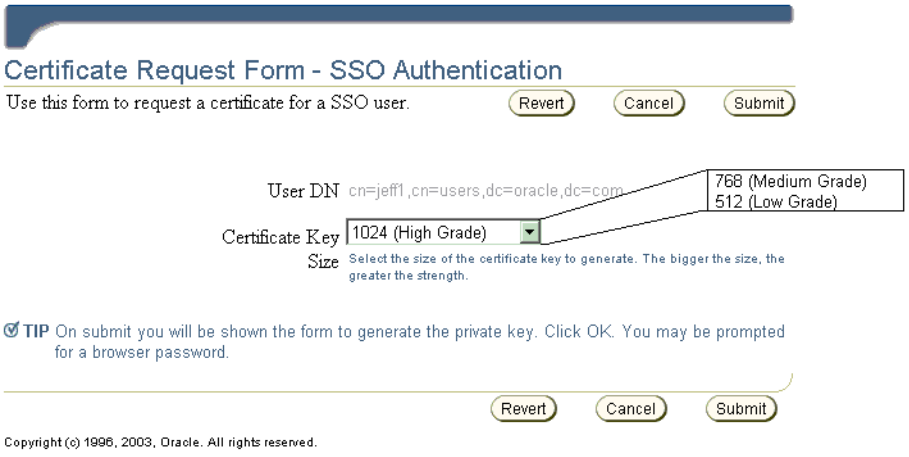
OracleAS Certificate Authority enables you to customize the SSO-OracleAS Certificate Authority interface by specifying your own headers and footers for the following three provisioning pages:

1. The Welcome screen

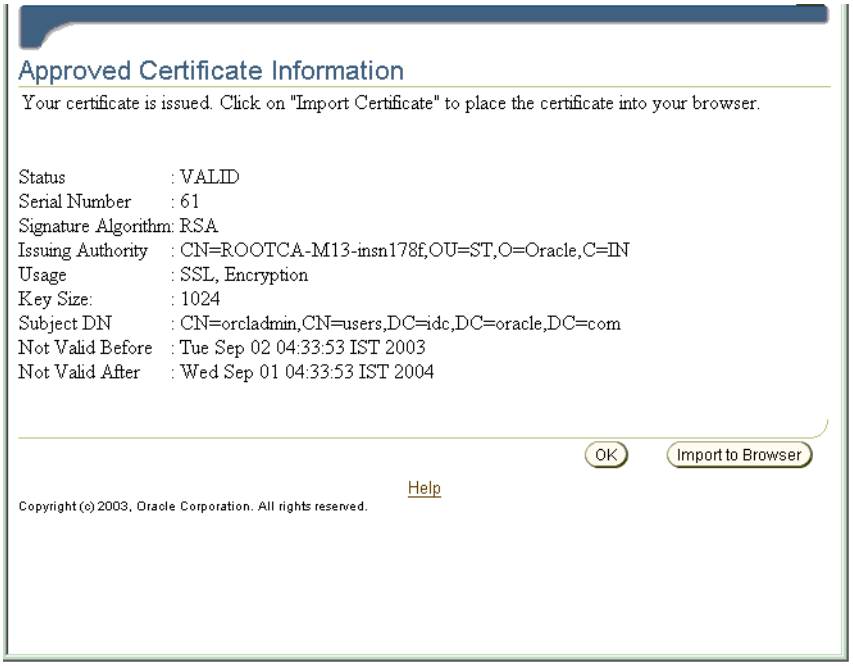


Note: To see this pop-up, users must have pop-up-blocking turned off in their browsers.

2. The Enrollment screen



3. The Install Certificate screen



See: "Single Sign-on and OracleAS Certificate Authority" in Chapter 4, "Introduction to Administration and Certificate Management"

Although OracleAS Certificate Authority will by default render the existing screens without customization, the OracleAS Certificate Authority administrator can customize any of the three with unique headers or footers. By providing a custom HTML file for any such screen, the administrator signals OracleAS Certificate Authority to use that customized screen in place of the corresponding default screen. These custom HTML files can contain static HTML content. If no customized HTML files are provided, or if they are of zero size, the default screens are used.

The templates for creating such a custom HTML file are in the directory named `$ORACLE_HOME/oca/templates/screens`. The administrator controls the look and feel of this content.

If any customization screen HTML file exists with nonzero size, its content is added to the default screen at the specified position indicated in Table 7–2.

Notes:

After making changes to any of these HTML files, the administrator must restart OracleAS Certificate Authority to cause those changes to be used.

Please note that OracleAS Certificate Authority is not responsible for the content, translation, or accessibility of anything customized, such as screens, messages, alerts, notifications, or other content, which will be rendered as is.

Table 7–2 Customization of Single Sign-On Popup Screens

Screen Name	Position of Replaceable Text	Files to Contain Replaceable Text[1]
Welcome Screen	For a header: between the OracleAS Certificate Authority lines reading "Welcome to OracleAS Certificate Authority" and "To get a certificate Click Here"	`$ORACLE_HOME/oca/templates/screens/homeheader.html`
	For a footer: under the line reading "To get a certificate Click Here"	`$ORACLE_HOME/oca/templates/screens/homefooter.html`
Enrollment Screen	For a header: between OCA's "Blue bar at the top" and the line reading "User DN"	`$ORACLE_HOME/oca/templates/screens/enrollheader.html`
	For a footer: under the lines at the bottom, reading "Key Size" and "SKI".	`$ORACLE_HOME/oca/templates/screens/enrollfooter.html`
Install Certificate Screen	For a header: between the OracleAS Certificate Authority lines reading "View Certificate" and "Certificate details"	`$ORACLE_HOME/oca/templates/screens/importheader.html`
	For a footer: between the OracleAS Certificate Authority lines reading "After certificate details" and "SKI" at the bottom.	`$ORACLE_HOME/oca/templates/screens/importfooter.html`

[1] If any file in this column exists with nonzero size, the corresponding header or footer will be replaced with that file's static HTML.

Log or Trace OracleAS Certificate Authority Actions

You can use the `ocactl` set command to enable log and trace, so that OracleAS Certificate Authority/Admin operations can be viewed in the log/trace storage.

Table 7–3 Storage Locations for OracleAS Certificate Authority Log and Trace Data

Type of Data	Storage Form	Location
OracleAS Certificate Authority Log	OracleAS Certificate Authority repository	OracleAS Certificate Authority repository
OracleAS Certificate Authority Trace	File: oca.trc	$ORACLE_HOME/oca/logs
ADMIN Log	File: admin.log	$ORACLE_HOME/oca/logs
ADMIN Trace	File: admin.trc	$ORACLE_HOME/oca/logs

The set command has the following format:

```
ocactl set  -type {LOG | TRACE} -mode {OCA|ADMIN} -state {ON|OFF}
```

Examples:

1. `ocactl set -type LOG -mode OracleAS Certificate Authority -state ON`

   Enables storing log messages in the OracleAS Certificate Authority repository.

2. `ocactl set -type TRACE -mode OracleAS Certificate Authority -state ON`

   Enables storing trace messages in the `oca.trc` file.

3. `ocactl set -type LOG -mode ADMIN -state ON`

Enables storing log messages in the `admin.log` file.

4. `ocactl set -type TRACE -mode ADMIN -state ON`

   Enables storing trace messages in the `admin.trc` file.

5. `ocactl set -type TRACE -state OFF`

   Turns off tracing: no trace data are stored.

Clearing Log or Trace Information for OracleAS Certificate Authority

The `ocactl` administrative command line tool enables removal of existing log or trace storage at the administrator's choice. The OracleAS Certificate Authority log will be stored in the OracleAS Certificate Authority repository, and the OracleAS Certificate Authority trace will be stored in `oca.trc` at `$ORACLE_HOME/oca/logs`. The Admin log will be stored in `admin.log` at `$ORACLE_HOME/oca/log`, and the Admin trace will be stored in `admin.trc` at `$ORACLE_HOME/oca/logs`.

Executing the clear command on an allowed type and mode deletes the old contents of such log or trace storage. Files affected by such commands (oca.trc, admin.trc, or admin.log) are simply removed from the file system.

The clear command has the following format:

```
ocactl clear -type {LOG |TRACE}  -mode {OCA|ADMIN}
```

Examples:

1. `ocactl clear -type LOG -mode ADMIN`

   Removes the Admin log file `admin.log` from `$ORACLE_HOME/oca/logs`

2. `ocactl clear -type TRACE -mode ADMIN`

   Removes the Admin trace file `admin.trc` from `$ORACLE_HOME/oca/logs`

3. `ocactl clear -type LOG -mode OCA`

   Removes log messages in the OracleAS Certificate Authority repository

4. `ocactl clear -type TRACE -mode OCA`

   Removes the OracleAS Certificate Authority trace file `oca.trc` from `$ORACLE_HOME/oca/logs`

Changing the Infrastructure Services

Changes to OracleAS Single Sign-On and Oracle Internet Directory, such as using a new port or host, can arise in a variety of ways, including the following situations:

- Restore operations after a backup
- Configuration changes to LDAP (directory) or the Oracle Database
- Migration from a pilot scenario to a production environment

See Also: The *Oracle Application Server Administrator's Guide*

OracleAS Certificate Authority is installed as part of the OracleAS Identity Management (IM) infrastructure and uses the services of Oracle Internet Directory, OracleAS Single Sign-On, and a metadata repository. If any of these components is replaced or restored, OracleAS Certificate Authority can be configured to use these new services. It can either use existing versions of these three components or work

with a new Oracle Internet Directory, OracleAS Single Sign-On and metadata repository.

Oracle Application Server supports two types of infrastructure change:

- Changing Identity Management (IM) Services
- Changing Metadata Repository (MR) Services

The following section describes the display of data regarding these services:

- Where Connection Information Is Stored and Displayed

Changing Identity Management (IM) Services

After installation of a new OracleAS Single Sign-On or Oracle Internet Directory, changing OracleAS Certificate Authority's IM Services requires two steps:

- Installing a new IM and migrating the existing data.
- Configuring OracleAS Certificate Authority to use the newly installed IM Services.

OracleAS provides scripts to migrate data from one IM instance to another, assuming that a new IM (OracleAS Single Sign-On/Oracle Internet Directory) has been installed. However, you cannot use the Change Identity Management Wizard on the Infrastructure page of the Application Server Control Console to change OracleAS Certificate Authority services because OracleAS Certificate Authority itself is an infrastructure component. So OracleAS Certificate Authority supports changing OCA's IM Services by providing the "changesecurity" command from the OracleAS Certificate Authority administrative command line tool `ocactl`.

See Also:

- Appendix A, "Command-Line Administration" for more details on the OracleAS Certificate Authority administrative command line tool, and
- The *Oracle Application Server Administrator Guide* for more details on changing the IM and Metadata Services of the Identity Management Infrastructure, including scripts.

The following steps establish the new IM services for OracleAS Certificate Authority:

1. Install Identity Management and Metadata Repository on Machine 1.
2. Install Identity Management on Machine 2.
3. Migrate IM data from Machine 1 to Machine 2 using the scripts provided by OracleAS.
4. In the machine with OracleAS Certificate Authority (Machine 1), bring down OracleAS Certificate Authority, OracleAS Certificate Authority's OC4J, and OHS. Use these commands:

```
$ORACLE_HOME/oca/bin/ocactl stop
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

5. In Machine 1, edit the `ias.properties` file to make the OIDhost and OIDport parameters under `$ORACLE_HOME/config` directory point to the new IM, that is, Machine 2.
6. On Machine 1, execute the following command:

```
$ORACLE_HOME/oca/bin/ocactl changesecurity -server_auth_port portno
```

This command performs the following two actions:

- Updates the file `oca.conf` at `$ORACLE_HOME/oca/conf` to point to the new IM Services machine (Machine 2)
- Registers OracleAS Certificate Authority with the new OracleAS Single Sign-On server (Machine 2)

Note: Identity Management (IM) reassociation can be used to accommodate changes to the configuration of OracleAS Single Sign-On or Oracle Internet Directory services for scalability or failover purposes, or to accommodate the transition from a pilot IM to production IM.

For more information on such reassociation, see *Oracle Application Server Administrator's Guide*.

Changing Metadata Repository (MR) Services

Changing OracleAS Certificate Authority's Metadata Services from one physical database to a different physical database is not supported. However, changes to connection strings, such as changing the listener or the port, are accommodated by using the `updateconnection` command as documented in Appendix A.

Where Connection Information Is Stored and Displayed

Information defining connections to the OracleAS Certificate Authority repository and directory (used for publishing certificates) is stored in Oracle Internet Directory. This connection information is originally written to Oracle Internet Directory when Oracle Application Server is installed, at which time it is also fetched from Oracle Internet Directory and written into the OracleAS Certificate Authority configuration file `$ORACLE_HOME/oca/conf/oca.conf`.

This connection information is displayed under Settings in the General subtab of the OracleAS Certificate Authority web interface for the administrator.

See Also: `updateconnection` in Table A–2, " Operations and Parameters of the OracleAS Certificate Authority (OCA) ocactl Tool" of Appendix A, "Command-Line Administration".

OracleAS Certificate Authority and High-Availability Features

The primary reference for Oracle Application Server high-availability features is the *Oracle Application Server High Availability Guide*. The following discussion is merely an overview to orient you to those features.

OracleAS Certificate Authority facilitates swift and easy use of certificates in real-world, high-availability systems. The linkages, procedures, conventions, and preparations supporting the high-availability capabilities of Oracle® Application Server Cold Failover Clusters and Real Application Clusters are discussed in the following sections of the *Oracle Application Server High Availability Guide*:

- OracleAS Certificate Authority Deployment Using Cold Failover
- OracleAS Certificate Authority Deployment Using Real Application Clusters

OracleAS Certificate Authority Deployment Using Cold Failover

In a cold-failover configuration, a number of physical hosts have access to a common store on shared disks, and each physical node can host one or more logical hosts at the same time. Using an Oracle® Application Server Cold Failover Cluster enables transparent failover of an Oracle Application Server instance from a failed node to a backup. The failover can also be initiated manually, for maintenance.

In this example, there is only one software and database installation to be performed, and two physical hosts share access to the disk on which the OracleAS Certificate Authority/Oracle Application Server software and database reside. If the hardware for Oracle Application Server is configured as a cluster of machines, then the installer recognizes the node as part of the cluster and asks for the name of the virtual host. When the physical host 1 fails or is taken offline for maintenance purposes, its logical hostname (virtual host A) will be migrated to the other physical host. Vendor-specific scripts and hardware cluster software can be used to start the required database, listener and OracleAS Certificate Authority/Oracle Application Server processes to effect transparent failover. Clients continue to talk to the same logical host as before, with minimal service disruption. OracleAS Certificate Authority, too, must be restarted with the `ocactl start` command, after HTTP server and OC4J are brought up.

See Also: *Oracle Application Server High Availability Guide*.

OracleAS Certificate Authority Deployment Using Real Application Clusters

OracleAS Certificate Authority provides limited support for Real Application Clusters (RAC). It can use the RAC configuration's other infrastructure components, such as Oracle Internet Directory, Oracle Database, and OracleAS Single Sign-On, but OracleAS Certificate Authority itself cannot be in the RAC mode.

See Also: *Oracle Application Server High Availability Guide* for guidance regarding how to install these components in the RAC mode.

Note: In OracleAS Certificate Authority 10*g* Release 2 (10.1.2), RAC is not supported on Windows.

OracleAS Certificate Authority Backup and Recovery Considerations

The phrase "backup and recovery" refers to the various strategies and procedures involved both in guarding against data loss and in reconstructing the data if a loss occurs. The Oracle Application Server backup recovery tool aids in backing up and recovering the Oracle Application Server environment in the event of a failure.

See Also: Full documentation of backup and recovery tools and procedures appears in the following books:

- For detailed descriptions of the various backup and recovery methods available, the installation and configuration of the Oracle Application Server backup/recovery tool, and component-wise backup and recovery, please refer to the backup/recovery documentation in the *Oracle Application Server Administrator's Guide*.
- For database backup, use the Oracle backup and recovery guidelines as described in the *Oracle10i Backup and Recovery Advanced User's Guide*.
- For backing up Oracle Internet Directory, use the *Oracle Internet Directory Administrator's Guide*.

The descriptions that follow are introductory only; full information is in the books listed earlier.

Scenarios in which backup/recovery techniques could be used to recover data include the following situations:

Table 7–4 Scenarios for Backup/Recovery

Situations	Responses
Loss of host	You can restore to a new host with the same hostname and IP address. Alternatively, you can restore to a new host with a different hostname and IP address.
Oracle software/binary loss or corruption	If any Oracle binaries are corrupted or lost, you must recover the entire infrastructure.
Metadata Repository instance failure, such as a failure of the database instance	Use database instance recovery methods to recover the metadata repository instance.
Metadata Repository database failure, meaning only the metadata repository has been corrupted, and not any other files in the infrastructure Oracle home.	Take a backup of the metadata repository using B/R scripts and recover the database using the OracleAS Backup and Recovery Tool.
Deletion/corruption of Oracle Application Server component runtime configuration files	Restore configuration files using a B/R script.
Metadata Repository listener failure	Kill and restart the listener process.

Various backup and recovery procedures protect and preserve OracleAS Certificate Authority content and capability in the event of required maintenance or unexpected loss of service.

Backup and corresponding recovery methods are supported by the following backup/recovery tools:

Table 7–5 Backup/Recovery Tools

Tool Name	Functionality
Cold Backup/Recovery	Refers to restoring the entire Oracle Application Server infrastructure instance including the Oracle home, configuration files, and database files that were backed up after completing a clean and normal shutdown of all Oracle Application Server infrastructure processes and metadata repository
Partial Online (hot) Backup/Recovery	Refers to restoring the Oracle Application Server infrastructure configuration files and database files that were backed up after completing a proper online backup of the OracleAS instance and metadata repository
Incremental Backup/Recovery of Configuration files	Refers to restoring only the Oracle Application Server infrastructure configuration files taken from an online backup

Since OracleAS Certificate Authority uses the Oracle Database as its primary repository, the OracleAS Certificate Authority information stored in that database will be automatically backed up when that database is backed up. Similarly, OracleAS Certificate Authority relies on Oracle Internet Directory for publishing certificates and for certain OracleAS Single Sign-On operations. The three books named at the start of this section provide the detailed information supporting all related backup and recovery operations.

In addition to information stored in the database and directory, OracleAS Certificate Authority also creates a number of important operating system files. These files should be backed up as part of the normal backup process. These files are as follows (where `$ORACLE_HOME` represents the home directory in which OracleAS Certificate Authority is installed):

- `$ORACLE_HOME/oca/templates/*`
- `$ORACLE_HOME/oca/policy/*`
- `$ORACLE_HOME/oca/logs/admin.*`
- `$ORACLE_HOME/oca/conf/oca.conf`
- `$ORACLE_HOME/oca/conf/ocaplugin.xml`
- `$ORACLE_HOME/oca/pwdstore/ocmpassword.p12`
- `$ORACLE_HOME/oca/wallet/ssl/cwallet.sso`
- `$ORACLE_HOME/oca/wallet/ssl/ewallet.p12`
- `$ORACLE_HOME/oca/wallet/ldap/ewallet.p12`
- `$ORACLE_HOME/Apache/Apache/conf/ocm_apache.conf`
- `$ORACLE_HOME/Apache/Apache/conf/osso/oca/osso.conf`

Restricting the Realm of Certificate Publication

Large organizations with geographically separate campuses can establish separate Certificate Authorities for each campus for more efficient local administration. These campuses could be in different states, such as Wyoming and New York, or in different countries, such as one in the United States and one in the United Kingdom. The different instances of OracleAS Certificate Authority may be Sub CAs or independent CAs that trust each other.

By default, when an OracleAS Certificate Authority instance is installed in a particular machine, an entry is placed into Oracle Internet Directory representing that installed OracleAS Certificate Authority instance, with the following DN:

```
cn=ocaN,cn=OCA,cn=PKI,cn=Products,cn=OraclContext
```

(where N is 1,2 …n)

To see the entry that corresponds to the current OracleAS Certificate Authority, go to the Administration page, to the Configuration Management tab and the General subtab. The DN under the Directory settings entry for Agent shows you the current Oracle Internet Directory for the current OracleAS Certificate Authority.

By default, each such CA is a member of the group cn=PKIAdmins,cn=Groups, cn=OracleContext, which is the top-level Oracle context.

When such a CA publishes a user certificate, that certificate is automatically placed in that user's DN entry in the corresponding subscriber realm within Oracle Internet Directory. By default, all CA's are trusted and can publish to any user entry in the whole directory. For example, a user in the US realm could receive a certificate from the UK OracleAS Certificate Authority, and the user certificate would be placed in that user's DN entry in the US realm.

However, it is possible to restrict the publishing rights of an OracleAS Certificate Authority instance so that it can only publish to a particular subscriber realm. For example, the UK OracleAS Certificate Authority could be restricted to publishing only to the UK subscriber realm. If this is done, then a certificate issued by the UK OracleAS Certificate Authority to a US user could not be published, because the user's standard realm would not be accessible to the UK OracleAS Certificate Authority.

To restrict an OracleAS Certificate Authority to a particular realm, you must remove it from the top-level group (cn=PKIAdmins,cn=Groups, cn=OracleContext) and add an entry for that OracleAS Certificate Authority to the desired group. For example, to restrict OCA2 to publish only to this subscriber dc=com,dc=acme, the following two commands would be used:

```
-remove cn=oca2,cn=cn=OCA,cn=PKI,cn=Products,cn=OraclContext from group
cn=PKIAdmins,cn=Groups,cn=OracleContext

-add cn=oca2,cn=cn=OCA,cn=PKI,cn=Products,cn=OraclContext to group
cn=PKIAdmins,cn=Groups,cn=OracleContext,dc=acme,dc=com
```

In addition, a custom plug-in can be written to limit the CA to manage only certificates from a specific set of DN's. For example, the sample plug-in developed in Chapter 6, "Managing Policies in Oracle Application Server Certificate Authority" restricts that CA to issuing certificates from non-U.S. domains only.

This restriction appears in line 12 of the example in that chapter's section entitled "An Example of a Custom Policy Plug-in":

```
12:             if (!policyRequest.getCountry().equals("US"))
```

A few alterations --- removing the "!" and changing "US" to whatever realm is desired --- plus fixing a few subsequent, dependent lines would restrict certificate issuance to that chosen realm.

Replacing the CA and Deinstalling OracleAS Certificate Authority

In the rare and drastic event that the root CA needs to be replaced, perhaps because its private key was somehow compromised, OracleAS Certificate Authority should be deinstalled and then reinstalled. The deinstallation will remove all traces of the original installation's database and Oracle Internet Directory entries.

To accomplish this deinstallation, follow the instructions in Section C.1.5 of the Oracle Application Server 10g Installation Guide.

8

End-User Interface of the Oracle Application Server Certificate Authority

The term "end-users" includes persons, of course, but also server entities that acquire certificates to facilitate authentication among servers and applications.

Separate HTML interfaces exist for end-user and administrator interaction with the OracleAS Certificate Authority server. Using these HTML forms, end-users can perform personal certificate-related operations and the administrator can perform certificate administration and management.

> **See Also:** For the OracleAS Certificate Authority web administrator interface, see
>
> - Chapter 4, "Introduction to Administration and Certificate Management"
> - Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics"

The present chapter describes the end-user interface in the following sections:

- Accessing the User Interface
- End-User Tabs and Processes
  - User Certificates Tab
  - Certificate Retrieval, Renewal, and Revocation
  - Server/SubCA Certificates Tab
  - Subordinate CA Certificates
- Installing a CA Certificate
- Handling Certificate Revocation Lists (CRLs)
- Importing a Newly Issued Certificate to Your Browser
- Exporting (Backing up) Your Wallet from Your Browser
- Importing a Certificate from Your File System

Both Netscape and Internet Explorer are supported.

Accessing the User Interface

To access the home page for the end-user interface to OracleAS Certificate Authority, launch your web browser and enter the URL and port number of the administration server as they were displayed at the end of installation. For example:

```
https://server1.example.com:6600/oca/user
```

The Oracle Application Server Certificate Authority user home page appears:

As the page itself explains, you can use this web interface to request, renew, revoke, or find any certificate or certificate request. To access these capabilities, you can click either the **User Certificates** tab or the **Server/SubCA Certificates** tab.

You can also use the **click here** links to install into your browser the Certificate Authority's certificate or the latest certificate revocation list (CRL).

Similarly, administrators can use their **click here** links to save the CA certificate or CRL into their file system for additional uses.

End-User Tabs and Processes

The OracleAS Certificate Authority web interface enables two types of end-user interaction with OracleAS Certificate Authority, as represented by the two tabs:

From the **User Certificates** tab you can

- authenticate yourself to the OracleAS Certificate Authority, either by using existing Single Sign-On or SSL certificates, or by requesting manual authentication by an administrator,
- create a new certificate request for manual approval by the OracleAS Certificate Authority administrator (for end-users or servers),
- request and receive a certificate automatically (for SSL and OracleAS Single Sign-On users),
- install, view, revoke, or renew your certificates,

- change your authentication method,
- save or install the CA certificate, or
- save or install the latest certificate revocation list (CRL).

Table 8–1 lists the types of certificates that Oracle Application Server Certificate Authority supports and provides a brief explanation for each.

Table 8–1 Choices for Certificate Usage

Function	Description
Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.) The user of an Authentication certificate intends the certificate to be used during SSL authentication.
Encryption	Enables encrypting and decrypting electronic documents, including email messages, using SMIME. When using an Encryption certificate to encrypt email, the user provides it to others to enable them to send messages to him, encrypted with his public key. Then only he can decipher them using the private key. (Note 1) To use an Encryption certificate with mail clients, such as Outlook or Mozilla, see Appendix G, "S/MIME with OracleAS Certificate Authority".
Signing	Enables verifiable signature for (and assures non-tampering of) electronic documents, including email (using S/MIME, the Secure Multipurpose Internet Mail Extension) The user of a Signing certificate intends to use it to sign message digests with his private key, enabling others to use his public key to verify that he originated the message and it is unchanged. To use a Signing certificate with mail clients, such as Outlook or Mozilla, see Appendix G, "S/MIME with OracleAS Certificate Authority".
Authentication, Encryption	Certificate can be used for both purposes.
Authentication, Signing	Certificate can be used for both purposes.
Authentication, Encryption, and Signing	Certificate can be used for all three purposes.
Encryption, Signing	Certificate can be used for both purposes.
CA Signing	Enables requesting subordinate CA certificates A Certificate Authority uses the private key of a CA Signing certificate to sign the certificates it issues, enabling recipients to use its public key to verify that the certificate was in fact signed by this specific Certificate Authority.
Code Signing	Provides verifiable signature for the provider of (and assures non-tampering of) Java code, JavaScript, and other signed files. The user of a Code Signing certificate intends to sign software his private key, enabling clients to use his public key to verify that he is indeed the source of the software.

From the **Server/SubCA Certificates** tab, you can

- search for certificates and certificate requests by ID, serial number, or Common Name, and so on,
- request server and sub-CA certificates, or
- install the CA certificate or the certificate revocation list (CRL).

User Certificates Tab

Upon first entering this tab, you see the Authentication page, which enables you to select how you authenticate yourself to Oracle Application Server Certificate Authority.

Table 8–2 lists the available types and methods:

Table 8–2 Types of Authentication

Authentication Type	Description	Method in brief (details in following sections)
Single Sign-On	Authentication is automated, based on your single sign-on server. Typically it is password-based.	Click the radio button labeled **Use your Oracle Single Sign-on name and password** and then click **Submit.**
Secure Socket Layer (SSL)	Authentication is automated, based on your pre-issued SSL certificate.	Click the radio button labeled **Use Your Existing Certificate** and then click **Submit**
Manual	Authentication is not automated. You must fill out a Certificate Request form, submit it, and wait for approval from the administrator.	Click the radio button labeled **Use manual approval/authentication** and then click **Submit.**

> **See Also:** Chapter 2, "Identity Management and OracleAS Certificate Authority Features" about authentication.

These types and methods are explained in greater detail in the following sections:

- Single Sign-on Authentication (SSO)
- Configuring Your Browser to Trust OracleAS Certificate Authority
- Secure Sockets Layer (SSL) Authentication
- Manual Authentication
- Certificate Retrieval, Renewal, and Revocation
- Server/SubCA Certificates Tab
- Subordinate CA Certificates

Note: For both end-users and administrators, the default in Mozilla is the 2048 key size.

Internet Explorer offers 512 bits (basic) or 1024 bits (enhanced) or 2048 bits (strong). The default is "strong" or, if that is unavailable, "enhanced," and if that choice is unavailable, "basic." If the computer does not have a smart card reader, then choosing Gemplus gives users an error because no key-size resolution is found. If there *is* a card-reader, then the smart card choice determines the key size.

Single Sign-on Authentication (SSO)

The following steps enable OracleAS Single Sign-On users to get a certificate automatically, or to manage their certificates, by supplying the required OracleAS Single Sign-On authentication information, such as username and password:

1. In the Authentication form, select the option labeled **Use Your Oracle Single Sign-On Name and Password** and click **Submit**.
   You will be redirected to the OracleAS Single Sign-On login page.
2. Enter your OracleAS Single Sign-On user name and password. The **User Certificates - SSO** form appears, showing your valid certificates and enabling you to do the following tasks:
   - Get a Certificate.
   - View Details of a Selected Certificate.
   - Renew a current certificate.
   - Revoke a current certificate.

   To get a certificate, do steps 3 through 5:
3. Click **Request a Certificate** on the User Certificates - SSO form to display the Certificate Request form.
4. In the Certificate Request form, enter the information appropriate to you and submit the form. The choices you see when using Netscape are slightly different from those you see when using Internet Explorer:
   - **In Netscape**, the phrase **Certificate Key Size** appears, referring to the size, in bits, of the key-pair to be generated: 512, 1024, ...

     In Internet Explorer, the phrase **Cryptographic Service Provider** appears, referring to a choice of providers for cryptography service. Standard choices include Microsoft Basic Crypto Provider, Microsoft Enhanced Crypto Provider, and Microsoft Strong Crypto Provider. The OracleAS Certificate Authority default is the "Strong" choice, if available, followed by Enhanced, if available, and then by Basic. Other choices may also be present, such as Gemplus for smartcard usage. Select the size according to your requirements.
   - **Certificate Usage:** Choose the types of operations for which you will use this certificate. Your OracleAS Certificate Authority administrator sets the standard default shown first in the list, but you can, if you wish, choose a different item from the drop-down list, as shown in Table 8–1.
   - **Validity Period:** Duration of the certificate's validity, in days. However, OracleAS Single Sign-On users need not key in the validity period information because it is automatically set by the Oracle Application Server Certificate Authority, using the number specified for the "default Validity period" in the ValidityRule policy.

   After you submit the filled-out form, the Certificate form appears, showing the information recorded on the certificate.
5. After checking that the information about you is correct, make a note of the name of the signer of the certificate: you will need that name later. Then click the **Install to Browser** button to install the certificate into your browser. Netscape and Internet Explorer report successful installation differently:

Note: If you click **OK** instead of **Install in Browser**, your certificate ***is*** created, stored in the OracleAS Certificate Authority repository, and published to the Oracle Internet Directory. However, your browser cannot supply it to a server until you install it. See "Importing a Newly Issued Certificate to Your Browser".

- **In Netscape**, you will know the certificate has been installed when you see the words "Document Done" in the status bar at the lower left of your browser. At that point, click OK: even though the cursor continues to show the hourglass, the action is completed. The corresponding CA's (Signer's) certificate has also been installed.

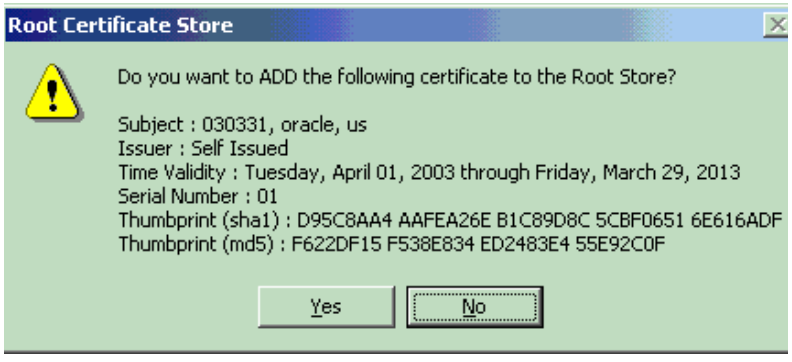
Note: For this certificate to be trusted, you need to edit the CA certificate's uses, specifying that you trust certificates issued by this Certificate Authority for network sites, email users, software developers, or all three. Checkboxes for these choices are reached through the Security choice on Netscape's menu bar: see "Trusting a Certificate Issuer in Netscape".

- **In Internet Explorer**, you know the certificate has been installed when you see the message "Certificate has been installed successfully". You are also asked whether you want the Signer's certificate installed, on a window showing the details of that CA. Click OK to ensure that certificate is also installed. Internet Explorer automatically treats such a certificate as trusted.

Configuring Your Browser to Trust OracleAS Certificate Authority

This process is slightly different in Internet Explorer, Netscape, and Mozilla Firefox.

Trusting a Certificate Issuer in Internet Explorer When you install a certificate using Internet Explorer, it asks whether you wish to add that certificate to the Root Store:



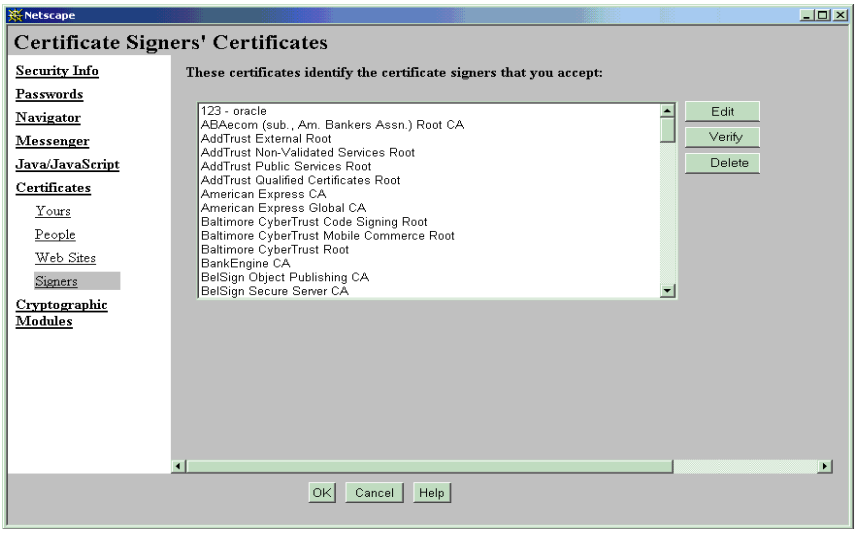
Clicking **Yes** installs the certificate and sets the issuer as "trusted." You can view your certificates by selecting the menu choices "Tools - Internet Options - Content - Certificates." The four tabs then shown enable you to see your own certificates, those supplied by others to authenticate them to you, intermediate certificate authorities who have supplied certificates to you, and the root certificate authorities you have chosen to trust.

Trusting a Certificate Issuer in Netscape When you install a certificate using Netscape, it installs both the certificate you requested and the certificate representing the certificate authority that signed and issued your new CA certificate. The only notification you get is the message "Document Done" in the lower-left status-bar area of your browser. However, your new certificate is not trusted until you explicitly identify for Netscape those activities for which you want to trust the signer's certificate.

You do so by the following steps:

1. Open Netscape's Security Info page by clicking the "lock" icon in the status bar at the lower left of your browser. (Or by selecting "Communicator - Tools - Security Info" from the menu bar.) A page like the following appears:

2. Click the "Signers" link. A page like the following appears:

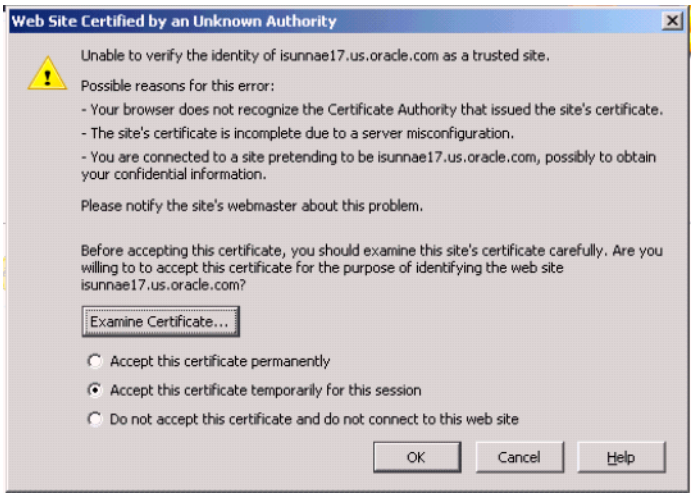


3. Click the name of the signer that you noted when viewing the certificate's details, and click **Edit**. A page like the following appears:

4. Click the three checkboxes shown as checked in the illustration, and then click **OK**.

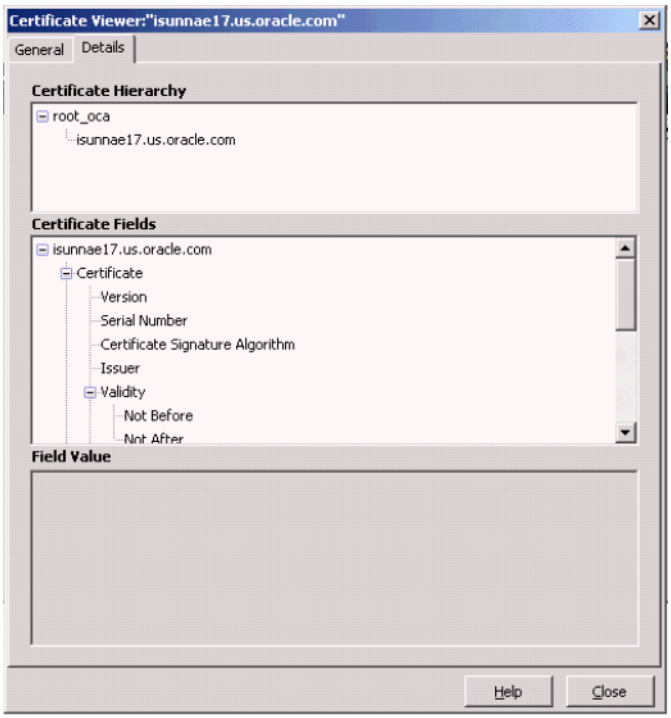
The CA certificate is now trusted to verify the certificates of network sites this browser connects to, of signed or encrypted messages received, or of signed software.

Trusting a Certificate Issuer in Mozilla Firefox When you install a certificate using Mozilla Firefox, it notifies you that the issuing certificate authority is unknown:



To make this a trusted certificate, select the **Accept this certificate permanently** checkbox and click **OK**.

You can inspect a certificate before accepting it, by clicking the **Examine Certificate** button. You will see a display like the following:



Select any field to display its value.

Secure Sockets Layer (SSL) Authentication

If you already have an SSL certificate from the Certificate Authority, you can obtain an OracleAS Certificate Authority certificate for future authentication purposes by using the current SSL certificate as identification, as follows:

1. From the Authentication form, select **Use Your Existing Certificate** option and click **Submit**. The **User Certificates - SSL** form appears, from which the following tasks can be performed:
   - Get a Certificate.
   - View Details of a Selected Certificate.
   - Renew a current certificate.
   - Revoke a current certificate.

   To get a certificate, do steps 2 through 5:
2. From the **User Certificates - SSL** form, click **Request a Certificate** to display the Certificate Request form.
3. In the **Certificate Request** form, enter the information appropriate to you and submit the form. The Netscape interface is slightly different from that of Internet Explorer, as explained earlier in "Single Sign-on Authentication (SSO)".

   After you submit the filled-out form, the **Certificate** form appears, showing the information recorded on the certificate.
4. After checking that the information is correct, click the **Install in Browser** button to install the certificate into your browser.
5. Click OK to return.

Manual Authentication

To obtain a certificate using manual authentication, perform the following steps:

1. From the Authentication form, select **Use Manual Approval Authentication** and click **Submit**. The **User Certificates** form appears, enabling you to specify your DN and contact information, as well as select the key size, usage, and validity period for the certificate you are requesting.
2. On the User Certificates form, click **Request a Certificate** to display the Certificate Request form.
3. In the Certificate Request form, enter the DN and contact information appropriate to you. (Separate DN entries with a comma.) Use the Enrollment form's drop-down list to select key size and Authentication (SSL) certificate (plus Encryption or Signing if desired), and then submit the form to the Oracle Certificate Authority administrator.

A Request ID is allocated, specific to this user request, which you use to locate the certificate once it is approved.

The certificate becomes available only after receiving the administrator's approval.

Once the administrator communicates that the certificate is approved, go to the Certificate Retrieval form, search for your certificate using your Request ID or DN, and install the certificate.

Certificate Retrieval, Renewal, and Revocation

After a certificate request is approved, the issued certificate can be retrieved for review and installation. Use the same machine and browser as when you requested the certificate.

After an OracleAS Single Sign-On- or SSL-certificate has been in use for a period, it can be renewed during a configurable time-window around its expiration date.

An issued certificate can be revoked if it is, for some reason upon review, incorrect or inappropriate or no longer valid for its intended user or activities.

These certificate operations are described in the following sections:

- Certificate Retrieval
- Certificate Renewal
- Certificate Revocation

Certificate Retrieval

After you receive notification that your manual-authentication certificate request is approved, you need to review the certificate and install it. You can find your certificate by entering the serial number from that notification into the search field on the User Certificates page. After it is found and you select it by clicking the radio button next to the serial number, you can click **View Details** to review the data used in generating it. Then you can install it as described in "Importing a Newly Issued Certificate to Your Browser".

If, for a particular certificate, these data are not correct, then that certificate should be revoked and replaced by applying for a new certificate.

Certificate Renewal

OracleAS Single Sign-On and SSL certificate users can renew their certificates

A user can renew such a certificate within a certain period of days before and after a certificate is due to expire. By default, this period is 10 days before and 10 days after the certificate's expiration date. However, the administrator can alter this period by using the Configuration tab in the administration web interface. Users can select a certificate, click **View Details**, and then renew the certificate.

Certificate Revocation

OracleAS Single Sign-On and SSL certificate users can revoke certificates.

If errors or problems are found with a certificate, or if a private key is stolen, and so on, the certificate should be revoked. The user can supply correct information for a new certificate. Using the new certificate should cancel out whatever issues were associated with the earlier one.

Revoking a certificate will mark it as revoked in OracleAS Certificate Authority repositories and causes it to be added to the CRL the next time the CRL is generated. However, revoked certificates are not removed automatically from your browser database. You should remove them manually. In Netscape, you click the Security icon on the browser, click the **Yours** choice under **Certificates**, select the revoked certificate from the list displayed, and click **Delete**.

Server/SubCA Certificates Tab

An administrator for any server can obtain a server certificate enabling PKI authentication for that server with other servers or users. To do so, a PKCS#10 request form is needed, which can be generated using Oracle Wallet Manager (or an equivalent third-party tool). See the Oracle Wallet Manager chapter in the Oracle Application Server Security Guide.

From the **Server Certificates** tab page, use the following steps:

1. On the Home page, select the **Server/Sub CA Certificates** tab to display the Server Certificate form.
2. Click the **Request a Certificate** button.
3. On the **Server / SubCA Certificate Request** form, you paste in the completed PKCS#10 request form generated earlier by Oracle Wallet Manager, and choose the type of certificate you want. You can request Authentication (SSL)/Encryption, Signing, Code Signing, or CA Signing server certificates. To function as a subordinate CA, specify **CA Signing** as the certificate usage in the enrollment form. You also choose the validity period for your requested certificate, from the drop-down choices presented.
4. Enter the appropriate information and submit the form to the administrator.

The server administrator obtains authentication only after the administrator approves this request.

Subordinate CA Certificates

In circumstances where a single CA is impractical, such as separate continental divisions in a single company, multiple CAs can be maintained within the PKI structure. In a hierarchical PKI, the root CA is the single CA trusted by all users. The root CA's public key is what serves as the beginning of the trust path for a security domain.

OracleAS Certificate Authority can be a root CA or it can obtain a Subordinate CA certificate from a third-party CA. OracleAS Certificate Authority can certify the

certificate signature of another CA, thereby creating a subordinate CA. The subordinate CA may in turn issue certificates to even lower-level CAs, creating what is called a certificate chain. An individual certificate signed by one of the subordinate CAs must present the certificates of all CAs up to the root. Because each authority's certificate is signed by a higher CA, a user can verify the validity of a particular certificate by tracing the certificate authority path back to the root CA.

To obtain a subordinate CA certificate, perform the following steps:

1. On the Home page, select the **Server/Sub CA Certificates** tab to display the Subordinate CA Certificates form.
2. Click the **Request a Certificate** button.
3. In the **Subordinate CA Certificate Request** form, enter the appropriate information, select certificate usage type as **CA signing**, and submit the form to the administrator.

The requester obtains a certificate only after the administrator approves this request.

Installing a CA Certificate

In Netscape, after you click **Request a Certificate**, OracleAS Certificate Authority presents a sequence of dialog boxes. These dialogs describe the operations that need to happen in order to accept the OracleAS Certificate Authority certificate. Click **Next** on each dialog box as it is presented, and **Finish** on the last one. Your CA certificate will be automatically installed into your browser.

For Internet Explorer, you are asked simply to accept or reject the CA Certificate install. You may wish to do so simply to trust servers whose certificates are issued by this CA, even if you do not want to get a certificate from it. The browser will ask whether you want to save the certificate or open it from the current location. To install the CA certificate into your browser, you select **Open the file from its current location** and click OK. In the next window that opens, choose **Install Certificate** and accept the certificate install to place the CA certificate into the browser's repository.

Handling Certificate Revocation Lists (CRLs)

CRLs can be installed in your browser or saved to disk to enable recognition and rejection of certificates that have expired or been revoked.

Upon clicking **Save CRL**, the CRL is displayed, showing all revoked and expired certificates. At the bottom of the page are the buttons **Install CRL in Browser**, **Save Binary CRL to Disk**, and **Save BASE64 CRL to Disk**.

Installing a CRL into Your Browser

Installing a certificate revocation list enables your browser to warn you if an incoming certificate offered by an individual or company has been revoked. Use of a revoked certificate could indicate a possible problem with impersonation or with a product being offered or used. Being warned can help you avoid potentially inappropriate interactions.

The steps for installing the CRL are browser-dependent:

- Installing the CRL In Netscape 7.x and Mozilla Firefox
- Installing the CRL In Internet Explorer (IE)

The operations to save a CRL to the file system are discussed in "Saving the Binary or BASE64 CRL to Disk".

Installing the CRL In Netscape 7.x and Mozilla Firefox

From the User Certificates tab of Oracle Application Server Certificate Authority, do the following tasks:

1. Click the **Install CRL in Browser** button. A Netscape dialog box tells you the import was successful. If automatic update was enabled for this CRL, you can view those settings by clicking **Yes**, or dismiss the dialog by clicking **No**.
2. If you click **Yes**, you can see when the next update is scheduled and what site provides that update.

You can manually delete or update this CRL by using this navigation path:

- In **Netscape**, follow Edit/Preferences/Privacy & Security/Validation/Manage CRLs
- In **Firefox**, follow Tools/Options/Advanced/Validation/Manage CRLs

If you already have the CRL and its validity is the same or later than the CRL being installed, a small dialog box informs you that the CRL you are attempting to install is not later than one already in your browser.

Installing the CRL In Internet Explorer (IE)

In IE, the CRL is not directly imported into the browser. As in the case of importing a CA Certificate, IE asks the question **Save to Disk** or **Open from the Current Location**. In the latter case, the CRL is not imported. If you choose **Save to Disk**, you then do the following actions:

1. Select the directory in which you want to store the CRL.
2. Click **OK**.

Saving the Binary or BASE64 CRL to Disk

In addition to installing the CRL into your browser, you can also

- save a binary copy of the CRL (named OCAcrl.crl) by clicking **Save Binary CRL to Disk** and choosing the directory in which you want it stored, or
- save an copy in Base64 format (named OCAcrlBase64.txt), which you can cut and paste, by clicking **Save BASE64 CRL to Disk** and choosing the target directory.

Saving a certificate revocation list (CRL) to disk in your file system enables other programs to use it to detect revoked or expired certificates offered by an individual or a company. Avoiding the use of such a certificate can protect your resources and applications from inappropriate or unauthorized uses.

To save a CRL to disk, follow these steps:

1. Go to the OracleAS Certificate Authority User Certificates Page.
2. Click **Save CRL to Disk**.
3. Click either **Save Binary CRL to Disk** or **Save Binary BASE64 CRL to Disk.**
4. Save the CRL into a directory of your choice.
5. Modify your `http.conf` file, located in `$ORACLE_HOME/apache/apache/conf`, to include the "SSLCARevocationFilePath"

parameter, and point that parameter to the directory containing the new CRL file. For example:

```
SSLCARevocationFilePath=/usr/myname/certstoreject.crl
```

Importing a Newly Issued Certificate to Your Browser

After your request for a certificate is approved, OracleAS Certificate Authority displays its details for you in a new window so that you can check that the details match what you intended. Check that the name, validity period, and other attributes on the certificate are as they should be. If those details include any serious error, you should revoke this certificate and apply for a new one, specifying all the correct information on the request form.

When you are satisfied, click the **Import Certificate** button to import a copy of the certificate into your browser. You will see the message `Document Done` in the lower-left status-bar area of your browser. You can then click OK.

If you were to simply click **OK** without clicking **Import Certificate**, the server would have a copy of your certificate but your browser would not. It could not supply the certificate when needed for authentication to an application, a directory, or another server.

The action of importing the certificate also imports the chain of CAs up to the root CA. However, in Netscape and Mozilla Firefox, the CA certificate imported along with the user certificate is not automatically trusted. You need to establish the trust, as follows:

- **In Netscape:**
  - Click Edit/Preferences/Privacy & Security/Certificates/Manage Certificates
- **In Mozilla Firefox:**
  - Click Tools -> Options
  - In the left pane, select the Advanced tab
- In the right pane, expand the certificate item
- Click Manage Certificates
- In both Netscape and Mozilla Firefox:
  1. Click the **Authorities** tab.
  2. Select the appropriate CA Certificate by name. (You may be prompted for the repository password.)
  3. Click Edit.
  4. Check the appropriate check boxes to trust this certificate for identifying Web sites and encrypting Web site connections, for signing or encrypting email users, or for identifying software makers.
  5. Select **OK**.

  This process establishes the desirable trust relationships so that the browser will trust the certificates issued by this imported certificate for the purposes you selected as **Certificate Usage** when you try to establish an SSL session.

Exporting (Backing up) Your Wallet from Your Browser

You can (and should) export your wallet to your file system for safekeeping, so that you can restore the contents after any possible disruption to your system or your

browser. The wallet contains your certificate, private key, and the chain of certificates for the trusted Certificate Authority that issued your certificate.

Use the following steps to export a certificate:

- **In Netscape:**
  - Click Edit/Preferences/Privacy & Security/Certificates/Manage Certificates
- **In Mozilla Firefox:**
  - Click Tools -> Options
  - In the left pane, select the Advanced tab
- In the right pane, expand the certificate item
- Click Manage Certificates

In both Netscape 7.x and Mozilla Firefox, continue as follows:

1. Select the certificate that needs to be exported and click Backup.
2. Enter the file name for the PKCS#12 wallet and click Save.
3. Enter the Netscape repository password, and click OK.

   A window appears, with the prompt `Please enter the master password for the Software Security Device`. Upon entering the correct password (the browser repository password), a new window appears.
4. In this window, labeled **Choose a Certificate Backup password**, you enter the password with which the PKCS#12 wallet will be encrypted. You will need to enter the same password again to confirm the password. A password quality meter in this window gives information on the quality of the password provided.
5. Click OK. An alert appears saying that backup is successful.

In **Internet Explorer**, use the following steps to export a certificate:

1. From the Tools menu, select **Internet Options**.

   A window opens showing six tabs you can choose from.
2. Select the **Content** tab, and click the **Certificates** button.

   The **Certificate Manager** window opens, with four tabs enabling you to see your personal certificates, those of other people, plus the names and expiration dates for trusted and intermediate issuers of certificates.
3. In the **Personal** tab, click the particular certificate you want to export.
4. Click the **Export** button under the subordinate window.
5. Click **Next** in the **Certificate Manager Export Wizard.**
6. If you wish to export the private key, click the **Yes** radio button. (If not, click the **No** radio button.) Clicking Yes means your private key is also stored.
7. Click **Next**.
8. Choose PKCS #12 and check the two checkboxes beneath it, and click **Next**.
9. When asked, enter a password to preserve the security of the private key. You will be asked for it twice, and what you enter must match.

   As usual, you must remember this password in order to retrieve and reuse this private key. Without the password, it will not be usable.

10. When asked, enter the file system destination, path name, and filename where this encrypted certificate and key is to be stored.
11. A new window shows the choices you've made. After verifying this information, click **Finish**.

    A message appears saying `The export was completed successfully`.
12. Click **OK**, **Close**, and **OK** to exit from the windows used for this process.

Importing a Certificate from Your File System

You can import a certificate into your browser from a file stored on your file system. The file must be of type pkcs12, with extension `.p12`. You will need to know the password that was used to encrypt that wallet.

Use the following steps to import a certificate from a PKCS#12 wallet in Netscape and Mozilla Firefox browsers:

- **In Netscape:**
  - Click Edit/Preferences/Privacy & Security/Certificates/Manage Certificates
- **In Mozilla Firefox:**
  - Click **Tools -> Options**
  - In the left pane, select the **Advanced** tab
- In the right pane, expand the certificate item
- Click Manage Certificates

In both Netscape and Mozilla Firefox, continue as follows:

1. Click Import.
2. Choose the PKCS#12 wallet containing the certificate and key to be imported and click **Open**.
3. Enter the Netscape Repository password in the popup that appears, and click **OK**.

   A prompt appears: `Please enter the master password for the Software Security Device`. Upon entering that password, a new window appears, labeled **Password Entry Dialog**.
4. In this new window, enter the password that will be used to decrypt the PKCS#12 wallet, and click **OK**.
5. An alert appears, saying that restoration of the certificate and private key is successful.

In **Internet Explorer** (IE), use the following steps to import a certificate from a PKCS#12 wallet:

1. From the Tools men, select **Internet Options**.

   A window opens showing six tabs you can choose from.
2. Select the **Content** tab, and click the **Certificates** button. The **Personal** tab lists your certificates.
3. Click **Import**. The Certificate Import Wizard window appears.
4. Click **Next** and then **Browse** to the directory containing your desired certificate.

5. Double-click to put the full path into the Wizard, and then click **Next**.
6. Enter the password for the wallet you selected.
7. Click **Next**.
8. Internet Explorer can automatically select the certificate store based on the type of certificate, or you can tell it where you want the certificates by clicking the other radio button and entering the path to that store.
9. Click **Next**.
10. Click **Finish.**

    If the certificate store being used by IE does not yet contain the certificate of the the CA who issued your certificate, a dialog box will appear asking if you want to add it to that store.
11. Click **Yes**. Having that certificate makes it possible to authenticate with other servers or users whose certificates were also issued by that CA (or another authority in the same chain of trust).

    IE displays a dialog box telling you the import was successful.
12. Click **Close** and **OK** to exit from the certificate and security area of IE.

A

Command-Line Administration

This Appendix is a "quick help" reference to commands and options available using the Oracle Application Server Certificate Authority command-line tool `ocactl`. The detailed usage of these commands, with use cases, will be explained in Advanced Topics.

This Appendix describes how to do OracleAS Certificate Authority administration tasks using the administrative command line tool `ocactl`, operating through the computer hosting the OracleAS Certificate Authority instance.

This chapter contains the following topics:

Table A–1 Links to Commands and Configuration Operations

General Topic	Links to Specific Subtopics
Basic Administration: Commands and Operations	■ Command-Line Tool ■ Starting the Oracle Certificate Authority Server ■ Stopping the Oracle Application Server Certificate Authority Server ■ Finding the Status of the Oracle Certificate Authority Services ■ Changing Privileged Passwords ■ Updating OracleAS Certificate Authority Repository Connection Information
Root Certificate Operations	■ Regenerating the Root Certificate Authority's Certificate ■ Revoking a Root CA Certificate
SSL/OracleAS Single Sign-On Operations	■ Converting a CA SSL Server Wallet into SSO Form ■ Regenerating the Certificate Authority's SSL Certificate and Wallet ■ Setting SSO Authentication (linksso, unlinksso commands)
Sub-CA Operations	■ Generating a Sub CA Signing Wallet from OracleAS Certificate Authority ■ Installing/Importing a Sub CA Signing Wallet ■ Generating a CA SSL Wallet for a Sub CA
Log/Trace Operations	■ Setting Log/Trace Options ■ Clearing Log or Trace Storage

Command-Line Tool

As the OracleAS Certificate Authority administrator, you use the command line tool named `ocactl` to specify the parameters needed to perform the various OracleAS Certificate Authority operations. (You may need to add `oca/bin` to your path.) Each time this tool is invoked it requests your OracleAS Certificate Authority Administrator

password, which is always the same as the CA signing password. (If you use a slow telnet/rlogin session and backspace while entering the password, some portions of it are echoed.)+

The general form for using this command is:

```
ocactl operation -type related-parameters, if any
```

For example, to start OracleAS Certificate Authority, you would enter

```
ocactl start
```

As another example, to generate a certificate and wallet for CASSL operations in publishing certificates with mutual authentication between OracleAS Certificate Authority and Oracle Internet Directory, you would enter

```
ocactl generatewallet -type CASSL
```

Notice that not all commands have parameters. Those that do not use parameters also do not use the keyword "-type".

Those that do need parameters must use the keyword `-type` preceding the parameter.

The only exception is the "`convertwallet`" command, which has a special syntax explained after Table A–2.

Table A–2 shows the main operations (in alphabetical order) and their related parameters. After the table, additional parameters for the `convertwallet` command are explained.

The following operation-names are links directly into that table:

changesecurity, clear, generatewallet, help, importwallet, linksso, renewcert, revokecert, set, setpasswd, start, stop, unlinksso, updateconnection

Table A–2 Operations and Parameters of the OracleAS Certificate Authority (OCA) ocactl Tool

Operation	Parameters	Meaning
changesecurity	-server_auth_port port	Changes the Identity Management services (Oracle Internet Directory/OracleAS Single Sign-On Server) used by OracleAS Certificate Authority to the new Oracle Internet Directory and OracleAS Single Sign-On. Updates oca.conf with the new IM machine and port number, and uses the specified port while registering OracleAS Certificate Authority with the new OracleAS Single Sign-On server.
clear	LOG, TRACE OracleAS Certificate Authority or ADMIN	Clears the storage location specified in a prior `set` command, either a file or a database table, for the type of log or trace data chosen, either OracleAS Certificate Authority or ADMIN. (If OracleAS Certificate Authority is not running, all such data is cleared.) Examples of each command appear in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics" at Log or Trace OracleAS Certificate Authority Actions.
convertwallet	See next column	See later discussion after this table: "Converting a CA SSL Server Wallet into SSO Form".

Table A–2 (Cont.) Operations and Parameters of the OracleAS Certificate Authority (OCA) ocactl Tool

Operation	Parameters	Meaning
generatewallet	CA, CASSL, or CASMIME	Generates a certificate and wallet for the type specified: certificate authority signing certificate, or certificate authority SSL certificate. A sample "generatewallet" command will thus look like this: `ocactl generatewallet -type CASSL` Wallets of the following type are stored in the indicated place: ■ CA OracleAS Certificate Authority repository ■ CASSL `$ORACLE_HOME/oca/wallet/ssl` ■ CASMIME OracleAS Certificate Authority repository For the CA, key size choices are 512, 1024, 2048, and 4096. Default is 2048. For CASSL and CASMIME, key size choices are 512, 768, 1024, and 2048, with 1024 the default.
help	command name	Shows the syntax for the command specified by name. A sample "help" command will thus look like the following: `ocactl help setconfig`
importwallet	SUBCA	After prompting for the directory where the wallet should be stored, and the administrator's password, this command installs a wallet named ewallet.p12 as a subordinate CA server wallet. A sample `importwallet` command will thus look like this: `ocactl importwallet -type SUBCA` Note: Before importing a wallet, ensure it is corruption-free and contains one or more self-signed certificates. You can verify a wallet with the `orapki wallet display` command.
linksso	none	Registers OracleAS Certificate Authority with OracleAS Single Sign-On to display OracleAS Certificate Authority certificate enrollment form to OracleAS Single Sign-On users who lack a certificate, so they can request one. (This command does not require OracleAS Certificate Authority service to be shut down, but it won't take effect until the OracleAS Single Sign-On server is restarted.)
renewcert	CA, CASSL, CASMIME	When OracleAS Certificate Authority is not running, the administrator can use this command to renew the specified certificate, with a prompt for a new validity period, in days. A sample "renewcert" command will thus look like this: `ocactl renewcert -type CA`
revokecert (Revoking CA makes your OracleAS Certificate Authority installation inoperable.)	CA WEBADMIN (Be very careful and certain before taking this action.)	Usable only when OracleAS Certificate Authority is not operating. Revokes the root CA certificate. See "Revoking a Root CA Certificate" for additional reasons specifiable with the CA parameter. A sample "revokecert" command will thus look like this: `ocactl revokecert -type CA -reason SUPERSEDED` Please refer to Table A–5 for details on revocation reasons.

Table A–2 (Cont.) Operations and Parameters of the OracleAS Certificate Authority (OCA) ocactl Tool

Operation	Parameters	Meaning
set	LOG or TRACE, ON or OFF OracleAS Certificate Authority or ADMIN	Sets the OracleAS Certificate Authority configuration to use the additional parameters for state (ON or OFF) or mode (OracleAS Certificate Authority or ADMIN) specified after LOG or TRACE, as follows: Examples of each command appear in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics" at "Log or Trace OracleAS Certificate Authority Actions". The discussion in this Appendix is at "Setting Log/Trace Options".
setpasswd	CA, DB, CASSL, or `CASMIME`	Requests and resets the password for the specified role: administrator, database administrator, certificate authority SSL server, or email encryption. OracleAS Certificate Authority must be stopped before changing passwords. See text for detailed description of the use, setting, and storage of passwords relating to certificate generation and usage. A sample `setpasswd` command will thus look like this: `ocactl setpasswd -type DB`
start	no parameters	Starts the OracleAS Certificate Authorityservice. (OC4J, OHS, and the database must already be in operation for OracleAS Certificate Authority to start. You control OC4J and OHS with the command-line tool `opmn`.) A sample "start" command will thus look like the following: `ocactl start`
status	no parameters	Displays the status of the OracleAS Certificate Authority services. A sample "status" command will thus look like this: `ocactl status`

Table A–2 (Cont.) Operations and Parameters of the OracleAS Certificate Authority (OCA) ocactl Tool

Operation	Parameters	Meaning
stop	no parameters	Stops the OracleAS Certificate Authority service. (Does not stop database, web server, or Oracle Application Server. Relinquishes database connection pool; closes logger, tracer, and configuration data files.) A sample "stop" command will thus look like the following: `ocactl stop`
unlinksso	none	De-registers OracleAS Certificate Authority from the OracleAS Single Sign-On server, so the screens for welcome and enrollment form will not be shown. (This command does not require the OracleAS Certificate Authority service to be shut down, but it won't take effect until the OracleAS Single Sign-On server is restarted.)

updateconnection	no parameters	Writes the connection information stored in Oracle Internet Directory into the OracleAS Certificate Authority configuration file *$ORACLE_HOME/oca/conf/oca.conf*. These strings are used to
		■ connect to the OracleAS Certificate Authority repository and
		■ connect to the directory (used for publishing certificates).
		(This connection information is displayed under Settings in the General subtab of the OracleAS Certificate Authority web interface for the administrator.)
		OracleAS Certificate Authority connection information is originally written to Oracle Internet Directory when Oracle Application Server is installed; this data is then also fetched from Oracle Internet Directory and written into *oca.conf*. This information changes if OracleAS Certificate Authority is moved to another database or if any configuration information changes. Examples include altering nodes or ports in the connection strings, such as adding or removing RAC nodes in a RAC-enabled database. (No data needs to be migrated. If you are initiating a port change, use the proper steps as described in "Changing Infrastructure Ports" in Oracle Application Server Administrator's Guide.)
		Note: You must run `ocactl updateconnection` after any such change to configuration settings, and after using this command, you must restart OracleAS Certificate Authority by issuing the following commands:
		`$ORACLE_HOME/oca/bin/ocactl stop`
		`$ORACLE_HOME/oca/bin/ocactl start`

Converting a CA SSL Server Wallet into SSO Form

Table A–2 shows samples for most of the commands you can issue using `ocactl`. However, the `convertwallet` command uses a different syntax, which this section explains with examples.

You use `ocactl`'s `convertwallet` command to convert a CA SSL server wallet (ewallet.p12, in PKCS#12 format) into Oracle Single Sign-On format, with file name cwallet.sso. The command uses the current CA SSL wallet location, unless you specify a different location.

The advantage to using cwallet.sso is that HTTP Server can be brought up in SSL mode without requiring you to supply the wallet password. Otherwise, when HTTP

Server starts up in SSL mode using a PKCS#12 wallet, the wallet password is requested.

The SSO-format wallet is encrypted to discourage users from visually opening the file and extracting the keys. However, the operating system file permissions are relied upon to protect it, since it is created with owner-only permissions.

Thus the `convertwallet` command enables the OracleAS Single Sign-On server to bring up the web server in SSL mode automatically, without asking a human for the wallet password.

The `convertwallet` command must be run as root user, with *wlt-location* replaced by the desired destination. The command syntax is:

```
convertwallet -format SSO [-walletwrl wallet-location]
```

For example,

```
convertwallet -format SSO -walletwrl $ORACLE_HOME/wallets
```

The optional parameter `-walletwrl` identifies the next parameter as specifying the directory where the CA SSL PKCS#12 wallet is presently located, under the filename `ewallet.p12`.

When `-walletwrl` is specified, `ocactl` assumes the administrator is trying to convert a CA SSL wallet that was not created by OCA, but rather obtained from elsewhere. The administrator must then supply the original CA SSL wallet's password to read the wallet at the specified location, since OCA's password store does not contain that password. Once the wallet is opened, the certificate is converted to .sso format and stored back in the same place specified by `-walletwrl` *wallet-location*.

When -walletwrl is not specified, then ocactl assumes the wallet is the CA SSL wallet generated by OracleAS Certificate Authority during OracleAS Certificate Authority installation. This command therefore uses the OracleAS Certificate Authority administrator's password, already supplied to validate using the ocactl command, to open the internal password store containing the CA SSL password. It then uses this password to open and convert the CA SSL wallet (present at `$ORACLE_HOME/oca/wallet/ssl` directory).

If the destination *wlt-location* is not specified, then by default this wallet is stored in `$ORACLE_HOME/oca/wallet/ssl` (or the location specified during installation).

OracleAS Certificate Authority will use the new OracleAS Single Sign-On wallet stored at `$ORACLE_HOME/oca/wallet/ssl/` only after OHS, OracleAS Certificate Authority's OC4J, and OracleAS Certificate Authority are restarted (in that order). (To start the required infrastructure, see section 4.1 in Oracle Application Server Administrator's Guide. To start middle tier components like OHS and OC4J, see section 4.2.)

Starting the Oracle Certificate Authority Server

After OC4J, OHS, and the database are operating, you can start the OracleAS Certificate Authority services that support the forms for administrator and user access. To start OHS and OracleAS Certificate Authority's OC4J, use `opmnctl` commands in the following order:

```
$ORACLE_HOME/opmn/bin/opmnctl startproc type=oc4j instancename=OracleAS
Certificate Authority
$ORACLE_HOME/opmn/bin/opmnctl startproc type=ohs
```

To start Oracle Application Server Certificate Authority, issue the following command:

```
ocactl start
```

This command requests the administrator password and then starts the Oracle Application Server Certificate Authority engine, creating a database connection pool, logger and tracer files, and initializing configuration.

Stopping the Oracle Application Server Certificate Authority Server

The stop command stops the OracleAS Certificate Authority services. No other services are affected: database, OracleAS, and web server remain unchanged.

To stop the OracleAS Certificate Authority services, issue the following commands to stop OracleAS Certificate Authority's OC4J process and OracleAS Certificate Authority itself:

```
$ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j \
    instancename=OracleAS Certificate Authority
ocactl stop
```

Finding the Status of the Oracle Certificate Authority Services

You can display the status of the Oracle Application Server Certificate Authority services by issuing the status command. It requests the administrator password and then queries the OracleAS Certificate Authority engine. The response shows whether the following facilities are open or closed: the database connection pool; logger, tracer, and the password store.

To get the OracleAS Certificate Authority services status, issue this command:

```
ocactl status
```

Changing Privileged Passwords

Installation creates the Oracle Application Server Certificate Authority password store, which contains the initial passwords required for OracleAS Certificate Authority operations:

Table A–3 Password Types and Uses

Password Type	Password Usage
OracleAS Certificate Authority database user	Enables access to database tables containing OracleAS Certificate Authority information
CA SSL	Enables the Certificate Authority to communicate using SSL. It also allows this wallet to be accessed by Oracle Wallet Manager to add trust points, and so on. On install, a randomized password is used to encrypt the wallets. You can use this to set it to a known password, so that it can be opened by Oracle Wallet Manager.

The contents of this password store are encrypted using the OracleAS Certificate Authority administrator's password, which is also the CA signing password. This password is not stored in the password store.

At some point after installation, you can change the password for any of the following privileged operations for different types of administrators. Use the `setpasswd` command in the `ocactl` tool as follows:

Table A–4 Privileged Roles and the setpasswd Command

Privileged Role	Command to Change Password for Role	New Password Also Used As
Oracle Application Server Certificate Authority Administrator	`ocactl setpasswd -type CA`	Certificate Authority signing password
CA SSL server	`ocactl setpasswd -type CASSL`	CA SSL wallet password
Oracle Application Server Certificate Authority database administrator	`ocactl setpasswd -type DB`	Password for DB in the database; used by OracleAS Certificate Authority to log in to the database.
Administrator signing notification mails	`ocactl setpasswd -type CASMIME`	CA SMIME password in the password store

OracleAS Certificate Authority must be stopped before any password can be changed.

The changes resulting from executing these commands take effect after the next start of OracleAS Certificate Authority. After OracleAS Certificate Authority is restarted, the new passwords will be in effect.

Each use of `ocactl` requires the OracleAS Certificate Authority administrator password. Once this is authenticated, the command requests the new password for the role type specified in the command, which then replaces the one in the password store. The results are again encrypted using the latest OracleAS Certificate Authority administrator password.

Regenerating the Root Certificate Authority's Certificate

When installing OracleAS Certificate Authority as a root certificate authority (CA), the Root CA certificate and wallet are created. If the CA key is somehow compromised, this certificate can be regenerated using the `ocactl` administrative command line tool. The new CA certificate and private key will be stored in the OracleAS Certificate Authority repository. The private key is encrypted by the password that was requested during its generation.

The former CA signing certificate entry and all other certificates issued by that former CA signing certificate will become invalid. Critical wallets like CA SSL, CA SMIME need to be regenerated again. After re-generation of the CA signing wallet, a CRL issued by the old CA will not be useful.

See Also:

- Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics", specifically the sections entitled "Regenerating the CA Signing Wallet" and "Regenerating the CA SSL and CA S/MIME Wallets".
- For general information on SMIME, see Appendix G, "S/MIME with OracleAS Certificate Authority"

Regenerating the CA signing wallet and other critical wallets can only be done after OracleAS Certificate Authority is successfully installed and OracleAS Certificate Authority service is not running.

The root CA signing wallet can be generated only when OracleAS Certificate Authority is not running. If OracleAS Certificate Authority is running, stop OracleAS Certificate Authority and use this command to regenerate the Root CA signing wallet:

```
ocactl generatewallet -type CA
```

This certificate is stored in the OracleAS Certificate Authority repository.

The signing key is also stored in the OracleAS Certificate Authority repository, encrypted by the OracleAS Certificate Authority administrator password.

The password store is kept in the directory `$ORACLE_HOME/oca/pwdstore`, encrypted with the Administrator's password. The DB password is initially the same as the Administrator's password.

See Also: "Remembering and Restoring the Metadata Repository Password" in Appendix C, "Troubleshooting OracleAS Certificate Authority"

Regenerating the Certificate Authority's SSL Certificate and Wallet

The CA SSL certificate and wallet are generated during installation and are used to enable the OracleAS Certificate Authority engine to listen in HTTPS mode. If these are compromised or corrupted, or the CA signing wallet is regenerated, you must regenerate them in order to reestablish secure communications.

The CA SSL wallet can be generated only when OracleAS Certificate Authority is not running. If it is running, stop OracleAS Certificate Authority and use this command to regenerate the CA SSL certificate and wallet:

```
ocactl generatewallet -type CASSL
```

This wallet is stored in the directory `$ORACLE_HOME/oca/wallet/ssl`, encrypted by the password that was requested during its generation.

This command also generates CA SSL wallet in OracleAS Single Sign-On format and stores it as `cwallet.sso` at $ORACLE_HOME/oca/wallet/ssl.

Revoking a Root CA Certificate

Revoking a root CA certificate is a very drastic operation, which will make OracleAS Certificate Authority installation non-functional and invalidate the certificates already issued. This operation, revocation, should only be done when the CA key is compromised, so that you can install a new certificate authority.

The `revokecert` command enables you to revoke a root certificate authority certificate or an OracleAS Certificate Authority Administrator's certificate. It can only be used when OracleAS Certificate Authority is not operating.

Revoking a root certificate authority certificate is required before installing a new root CA for ongoing OracleAS Certificate Authority operations.

When you intend to install a new CA, use `revokecert` first to revoke the old CA signing wallet, giving the reason as a parameter. If the root CA certificate is revoked, all certificates issued by that CA will be in an inconsistent state. So before revoking the root CA certificate, first revoke all certificates issued by the existing CA and update the Certificate Revocation List. Otherwise, while the new CA signing certificate is being generated, all the old certificates signed by the old CA will be marked as Invalid in the OracleAS Certificate Authority repository.

Once the OracleAS Certificate Authority administrator certificate is revoked, the administrator cannot access any administrative functions on the web until he gets a new certificate. When he opens the Administration home page, it will require a new enrollment to get a new Administrator's certificate.

Revoking a root certificate authority certificate requires that you first stop OracleAS Certificate Authority. Then issue the following command:

```
ocactl revokecert -type CA -reason why
```

Since the primary reason for revoking a CA certificate is a compromised key, the actual command would be as follows:

```
ocactl revokecert -type CA -reason KEY_COMPROMISE
```

If other circumstances require a revocation, you can replace the *why* entry with whichever one of the following eight phrases is most appropriate:

Table A–5 Revocation Reasons for Use with `revokecert` ***Command***

Revocation Reason	Explanation
`AFFILIATION_CHANGE`	The organization has decided to use a different root CA.
`CA_COMPROMISE`	There may be reason to distrust the root CA (for example, the CA may key may be compromised), so a new CA is required.
`CERTIFICATE_HOLD`	The certificate is being held due to some suspicions.
`CESSATION_OF_OPERATION`	The present root CA has ceased operations, so a new CA is required.
`KEY_COMPROMISE`	The user's private key has been compromised.
`REMOVE_FROM_CRL`	Certificate status will be REVOKED, but this revoked certificate will not be added to the CRL.
`SUPERSEDED`	The root CA's certificate has been replaced. The old one must be removed and the new one installed.
`UNSPECIFIED`	No reason is available or has been given. This is the default reason.

Generating a Sub CA Signing Wallet from OracleAS Certificate Authority

You can generate a Sub CA signing wallet from OracleAS Certificate Authority as follows:

1. Create a new wallet and generate a certificate request using Oracle Wallet Manager or another similar tool.

   See also: *Oracle Advanced Security Administrator's Guide*

2. Using the Server/Sub CA enrollment form, submit the PKCS10 request and select certificate usage as CA signing.
3. Using the OracleAS Certificate Authority Administrative form, issue a Sub CA certificate. Please specify the path-length, that is, the number of levels of Sub CAs that it can have.
4. Go to the Server/Sub CA enrollment form and click **Down CA Certificate**, which will show the CA certificate along with the its ancestors, if there are any.
5. Copy the BASE64 certificate of the CA from the screen and import it as a Trusted certificate into Oracle Wallet Manager. If there are any trust points along with the CA, copy one by one into Oracle Wallet Manager using its **Import Trusted Certificate** option.
6. Using the Server/Sub CA enrollment form, get certificate details by giving the serial number or the common name of the Sub CA. Click **View Details** to view the Sub CA certificate in BASE64 format.

7. Copy the BASE64 format of the Sub CA certificate and import it into Oracle Wallet Manager as a user certificate.
8. Save the Sub CA signing wallet using Oracle Wallet Manager. The wallet will be stored as `ewallet.p12`.

Installing/Importing a Sub CA Signing Wallet

The steps in this section enable you to install and use a Sub CA signing wallet, creating a hierarchy of CAs. This wallet can be generated from OracleAS Certificate Authority, as in "Generating a Sub CA Signing Wallet from OracleAS Certificate Authority", a self-signed wallet created with the `mkwallet` or `orapki` utilities, or come from any X.509v3-compliant CA, such as CMS.

> **See Also:** To import an SSL wallet from any X.509 v3 CA, please follow the instructions for configuring the Oracle HTTP Server, as described in the OracleAS Security Guide. Also see the discussion of Oracle Wallet Manager in the *Oracle Advanced Security Administrator's Guide*.

Before importing a Sub CA signing wallet, you must

- Install OracleAS Certificate Authority successfully, which will create its repository, the password store, the Root CA signing wallet, and the CA SSL wallet.
- Ensure that the wallet is free of corruption and contains one or more self-signed certificates. You can verify a wallet with the `orapki wallet display` command.

Then take the following steps:

1. Stop OC4J and OHS if they are running, using these commands:

```
$ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=OracleAS
Certificate Authority
$ORACLE_HOME/opmn/bin/opmnctl stopproc type=ohs
```

2. Use the `ocactl importwallet` command to install the Sub CA signing wallet:

```
ocactl importwallet -type SUBCA
```

The command prompts for the administrator's password, for the directory where the wallet for the new Sub CA (`ewallet.p12`) is stored, and for that wallet's password. It then fetches the new CA's certificate and private key from that wallet, and stores them in the OracleAS Certificate Authority repository. The password used for the new CA's wallet, provided in response to the command prompts, is the new CA's signing password. This password now becomes the password of the OracleAS Certificate Authority Administrator.

See Appendix B, "Setting up a CA Hierarchy" for further detailed description in the present manual.

The former root CA certificate entry and all other certificates issued by that former root CA will become invalid. The old CA certificate and key in the OracleAS Certificate Authority repository will be overwritten by the new Sub CA certificate and key, respectively. The new Sub CA certificate's entry and Serial number will be added to the repository so that certificates issued by this Sub CA will have serial numbers greater than the serial number of the Sub CA certificate. Also, any administrator certificate issued by the old CA is removed from the password store. While importing

Sub CA signing wallet, `ocactl` ensures that the correct bits are set for BasicConstraintsExtension and KeyUsageExtensions to be DIGITAL_SIGNATURE, KEY_CERT_SIGN, CRL_SIGN and NON_REPUDIATION. Otherwise, if these extensions are not set, the wallet will not be accepted as Sub CA signing wallet.

Generating a CA SSL Wallet for a Sub CA

As is described in "Regenerating the Certificate Authority's SSL Certificate and Wallet", the CA SSL certificate and wallet are generated during installation. They enable OracleAS Certificate Authority to listen in HTTPS mode, and they can be regenerated if they become compromised or corrupted, in order to reestablish secure communications.

Generating the Sub CA SSL wallet is also done when OracleAS Certificate Authority is not running, using this command:

```
ocactl generatewallet -type CASSL
```

This wallet is signed by the Sub CA and stored in the directory `$ORACLE_HOME/oca/wallet/ssl`, encrypted by the password requested during its generation.

As root user, you can convert this wallet to OracleAS Single Sign-On format using the command

```
ocactl convertwallet -format SSO
```

Once you install a Sub CA, the earlier CA that issued the SSL certificate no longer exists. Clients connecting to OracleAS Certificate Authority will trust the current CA certificate. The CASSL issued by the previous CA is not trusted, so you should regenerate the CASSL certificate after importing Sub CA or after a CASSL wallet is corrupted or compromised.

After generating this CA SSL certificate and wallet, do the following steps:

1. Start HTTP Server.
2. Start OC4J.
3. Start OracleAS Certificate Authority.

OracleAS Certificate Authority will now use the Sub CA certificate for signing certificate requests.

Clearing Log or Trace Storage

The administrative command line tool enables removal of existing log or trace files at the administrator's choice. The clear command has the following format:

`ocactl clear -type {LOG |TRACE}` -mode {OCA | ADMIN}

The possible commands are

- `ocactl clear -type LOG -mode ADMIN`
- `ocactl clear -type TRACE -mode ADMIN`
- `ocactl clear -type LOG -mode OCA`
- `ocactl clear -type TRACE -mode OCA`

The result of each such command is to remove the corresponding log or trace data: clearing log data removes it from the OracleAS Certificate Authority repository; clearing trace data removes the file `oca.trc` from `$ORACLE_HOME/oca/logs`.

Updating OracleAS Certificate Authority Repository Connection Information

The connection information used for publishing certificates is displayed under Settings in the General subtab of the OracleAS Certificate Authority web interface for the administrator. This information includes the connection strings that OracleAS Certificate Authority uses to connect to its repository and to Oracle Internet Directory.

The `ocactl` command `updateconnection` writes the connection information into the OracleAS Certificate Authority configuration file `$ORACLE_HOME/oca/conf/oca.conf`.

> **See Also:** See changesecurity, clear, generatewallet, help, importwallet, linksso, renewcert, revokecert, set, setpasswd, start, stop, unlinksso, and updateconnection in Table A–2.

OracleAS Certificate Authority connection information is originally written to Oracle Internet Directory when Oracle Application Server is installed, when it is also fetched from Oracle Internet Directory and written into `oca.conf`. This information changes if OracleAS Certificate Authority is moved to another database.

Setting SSO Authentication (linksso, unlinksso commands)

Single Sign-on authentication facilitates fast access to resources and applications, and is even more rapid and efficient when certificates are used in place of username and password.

OracleAS Certificate Authority has an expedited process to enable OracleAS Single Sign-On Server-authenticated users to request and receive such certificates.

When the OracleAS Certificate Authority administrator executes the `ocactl linksso` command, it registers OracleAS Certificate Authority with the OracleAS Single Sign-On server to display the certificate authority's certificate enrollment form to OracleAS Single Sign-On users who lack a certificate. Using the short process thus presented, such users can request a certificate, which OracleAS Certificate Authority then issues, and the user can import it into the browser for future authentication.

All aspects of this process are discussed in Chapter 4, "Introduction to Administration and Certificate Management", in the section titled "Single Sign-on and OracleAS Certificate Authority". An overview appears in Chapter 7, "OracleAS Certificate Authority Administration: Advanced Topics", in the section "OracleAS Certificate Authority and High-Availability Features".

`The ocactl linksso` command does not require OracleAS Certificate Authority service to be shut down, but it takes effect only after the OracleAS Single Sign-On Server is restarted.

Setting Log/Trace Options

The administrator can initiate logging and tracing operations with the `ocactl set` command, specifying which type of data is desired and turning its generation on or off. The forms of the command are as follows:

- `ocactl set -type LOG -mode OCA  -state ON`
- `ocactl set -type LOG -mode ADMIN -state ON`

- `ocactl set -type TRACE -mode OCA -state ON`
- `ocactl set -type TRACE -mode ADMIN -state ON`

Data generated by these commands is stored in the following locations:

- OracleAS Certificate Authority LOG data goes into the OracleAS Certificate Authority repository.
- ADMIN LOG data goes into the `admin.log` file at `$ORACLE_HOME/oca/logs`.
- TRACE data goes into one of two files at `$ORACLE_HOME/oca/logs`:
  - OracleAS Certificate Authority trace goes to `oca.trc`, that is, `$ORACLE_HOME/oca/logs/oca.trc`.
  - ADMIN trace goes to `admin.trc`, that is, `$ORACLE_HOME/oca/logs/admin.trc`.

The OFF commands stop the process of generating LOG or TRACE data. Data already collected remains in the indicated locations until an `ocactl clear` command is issued. Files affected by such `ocactl clear` commands (`oca.trc, admin.trc,` or `admin.log`) are simply removed from the file system.

B

Setting up a CA Hierarchy

This Appendix describes how to acquire and import a subordinate certificate authority, which is a CA whose certificate is signed by some higher CA authority. This Sub CA could be authorized by the original Oracle Application Server Certificate Authority installed at a corporate headquarters, for use in a remote division. Or the new Sub CA could be authorized by (signed by) an entirely different certificate authority with a hierarchy and root different from OracleAS Certificate Authority.

The following summary gives an overview of the acquisition and import process:

As the administrator of OracleAS Certificate Authority, you obtain the Sub CA signing wallet and certificate by using Oracle Wallet Manager (OWM), or any similar third party mechanism. The first step is to generate a PKCS#10 certificate request, usually by filling in a form. OWM uses the completed form to create the request, which is an encrypted body of text containing all the supplied information necessary to authenticate the requesting entity.

> **See Also:** *Oracle Advanced Security Administrator's Guide*

You then copy this request from the OWM interface and paste it into the Certificate Issuance interface provided by the third party, receiving a certificate request ID. This ID can be used to fetch and display the BASE64 format certificate when it is issued. For other CAs, follow the CA-specific procedures. In some cases, the certificate is sent to your mail ID.

Once the certificate is received, use OWM to import it as a user certificate and add the CA that issued it as a trust point. After the certificate is approved, OWM stores it in a PKCS#12-format wallet that can then be used as a Sub CA signing wallet.

OracleAS Certificate Authority's administration tool has an import option to enable the administrator to import that stored SubCA signing wallet and certificate into an OracleAS Certificate Authority instance running as a Subordinate CA. The import operation includes an automatic change of encryption and location to fit OracleAS Certificate Authority's standard operations. The following sections of this Appendix describe these steps:

- Generating a Sub CA Signing Wallet
- Installing and Using the New Sub CA Signing Wallet
- Generating CA SSL and CA SMIME Wallets for a Sub CA

Generating a Sub CA Signing Wallet

The following steps tell you, as OracleAS Certificate Authority administrator, how to generate a Sub CA signing wallet:

1. Use Oracle Wallet Manager or a third-party tool to generate a PKCS#10 request.
2. Using OracleAS Certificate Authority's Server/Sub CA enrollment form, submit the PKCS#10 request and select **CA Signing** as the certificate usage.

   See Also: "Server/SubCA Certificates Tab" in Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority"

3. Using the OCA Administration form, issue the Sub CA certificate. (If a third party enrollment was used, await certificate notification.)

   See Also: "Approving or Rejecting Certificate Requests" in Chapter 4, "Introduction to Administration and Certificate Management"

4. After approving that certificate (or receiving approval notification from the third-party issuer, if you used one), go to the Server/Sub CA enrollment form and click **Save CA Certificate**. An **Advanced** button will appear. Clicking **Advanced** will show the CA certificate along with the trust points, if any, displayed under the CA chain in PKCS#7 format.
5. Copy the BASE64 certificate of the CA from the screen, go to Oracle Wallet Manager, and import that certificate as a Trusted certificate into OWM. If there are any trust points along with the CA, copy them one by one into Oracle Wallet Manager, using OWM's **Import Trusted Certificate** option.

   See Also: *Oracle Advanced Security Administrator's Guide*

6. Using the Server/Sub CA enrollment form, get the certificate details by giving the serial number or the common name of the Sub CA. Click **View Details** to view the Sub CA certificate in BASE64 format.
7. Copy the BASE64 format of the Sub CA certificate and import it into OWM as a user certificate.
8. Use OWM to save the Sub CA signing wallet to a file destination of your own choice.

Installing and Using the New Sub CA Signing Wallet

The steps in this section enable you to create a hierarchy of CAs. The wallet for the new Sub CA can be generated by OCA or by any X.509v3-compliant CA. It should be created through Oracle Wallet Manager immediately after the install and before any certificates are issued. Otherwise, such certificates become invalid after the new Sub CA is installed. Examples of third-party suppliers include iPlanet's Certificate Management System (CMS), Verisign, or others. To use a third party certificate, the certificate must conform to the extension requirements of OracleAS Certificate Authority as described in Appendix D, "Extensions".

See Also: "Subordinate CA Certificates" in Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority".

1. Install Oracle Application Server Certificate Authority, which will create an OracleAS Certificate Authority repository, create the password store, and create the Root CA signing wallet and the CA SSL wallet.

Note: The OracleAS Certificate Authority schema in one repository can only be used with one OCA.

When installing another OracleAS Certificate Authority instance, you must not choose a repository that has been used to install an earlier OCA: the OCA configuration tool will fail.

This failure will force you to exit and restart the whole installation.

---

2. Stop OC4J and Oracle HTTP Server (Apache) if they are running, using these commands:

```
$ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=oca
$ORACLE_HOME/opmn/bin/opmnctl stopproc type=ohs
```

3. Install the Sub CA signing wallet using the following command:

```
ocactl importwallet -type SUBCA
```

See Also: Appendix A, "Command-Line Administration" for details. For example, while importing the Sub CA signing wallet, `ocactl` ensures that the correct bits are set for the right extensions. The wallet can function as a Sub CA signing wallet only if the correct bits are set. BasicConstraintsExtension must show DIGITAL_SIGNATURE. KeyUsageExtensions must show KEY_CERT_SIGN ("Certificate Signing"), CRL_SIGN and NON_REPUDIATION: all three must be present.

Note: If importwallet gives an error message, import the certificate into your browser and view its details to see the error, which in Internet Explorer will be that one of those two subject types will fail to have the indicated necessary terms.

Installing the Sub CA signing wallet will:

a. Prompt for the existing administrator's password, for the directory where the wallet for the new Sub CA (ewallet.p12) is stored, and for that wallet's password.

   The password used for the new CA's wallet, provided in response to the command prompts, is the new CA's signing password. This password now becomes the password of the OracleAS Certificate Authority Administrator.

b. Fetch the new Sub CA's certificate, private key, and serial number from that wallet, and store them in the OracleAS Certificate Authority repository.

   This operation overwrites the corresponding earlier records in the OracleAS Certificate Authority repository. Thus, the new Sub CA certificate, key, and password replace the old root CA certificate, key, and signing certificate password, respectively.

c. Update the current Serial number of the Sub CA certificate, so that certificates issued by this Sub CA will have serial numbers greater than the serial number of the Sub CA certificate. Also, any administrator certificate issued by the old CA is removed from the password store.

At this point, you must do the following steps, as root user:

1. Generate a new CA SSL wallet, since the existing CA SSL was signed by the prior CA. Use the following command

   ```
   ocactl generatewallet -type CASSL.
   ```

   This generated CA SSL wallet will be signed by the new Sub CA certificate

2. Convert this wallet to OracleAS Single Sign-On format using the following command

   ```
   ocactl convertwallet -format SSO
   ```

3. Start HTTP Server by using the command-line tool `opmn`.
4. Start OC4J using the same command-line tool.
5. Start OracleAS Certificate Authority, which will now use the new Sub CA certificate for signing all future certificate requests.

See Also: *Oracle Advanced Security Administrator's Guide*

Configuring an OracleAS Certificate Authority Instance to Be a Subordinate CA of Another CA

When a huge organization has multiple geographical locations, it can be useful to get a Sub CA signing wallet from the Root CA and install that Sub CA in another OracleAS Certificate Authority installation. The parent organization with the Root CA signing wallet can issue Sub CA signing wallets to each subordinate organization or department. Each such Sub CAs will act as the Certificate Authority CA in its respective location to manage certificates specific to that organization. Preventing a Sub CA from issuing another Sub CA signing wallet can be done by setting the path length when that Sub CA's wallet is issued by Root CA.

The following steps enable you to generate and use a Sub CA signing wallet from OracleAS Certificate Authority:

1. Create a new wallet and generate a PKCS#10 certificate request using Oracle Wallet Manager (OWM). Copy the request for submission to OracleAS Certificate Authority.

   See Also: *Oracle Advanced Security Administrator's Guide*

2. Using the Server/Sub CA enrollment form of the user interface described in Chapter 8, paste in the PKCS#10 request you generated with OWM and select certificate usage as CA signing.
3. Using the OracleAS Certificate Authority Administrative form in the administrative interface described in Chapter 4, issue a Sub CA certificate. Specify its path-length, that is, the number of levels of Sub CAs that it can have.
4. After that approval, go back to the Server/Sub CA enrollment form and click **Save CA Certificate**, which will show the CA certificate along with its ancestors, if there are any.

   See Also: "Server/SubCA Certificates Tab" in Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority"

5. Click **Advanced** to show the BASE64-encoded certificates.

6. Copy the BASE64 certificate of the CA from the screen and import it as a Trusted certificate into Oracle Wallet Manager. If the CA is a subordinate CA in a hierarchy of CA's, all the CA's in the hierarchy must be imported into OWM. Copy them one by one into Oracle Wallet Manager using its Import Trusted Certificate option.

At this point you must copy the details of the certificate into OWM and then save that wallet, using the following steps:

1. Using the Server/Sub CA enrollment form, use the serial number or the common name of the Sub CA to find this particular certificate.
   a. To use the serial number, click its radio button on the left to select it and then click the hypertext link on the right, to display it.
   b. To use the common name, you enter it, click **Go**, and select the desired certificate from those listed.
2. Click **View Details** to view the Sub CA certificate in BASE64 format.
3. Copy that BASE64 format of the Sub CA certificate and import it into Oracle Wallet Manager as a user certificate.
4. Save the Sub CA signing wallet using Oracle Wallet Manager. The wallet will be stored as ewallet.p12.

Generating CA SSL and CA SMIME Wallets for a Sub CA

As described in Chapter 7 in the section entitled "Regenerating the CA SSL and CA S/MIME Wallets", the CA SSL wallet is generated during installation. It enables OracleAS Certificate Authority to listen in HTTPS mode, and it can be regenerated if necessary, to reestablish secure communications. Circumstances requiring such regeneration include a wallet becoming compromised or corrupted, or the CA signing wallet being regenerated, or a new Sub CA certificate being imported.

Generating the Sub CA SSL wallet is also done when OracleAS Certificate Authority is not running, using this command:

```
ocactl generatewallet -type CASSL
```

This wallet is signed by the Sub CA and stored in the directory `$ORACLE_HOME/oca/wallet/ssl`, encrypted by the password requested during its generation.

Once you install a Sub CA, the earlier CA that issued the SSL certificate no longer exists. Clients connecting to OracleAS Certificate Authority will trust the current CA certificate. The CA SSL issued by the previous CA is not trusted, so you should regenerate the CA SSL certificate after importing a Sub CA or after a CA SSL wallet is corrupted or compromised.

Similarly, after importing a Sub CA, the CA SMIME wallet previously issued by the prior CA is no longer valid. The CA SMIME wallet must be generated to sign alerts and notifications when "Send SMIME E-Mails" is enabled in the Notification page of Configuration Management in OracleAS Certificate Authority Admin page. Use this command to generate the CA SMIME wallet:

```
ocactl generatewallet -type CASMIME
```

After generating the CA SSL and CA SMIME wallets, do the following steps:

1. Start OC4J and HTTP Server.
2. Start OracleAS Certificate Authority.

OracleAS Certificate Authority will now use the Sub CA certificate for signing certificate requests.

C

Troubleshooting OracleAS Certificate Authority

This appendix describes common problems that you might encounter when using OracleAS Certificate Authority and explains how to solve them. It contains the following topics:

Problems and Solutions

This section describes common problems and solutions. It contains the following topical groups:

Prerequisite Issues and Warnings

This section describes certain issues that need to be addressed before further progress in using OracleAS Certificate Authority can go forward, and are therefore termed "prerequisite":

Key Pair Generation Fails during Certificate Requests on Windows

Problem

For Windows client machines, this operation requires NT to have Service pack 5 or higher.

Solution

Visit Microsoft's Web site and download the necessary upgrades for your configuration.

Cannot Log in as Administrator after Logging in as Normal User

Problem

If you first log in to OracleAS Certificate Authority as a normal user through SSL, then trying to go to Certificate Management causes a JAZN error. The reason is that you are not recognized as the web administrator unless you log in as such, even though you are enrolled as the web administrator. The SSL session established between OracleAS Certificate Authority and you as a non-administrative user remains active; your enrollment does not change your SSL session.

Solution

To log in as web administrator, you must

1. Enroll as web administrator if you do not have a web administrator certificate
2. Exit your browser, and
3. Log in as web administrator, by choosing your web administrator certificate for authentication.

For more information, see Chapter 5, "Configuring Oracle Application Server Certificate Authority".

Note: This login issue is due to a Netscape browser problem.

Changing Passwords Requires OracleAS Certificate Authority's Command-line Tool ocactl

Problem

OracleAS Certificate Authority uses passwords for a number of tasks; for example, there are passwords for the CA SSL wallet, the internal metadata repository, and the OracleAS Certificate Authority administrator. It may occasionally be desirable or advisable to change a password. Generally speaking, if any tool other than `ocactl` is used to change any of these passwords, OracleAS Certificate Authority will stop working.

For example, if the metadata repository password is changed outside OracleAS Certificate Authority, that is, by using a tool other than ocactl, then OracleAS Certificate Authority will not start up.

Solution

The following discussion examines the implications of changing passwords outside OracleAS Certificate Authority.

OracleAS Certificate Authority's Metadata Repository Password

The OracleAS Certificate Authority metadata schema password is initially set (at install time) to be the same as the administrator password, but either password can be

changed independently with the ocactl setPassword –type DB command and the ocactl setPassword –type CA command. As mentioned earlier, if this password is changed outside of OracleAS Certificate Authority (that is, not using the ocactl tool), then OracleAS Certificate Authority will not start up. This circumstance also prevents you from resetting the repository password with ocactl. To resolve this, you must log in to the database as any DBA, such as SYS or SYSTEM, and change the password back to its original value.

For additional information about this password, see "Remembering and Restoring the Metadata Repository Password".

OracleAS Certificate Authority's Administrator Password

The administrator password cannot be changed outside OracleAS Certificate Authority.

OracleAS Certificate Authority's SSL Password

The OracleAS Certificate Authority SSL password (the password for the SSL wallet, which is in *oca/wallet/ssl*) should only be changed using `ocactl`. Changing this password with Oracle Wallet Manager will disable OracleAS Certificate Authority because the changed password is no longer reflected in the OracleAS Certificate Authority password store. However, you can recover from this situation by using `ocactl setpasswd CASSL` to reset the SSL password.

OracleAS Certificate Authority's S/MIME Password

The OracleAS Certificate Authority S/MIME password (the password for the SMIME wallet, which is stored in the database, not on the file system) cannot be changed using Oracle Wallet Manager. You can only change it through ocactl.

OracleAS Certificate Authority's *Oracle Internet Directory* Password

This is a randomly generated password. It cannot be changed through ocactl. But if it is altered using the Oracle Internet Directory administration tool, OracleAS Certificate Authority will not be able to talk to Oracle Internet Directory as it does not know the new password.

WARNING: Generally speaking (subject to the rules mentioned in the preceding discussion), always use `ocactl` to change any password related to OracleAS Certificate Authority. Never use any other tool; OracleAS Certificate Authority will stop working.

Remembering and Restoring the Metadata Repository Password

Problem

Complex sites with separate administrators for different functions, components, or organizations can sometimes encounter conflicts. For example, a database administrator can change the password for the OracleAS Certificate Authority metadata repository (schema) without realizing that this should only be done through OracleAS Certificate Authority itself. This change prevents OracleAS Certificate Authority from working.

Solution

Understanding the following scenarios can aid in preventing or resolving such a conflict:

1. If the DB password in the password store has never been changed from the default (which happens to be `OCA-admin-password` as established during

installation), then regaining access to the database (after someone changed the password originally recognized by the repository) can be accomplished by this command:

```
alter user OracleAS Certificate Authority identified by OCA-admin-password
```

This resetting of the repository password to the `OCA-admin-password` causes it to match what is in the password store as the repository password.

2. If the DB password in the password store has been changed and the OracleAS Certificate Authority administrator does know what it is (for example, `new_DB_pswd_in_store`, then if the repository password is changed (by a database administrator, perhaps), the OracleAS Certificate Authority administrator can restore database accessibility by using the command:

   ```
   alter user OracleAS Certificate Authority identified by new_DB_pswd_in_store
   ```

3. If the DB password in the password store has been changed and the OracleAS Certificate Authority administrator does not know (or remember) what it is, changing the *repository* password will prevent OracleAS Certificate Authority operations. Here's why: database access will not be granted unless the password offered by OracleAS Certificate Authority for the password store matches the current repository password. If the repository password is changed, then either that password or the DB password in the password store must be changed so that they again match. Since the DB password in the password store is unknown, the administrator cannot supply it in an "alter user" command. Nor can she change the DB password in the password store, because `ocactl` requires the current DB password before allowing it to be changed. So no recovery is possible. The unknown DB password remains unchangeable.

These resolutions all rely on the OracleAS Certificate Authority administrator retaining the privileges necessary to invoke alter user oca.

Using ocactl raises "Error:Password store missing" message

Problem

When Oracle Application Server 10g was originally installed, the option to install OracleAS Certificate Authority was not selected. Consequently no password file was created, and it cannot be created after the fact in the original Oracle home. The majority of OracleAS Certificate Authority files do get installed, but OracleAS Certificate Authority is unusable since it was not installed and configured during the original Oracle Application Server 10g installation.

Solution

Install a new instance of OracleAS Certificate Authority in a new Oracle home. It can be installed:

- on the same computer as the OracleAS Infrastructure
- on a different computer
- with its own OracleAS Metadata Repository
- against an existing OracleAS Metadata Repository.

As explained in the following discussion, practical considerations determine how these options are combined.

Installing OracleAS Certificate Authority only

In this case, OracleAS Certificate Authority will share the previously installed OracleAS Metadata Repository. If you are installing OracleAS Certificate Authority on the same computer as the OracleAS Infrastructure instance, sharing the repository is preferable for performance reasons.

Installing OracleAS Certificate Authority with its own OracleAS Metadata Repository

If you are installing OracleAS Certificate Authority with its own repository, it is preferable to install it on a separate computer from the OracleAS Infrastructure; otherwise you would need to run two databases on the same computer, which could degrade performance.

References

- *Oracle Application Server Installation Guide*, Section 6.23, "Installing Identity Management Components Only (Excluding Oracle Internet Directory)"
- *Oracle Application Server Installation Guide*, Section 15.6, "OracleAS Certificate Authority Topology"

Browser Issues

This section describes these known browser-related issues:

- Browser issues a warning if the CA SSL Server's CN does not match the machine name
- Certificate list shows all users as "Users"
- Netscape/Mozilla Issues
- Internet Explorer (IE) Issues

> **Note:** These issues are explicitly related to browsers and occur only when you are using a certain type or level of browser. Unless stated otherwise, they can typically be resolved within the browser itself; contact the browser vendor for assistance if necessary.

Browser issues a warning if the CA SSL Server's CN does not match the machine name

The machine name is likely used widely and inconvenient to change. Therefore, the CN for the CA SSL Server must be made identical to that machine name, requiring a new certificate.

Certificate list shows all users as "Users"

Problem

When a DN has more than one CN component, the browser names the certificate for that DN using only its first CN component (from the right). Consequently, the popup display for SSL Mutual Authentication lists all the certificates as "users" (in both MicroSoft Internet Explorer and Netscape/Mozilla), making it impossible to distinguish different users.

Solution

You can identify the user and obtain additional details by viewing the certificate.

Netscape/Mozilla Issues

The following issues affect only Netscape clients:

"Certificate is expired" warning appears

Problem

If the time zone of the client is behind that of the server, there can be a period of time in which Netscape/Mozilla might issue a 'certificate is expired' warning. The reason is that the CASSL certificate is not yet valid in the user's time zone.

Solution

The problem should resolve itself in a relatively short period of time, depending on the time zone differential.

SubCA and CA SSL client certificates are listed

Problem

If the user has two SSL client certificates, one from the CA and another from a SubCA of that CA, then during client authentication to the SubCA, both certificates are listed.

Solution

Select the certificate appropriate to the CA in use for this SSL site.

Internet Explorer (IE) Issues

The following issues affect only Internet Explorer clients:

Failure to import CRL to Browser

Problem

The Internet Explorer Import... button does show the CRL for viewing, but it does not actually install the CRL into the browser.

Solution

Save the CRL to disk and use the following Internet Explorer menu command sequence: Tools -> Internet Options -> Content -> Certificates -> Import. This brings you to the Certificate Import Wizard; follow the steps indicated by the wizard to complete the import.

Message that a page contains both secure and non-secure information

Problem

In User Pages -> Manual Authentication -> Save CA certificate -> Advanced, clicking **Help** opens a new window that may display an error message saying that the page contains both secure and non-secure information. This is not a security breach.

Opening online Help can generate a security alert

Problem

When online help is opened while using OracleAS Certificate Authority, IE will display a security alert. It appears that the alert is generated whenever an https URL is in use and then a second https URL is invoked.

Solution

This behavior can be switched off by changing the security options under Tools -> Internet Options -> Security -> Custom Level. Under Settings, look for "Display Mixed Content" and select the enable option under that heading.

Message about generating an excessive number of certificate requests

Problem

Sometimes after generating many certificate requests using Internet Explorer, an additional dialog box may appear containing such a message.

Solution

You can continue by clicking "Yes", indicating you are generating certificate requests to a certification authority.

You can remove excess certificate requests using the instructions in the online Microsoft Internet Explorer guide, in the section "Deleting a Certificate Request".

VBScript error when importing a certificate You may encounter the following VBScript error message when attempting to import a user certificate to the browser:

```
Failed to import certificate. Check your browser repository.
Please contact Administrator.
```

This error occurs if an incorrect certificate key store was specified when submitting the request.

Solution

When requesting a new certificate on Internet Explorer, specify the correct key store, for example Microsoft Enhanced Cryptographic Provider v1.0. The Key Store choices presented on the certificate request screen vary, depending on the browser and the existence and type of smart card service on the machine where the certificate was requested. See "User Certificates Tab" in Chapter 8 for details.

Network Issues

The following network-related messages or issues may arise during OracleAS Certificate Authority operation:

- Error message when logging on to OracleAS Certificate Authority using SSO username/password
- "Network Error" message
- OracleAS Certificate Authority Stops Working, or Network/Server Messages Appear

Error message when logging on to OracleAS Certificate Authority using SSO username/password

Problem

The following message:

```
"Forbidden
You don't have permission to access /oca/sso/ssoInitServlet on
this server"
```

arises from an IP address check if a proxy server with multiple IP addresses is used between the browser and the OracleAS Single Sign-On server.

Solution

- When the access is through an intranet, the browser should be configured not to use a proxy, following the instructions in the browser documentation.
- If this is not the case, or if such a change does not solve the problem, then the value of the OssoIpCheck directive in the OracleAS Single Sign-On configuration file must be set to "off". To make this server-side change, navigate to the file located at

  `$ORACLE_HOME/Apache/Apache/conf/mod_osso.conf`

  and edit the line containing OssoIpCheck to say

  ```
  OssoIpCheck off
  ```

- After modifying the configuration file, restart the Oracle HTTP Server by executing the following stop and start commands:

  ```
  dcmctl updateConfig -v -d
  opmnctl stopproc process-type=HTTP_Server
  opmnctl startproc process-type=HTTP_Server
  opmnctl stopproc process-type=OC4J_SECURITY
  opmnctl startproc process-type=OC4J_SECURITY
  ```

"Network Error" message

Problem

This message can arise when a browser requires re-authentication because an operation was attempted with Oracle Application Server Certificate Authority after some period of inactivity.

Solution

You need to re-authenticate yourself to OracleAS Certificate Authority by going to the Certificate Management tab and, when asked, choosing the Web Admin Certificate.

OracleAS Certificate Authority Stops Working, or Network/Server Messages Appear

Problem

These symptoms can arise when a configuration change has altered the connection strings that OracleAS Certificate Authority uses to connect to its repository or to Oracle Internet Directory (for publishing certificates). Changes can include altered ports or Real Application Clusters (RAC) nodes, for example. The messages may say "Cannot Establish Connection" or "Internal Server Error".

Solution

Enable OracleAS Certificate Authority to re-acquire the new connection strings by issuing the following command:

`$ORACLE_HOME/oca/bin/ocactl updateconnection`

Command completion updates the configuration file at `$ORACLE_HOME/oca/conf/oca.conf`.

After using this command, you must restart OracleAS Certificate Authority by issuing the following commands:

```
$ORACLE_HOME/oca/bin/ocactl stop
$ORACLE_HOME/oca/bin/ocactl start
```

Certificate Issues

The following issues relate primarily to certificates or certificate management:

Installing user certificate does not install CA certificate on Netscape/Mozilla

Problem

An attempt to install a user certificate does not succeed.

Solution

- All CA/Sub CA certificates must contain the O (Organization) component in their Subject DN. The components mandatory in the CA/Sub CA DN are C, O, and CN each separated from the next by a comma.
- When installing Oracle Application Server Certificate Authority, or regenerating the Root CA, users should input a DN that includes at least country, organization, and common name ("C, O, CN").
- When installing a Sub CA, ensure that the DN of the CA signing certificate has O (organization) RDN in its subject DN.

Inability to Access or Use the Certificate Management Tab

Problem

Attempts to access or use the Certificate Management facility fail.

Solution

Before you can access Certificate Management, your browser must have imported a valid Web Administrator certificate. You must apply for and receive such a certificate before clicking Certificate Management. You do so in the Administration Setup tab, by clicking the button labeled Web Administrator Enrollment... .

Administrator Needs to Work from a Different Machine

Problem

An OracleAS Certificate Authority administrator may wish to do certificate management tasks from any of multiple machines. However, his Web Administrator certificate is contained in the browser of the machine he used when originally authenticating himself to be the OracleAS Certificate Authority Web Administrator.

Solution

To switch from one machine to another and maintain the ability to do certificate management tasks, you need to export the certificate from the previous browser and import it into the new browser, as follows:

- Exporting the certificate on Netscape/Mozilla: Choose Security->Certificates->Yours->choose the Web Admin Cert ->Export
- Importing the certificate on Netscape/Mozilla: Choose Security->Certificates->Yours->Import Certificate.
- Exporting the certificate on Internet Explorer: Choose Internet options ->Content->Certificates->Personal-><choose your Web Admin Cert> ->Export
- Importing the certificate on Internet Explorer: Choose Internet options->Content->Certificates->Personal->Import

Single Sign-on Issues

Some issues relate primarily to Single Sign-on capabilities:

Name shown on an SSO certificate appears only as "User"

Problem

These certificates do not show the common name or DN. They are distinguishable only by having different certificate serial numbers.

Solution

Click "View" to check the certificate serial number, and pick the certificate identified by the serial number you wish to use.

VBScript Error Message While Generating Keys

Problem

In Oracle Application Server Single Sign-On, you request a certificate by clicking "Submit" in the popup window. Since there is no message to wait and no visible indication of progress, users sometimes click "Submit" again, causing this error.

Solution

Try again, being sure to click "Submit" only once and to wait until the certificate is returned.

"Page can not be displayed" Message in Internet Explorer

Problem

After logging in to OracleAS Single Sign-On by name and password, but then changing authentication by choosing SSL, a known Internet Explorer bug gives the "Page cannot be displayed" error.

Solution

Try to reload the page. If that does not resolve the issue, exit from the current browser session, return to OracleAS Certificate Authority and try again.

Going to SSO login page in IE can get a security warning dialog

This is expected behavior; it is a warning that is issued due to switching from SSL protocol (https) to non-SSL protocol (http). No action is needed.

Certificate Acquired with Single Sign-on not Seen for SSL Authentication

Problem

After using the Mozilla browser to log in to OracleAS Single Sign-On, get a certificate, and import it, a user might still not see this just-imported certificate in the client authentication window.

Solution

- If the user did not include "Authentication" among the intended usages specified when requesting the certificate, then that certificate will not appear in the client authentication box for authentication use.

  To confirm the chosen usages, search for the certificate by its the serial number and see its details. If the Usages do not show Client Authentication, then this certificate cannot be used for SSL authentication.

  The solution is to request a new certificate, ensuring that Authentication is specified as one of the usages for the certificate.

- Another reason the certificate might not appear is that the CASSL certificate is unusable for some reason. In this case, the administrator must replace it.

See Also:

"Default Install Values for OracleAS Certificate Authority" in Chapter 4.
"Regenerating the CA SSL and CA S/MIME Wallets" in Chapter 7.

Backup Protection Issue

The following issue relates to making recovery possible after a failure.

Ensuring Recoverability of the OracleAS Certificate Authority Internal Repository

Problem

Errors and unpredictable events can threaten the continuity of OracleAS Certificate Authority operations.

Solution

Take a backup of the metadata repository periodically. For details, see *Oracle Application Server Administrator's Guide*, particularly the sections on Backup Strategies and Procedures and Recovery Strategies and Procedures.

Recovery Issue

This section describes how to recover from a major issue affecting OracleAS Certificate Authority operation:

- Clicking on the Certificate Management tab from the OracleAS Certificate Authority Administrative page returns a browser 404 error

Clicking on the Certificate Management tab from the OracleAS Certificate Authority Administrative page returns a browser 404 error

Problem

Under certain conditions, the OracleAS Certificate Authority Administrator may be unable to access the Certificate Management page. The browser reports a 404/Page Not Found error. Possible conditions for this error include, but are not limited to, the following:

- The administrator certificate is installed on one browser, but you try to access the Certificate Management page from a different browser.
- When applying for the CA certificate, the DN's specified the machine name only, and the domain information was omitted. For example, "CN=asunmach17 admin user,C=US" was specified instead of "CN=asunmach17.us.mycompany.com admin user,C=US".

Solution

If the problem is due to incorrect domain information in the CA certificate, you must re-create the CA's SSL wallet and refresh affected components using these steps:

> **WARNING: This is a last-resort workaround and is not to be used casually. Implement it only if you have exhausted other possibilities.**

1. Regenerate the new CA SSL wallet. Make sure the CN is the same as the host name; domain is optional.

   See "Regenerating the CA SSL and CA S/MIME Wallets" in Chapter 7 for details.
2. Restart OHS.
3. Once the CA certificate is regenerated, create the CASSL wallet. This operation is performed by the new CA.
4. Restart OHS to pick up the new CA SSL wallet.
5. Refresh the SSL session between the client's browser and the OracleAS Certificate Authority server.

General Issues

The following issues are general in nature and do not fall into the previous categories:

- Pages taking too long to load, or hanging
- No SMIME signing certificate in Outlook Express
- Browser warning about CA SSL Server's CN

Pages taking too long to load, or hanging

Problem

Sometimes such delays can occur, possibly after OracleAS Certificate Authority has been in operation for a substantial period.

Solution

Restart OracleAS Certificate Authority's OC4J instance, which will return you to faster operations.

> **See Also:** For additional performance tips, see "Performance Tuning for OracleAS Certificate Authority" in Chapter 7.
>
> For restart operations, see "Starting and Stopping Oracle Application Server Certificate Authority" in Chapter 4.

No SMIME signing certificate in Outlook Express

Problem

In some Windows environments, when you select the certificate for SMIME signing in Outlook Express, there is no certificate listed. The reason is that there is an installed version of Microsoft Outlook.

Solution

You will need to use Microsoft Outlook and not Outlook Express.

> **See Also:** Appendix G, "S/MIME with OracleAS Certificate Authority".

Browser warning about CA SSL Server's CN

Problem

This warning is raised if the CA SSL Server's CN is not identical to the machine name.

Solution

You will need to make the CN and machine name the same.

Need More Help?

You can find more solutions on Oracle *MetaLink*, `http://metalink.oracle.com`. If you do not find a solution for your problem, log a service request.

> **See Also:** *Oracle Application Server Release Notes*, available on the Oracle Technology Network: `http://www.oracle.com/technology/documentation/index.html`

D

Extensions

Oracle Application Server Certificate Authority is compliant with the X.509 V3 and IETF's PKIX standards, and supports standard extensions as described in this Appendix.

Certificate Usage

OracleAS Certificate Authority enables users to select the function of a requested certificate to fit their intended applications and their enterprise policies. The default as shipped is "Authentication, Encryption, and Signing," but the administrator can configure a different choice, which then becomes the preselected default for that site. Table D–1 shows the possible choices:

Table D–1 Types of Certificate Usage

Function	Description
Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.)
Encryption	Enables encrypting and decrypting electronic documents
Signing	Enables verifiable signature for (and assures non-tampering of) electronic documents, including email (using S/MIME, the Secure Multipurpose Internet Mail Extension)
Authentication, Encryption	Certificate can be used for both purposes.
Authentication, Signing	Certificate can be used for both purposes.
Authentication, Encryption, and Signing	Certificate can be used for all three purposes.
Encryption, Signing	Certificate can be used for both purposes.
CA Signing	Used to sign users' certificates or Certificate Revocation List (CRL).
Code Signing	Provides verifiable signature for the provider of (and assures non-tampering of) Java code, JavaScript, and other signed files.

Policy Application to Certificates

Certain policies apply to certificates intended for particular uses, as described in Table D–2.

Table D–2 Policies Applied for Particular Certificate Usages

Certificate Usage	Basic Constraints (Critical)	Key Usage (Non Critical)	Extended Key Usage (Non Critical)	Subject Alternate Name (Non Critical)
CA certificate	CA flag set to true PathLength: + root CA (generated during installation), value hardcoded to 3 root CA (generated using `OCACTL`), value can be chosen.	Signing Certificates (Keys) Signing CRLs		
Client Authentication		Digital Signature	clientAuth	rfc822Name=email AND/OR otherName=UID
Server Authentication		Digital Signature Key Encipherment	serverAuth	rfc822Name=email AND/OR otherName=UID
Signing		Digital Signature Non-Repudiation	emailProtection	rfc822Name=email AND/OR otherName=UID
Encryption		Data Encipherment Key Encipherment	emailProtection	
Code Signing		Digital Signature	codeSigning	rfc822Name=email AND/OR otherName=UID

E

Enabling SSL and PKI on SSO

The procedures in this Appendix are all the necessary and advisable steps for enabling SSL and PKI on Oracle Application Server Single Sign-On as of OracleAS 10*g* Release 2 (10.1.2). Detailed descriptions with additional context explanations appear in the following manuals:

- *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle HTTP Server Administrator's Guide*
- *Oracle Advanced Security Administrator's Guide*

By default, OracleAS Single Sign-On uses the HTTP port of the Oracle HTTP Server, and single sign-on authentication is based on user name and password. However, OracleAS Single Sign-On can be configured for SSL to authenticate a user based on the user's certificate. Although the configuration steps are already documented in OracleAS Single Sign-On and OHS documentation, they are scattered in many places. For convenience, these steps are combined in this Appendix.

Three separate steps are needed to configure this feature: enable SSL for OracleAS Single Sign-On server, configure OracleAS Single Sign-On to use certificates, and register OracleAS Certificate Authority with the SSL-enabled OracleAS Single Sign-On server.

> **Note:** This document applies to both UNIX and Windows platforms, except that for Windows, the path separator should be '\', instead of '/' and variables are dereferenced with '%' instead of '$'.

To achieve the objective of enabling SSL and PKI on OracleAS Single Sign-On, you must complete three sets of procedures:

- Enabling SSL on SSO
- Enabling PKI on SSO
- Re-registering the Virtual Host with the SSL-Enabled SSO

Enabling SSL on SSO

You can configure SSL for OracleAS Single Sign-On using either an automated or manual approach.

Automated SSL Configuration

For common topologies, the SSL Configuration Tool can perform the steps required to enable post-installation SSL of the Oracle HTTP Server. For details about the tool and

how to run it, see "Using the SSL Configuration Tool" in the *Oracle Application Server Administrator's Guide*.

Manual SSL Configuration

> **Note:** For detailed information, refer to the *Oracle Application Server Single Sign-On Administrator's Guide*, specially the chapter on Enabling SSL.

For this section, use the `ORACLE_HOME` location where the OracleAS Single Sign-On server is installed.

1. Edit the `$ORACLE_HOME/opmn/conf/opmn.xml file`:
2. Search for `id="HTTP_Server"`, and then, four lines down, change the following line:

   ```
   <data id="start-mode value="ssl-disabled">
   ```

   to read instead as follows:

   ```
   <data id="start-mode value="ssl-enabled">
   ```

3. Restart opmn using the new xml file:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl reload
   ```

4. Edit the `$ORACLE_HOME/Apache/Apache/conf/ssl.conf` file:
5. On the line before `</VirtualHost>`, add the following:

   ```
   RewriteEngine on
   RewriteOptions inherit
   ```

6. Disable the SSL session cache to force SSL to perform a handshake when logging out of OracleAS Single Sign-On, as follows:

   Comment out the SSLSessionCache and SSLSessionCacheTimeout directives in ssl.conf.sec:

   ```
   # SSLSessionCache
   # SSLSessionCacheTimeout 15
   ```

   Then add the following line:

   ```
   SSLSessionCache none
   ```

7. Update the wallet. If OracleAS Certificate Authority was installed in the same machine, you can use its SSL wallet for the OracleAS Single Sign-On server.

   If not, you need to use Oracle Wallet Manager to generate a wallet for the OracleAS Single Sign-On server: see its documentation in the *Oracle Advanced Security Administrator's Guide*.

   Typically an existing SSL wallet generated by OracleAS Certificate Authority is located in `/app/oracle/oca/wallet/ssl`. Locate the SSLWallet directive in this file (`ssl.conf`) and comment it out:

   ```
   # SSLWallet file:/app/oracle/product/sec_inf/Apache/Apache/conf/ssl.wlt/default
   ```

   and insert a new one that reads as follows:

```
SSLWallet file:/app/oracle/oca/wallet/ssl
```

8. Set client authentication by commenting out the following line:

```
# SSLVerifyClient require
```

and inserting a new one that reads as follows:

```
SSLVerifyClient optional
```

9. Reconfigure the OracleAS Single Sign-On server to use the SSL port. The command form is:

```
$ORACLE_HOME/sso/bin/ssocfg.sh https  hostname  ohs_ssl_port
```

So if the hostname is sso.us.oracle.com and ohs_ssl_port is 4443, then the command becomes the following line:

```
$ORACLE_HOME/sso/bin/ssocfg.sh https  sso.us.oracle.com 4443
```

10. Register mod_osso for sso by running the following command in the Oracle home where OracleAS Single Sign-On was installed (UNIX):

```
$ORACLE_HOME/sso/bin/ssoreg.sh \
-oracle_home_path $ORACLE_HOME   -site_name sso   -config_mod_osso TRUE \
-mod_osso_url  https://hostname.domain.com:ohs_ssl_port \
-update_mode CREATE   -u root
```

Note: For Windows, the command is:

```
%ORACLE_HOME%\sso\bin\ssoreg.bat
-oracle_home_path orcl_home_path
-site_name site_name
-config_mod_osso TRUE
-mod_osso_url mod_osso_url
-u userid
-virtualhost
-update_mode CREATE
```

11. Restart OHS for OracleAS Single Sign-On by running the following command:

```
$ORACLE_HOME/opmn/bin/opmnctl restartproc type=ohs
```

Enabling PKI on SSO

For this section, the Oracle home to use is the location where the OracleAS Single Sign-On server is installed.

The following steps enable PKI on OracleAS Single Sign-On:

1. Edit `$ORACLE_HOME/sso/conf/policy.properties` to set the default authentication level to High and to set the correct corresponding plugin, as follows:

```
DefaultAuthLevel = MediumHighSecurity

MediumHighSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOX509CertAuth
```

2. Configure OracleAS Certificate Authority to use username and password for provisioning, using lines of the following form:

```
MediumSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOServerAuth
Oca_hostname\:port = MediumSecurity
```

   For example, if the Oca_hostname is oca.us.oracle.com and the OracleAS Certificate Authority port is 6600, then this option is written as follows:

```
oca.us.oracle.com\:6600=MediumSecurity
```

3. With these options all set, a user logging in to any partner application is required to have a certificate, except for OracleAS Certificate Authority, where he can get a certificate.

   Restart the OracleAS Single Sign-On server using the following commands:

```
$ORACLE_HOME/opmn/bin/opmnctl stopproc  process-type=OC4J_SECURITY
$ORACLE_HOME/opmn/bin/opmnctl startproc  process-type=OC4J_SECURITY
```

Re-registering the Virtual Host with the SSL-Enabled SSO

For this section, the *ORACLE_HOME* to use is the location where OracleAS Certificate Authority is installed.

Each time the administrator enables the OracleAS Single Sign-On server to use SSL, the OracleAS Certificate Authority virtual host must be re-registered with the SSL-enabled OracleAS Single Sign-On server. All OracleAS Single Sign-On-using applications must do so. Re-registration is done by using the single sign-on registration tool, `ossoreg.jar`. OracleAS Certificate Authority's use of this tool is explained here; its general use for all Single Sign-On enabled applications is explained in *Oracle Application Server Single Sign-On Administrator's Guide*.

1. Re-register mod_osso for OracleAS Certificate Authority by running the following command:

```
$ORACLE_HOME/sso/bin/ssoreg.sh \
-oracle_home_path $ORACLE_HOME -site_name OracleAS Certificate Authority \
-config_mod_osso TRUE \
-mod_osso_url https://hostname.domain.com:oca_ssl_port  -u root \
-virtualhost \
-config_file $ORACLE_HOME/Apache/Apache/conf/osso/oca/osso.conf
```

   Running this tool on the machine hosting the OracleAS Single Sign-On server generates OracleAS Certificate Authority's mod_osso record in the osso.conf file, reflecting SSL settings on the single sign-on server.

2. Restart OHS for OracleAS Certificate Authority by running the following command:

```
$ORACLE_HOME/opmn/bin/opmnctl restartproc type=ohs
```

Example of Re-Registration

Suppose that the OracleAS Certificate Authority host name is `myoca.mysite.com` and the `OracleAS Certificate Authority` server authentication port is 6600. The following steps accomplish the re-registration:

1. Use these two commands to set the variables to be used by the actual command (in step 2):

   On csh and tcsh:

```
setenv ORACLE_HOME /sso_server/oracle_home
setenv LD_LIBRARY_PATH $ORACLE_HOME/lib
```

On Bourne and ksh shells:

```
set ORACLE_HOME=/sso_server/oracle_home; export ORACLE_HOME
set LD_LIBRARY_PATH=$ORACLE_HOME/lib; export LD_LIBRARY_PATH
```

2. Using these variables as set, the actual command on a UNIX system would be as follows (although on a single line):

```
$ORACLE_HOME/sso/bin/ssoreg.sh \
-oracle_home_path $ORACLE_HOME -site_name my_oca_site_name \
-config_mod_osso TRUE  -mod_osso_url https://myoca.mysite.com:6600 \
-u root -config_file $ORACLE_HOME/Apache/Apache/conf/osso/oca/osso.conf \
-virtualhost
```

For Windows, the commands are:

```
set ORACLE_HOME=c:\sso_server\oracle_home

%ORACLE_HOME%\sso\bin\ssoreg.bat
-oracle_home_path $ORACLE_HOME
-site_name "my_oca_site_name"
-config_mod_osso TRUE
-mod_osso_url https://myoca.mysite.com:6600
-u SYSTEM
-config_file $ORACLE_HOME\Apache\Apache\conf\osso\oca\osso.conf
-virtualhost
```

F

External Access to Protected OracleAS Certificate Authority

Secure processes protected behind a firewall, like OracleAS Certificate Authority, can still serve customers outside the firewall by using a proxy server.

This intermediary server securely intercepts all user requests for certificate services and forwards them to OracleAS Certificate Authority. The proxy server uses only two ports: port 443 (for SSL communications) and port 80 (for non-SSL communications).

Since OracleAS Certificate Authority has two virtual hosts, one for server authentication and one for mutual authentication, two proxy servers are required, as illustrated by the following example:

Example F–1 Proxy Server Example

A proxy server for server authentication could use this URL:

`https://myproxy_server1.acme.com` (with default SSL port 443)

which maps to

`https://myoca.acme.com:6600` (server authentication)

A second proxy server, for mutual authentication, could use this URL:

https://myproxy_server2.acme.com (with default SSL port 443)

which maps to

https://myoca.acme.com:6601 (mutual authentication)

This Appendix explains how you enable OracleAS Certificate Authority to support proxy servers and how to map a proxy server to an OracleAS Certificate Authority virtual host.

Enabling OracleAS Certificate Authority to Support Proxy Servers

The following steps enable OracleAS Certificate Authority to support proxy servers:

1. Log on to the database as an OracleAS Certificate Authority user.
2. Run script `$ORACLE_HOME/oca/sql/ocabigipon.sql`.
3. Enter the proxy server's hostname and SSL port that maps to the OracleAS Certificate Authority mutual authentication port (in Proxy Server Example, it's myproxy_server2.acme.com and port 443)
4. Map the proxy server to the OracleAS Certificate Authority virtual host.

Disabling OracleAS Certificate Authority's Support for Proxy Servers

The following steps disable OracleAS Certificate Authority's support for proxy servers:

1. Log on to the database as OracleAS Certificate Authority user.
2. Run script `$ORACLE_HOME/oca/sql/ocabigipoff.sql`.

G

S/MIME with OracleAS Certificate Authority

S/MIME applications, such as Outlook, Mozilla, or Netscape mail clients, can sign and encrypt mail messages based on PKI.

SMIME Operations

A sender can sign a message by using his signing private key and the recipient can verify the signature by using the sender's signing certificate (usually sent along with the signed mail message).

A sender can encrypt a message by using the recipient's encryption certificate and recipient can decrypt the message by using his encryption private key.

Users can request and get signing, or encryption, or both types of certificates from OracleAS Certificate Authority using browser (Internet Explorer, Mozilla or Netscape).

Setup

Setting up S/MIME operations involves getting certificates and establishing the SMIME parameters.

Getting certificates

To get the S/MIME certificates from OracleAS Certificate Authority, please refer to"User Certificates Tab" on page 8-4 in Chapter 8, "End-User Interface of the Oracle Application Server Certificate Authority". Be sure to install the certificate into the browser.

A user can get a single certificates that does both signing and encryption, but security is better when each user gets two certificates: one for signing and one for encryption. The signing key should be kept securely on the user's machine or smart card for non-repudiation purpose. The encryption key should be archived for recovering the encrypted message if the encryption key is lost.

Setting S/MIME parameters

You can set your S/MIME parameters in an Outlook Mail client or a Mozilla/Netscape Mail client.

Outlook Mail Client In Outlook, select **Tools-->Options-->Security-->Setup Secure Email**:

- In **Security Settings Name**, put the name you want.
- In **Certificates and Algorithms:**

  - Choose your signing certificate. You will sign outgoing message by using this certificate.
  - Choose your encryption certificate. People will encrypt messages sent to you by using this certificate.
- Check the box **Send these certificates with signed messages**.
- Click **OK** repeatedly to finish this setting process.

Mozilla/Netscape Mail Client In a Mozilla/Netscape Mail client, select **Edit-->Mail & Newsgroups Account Settings-->Security:**

- In the **Digital Signing** pane, click **Select** to choose the *signing* certificate you created for that purpose.
- In the **Encryption** pane, click **Select** to choose the *encryption* certificate you created for that purpose. (The same certificate can server both purposes if the **Usage** you selected included both Encryption and Signing.)

OCA Configuration Notifications are sent by OCA to the administrator and users. These notifications can be encrypted using SMIME: see the "Notification Sub-tab" section in Chapter 5, "Configuring Oracle Application Server Certificate Authority".

Sending Messages

After composing the mail message, do the following steps before sending the message:

Outlook Mail Client

In Outlook, take these steps to encrypt your message, or sign it, or both:

- To encrypt the message, go to **Options** and check the box **Encrypt message contents and attachments**. Make sure that you have the encryption certificate for each and every recipient. (To get a recipient's encryption certificate, see "Getting Other People's Encryption Certificates".)
- To sign the message, go to **Options** and check the box **Add digital signature to outgoing message**.

Mozilla/Netscape Mail Client

In a Mozilla/Netscape Mail client:

- To encrypt the message, click **Security**-->**Encrypt This Message**.
- To sign the message, click **Security**-->**Digital Sign This Message**.

Receiving Messages

You can read an encrypted message if you have the private key of the certificate used to encrypt the message. This key also enables you to verify the signature of the sender, if you trust the CA that signs the sender's signing certificate. To view the security information of the message, click the message, and then do the following steps corresponding to your particular mail client:

Outlook Mail Client

Go to **File**-->**Properties**-->**Security**.

Mozilla/Netscape Mail Client

Go to **View-->Message Security Info**.

Getting Other People's Encryption Certificates

You can encrypt messages you intend to send to a particular recipient by using that recipient's certificate. You can acquire those certificates as follows:

- If you receive a message that includes the sender's encryption certificate, then that certificate will automatically be saved in your certificate store.
- You can ask people to send you their encryption certificates (with no private key), which you can then save in your certificate store.
- You could also retrieve encryption certificates from an LDAP directory:
  - In Outlook, the following circumstances cause automatic retrieval of another user's certificate from the LDAP directory.
    * If you are using Internet Only mode with a standard LDAP server, sending an encrypted e-mail message to a user in that LDAP server causes retrieval of his certificate. For this to work, you must be enrolled in S/MIME security and you must have a Digital ID for your e-mail account.
    * When you use Corporate/Workgroup mode with Microsoft Exchange Server, you can obtain certificates from the Global Address Book. You must be enrolled in Exchange Advanced Security.
    * In Mozilla, getting certificates *automatically* from LDAP is not yet supported.

      Using LDAP commands directly, you could retrieve each desired certificate from the directory, store it in a file, and finally save it into your certificate store.

H

Configuring OracleAS WebCache for OracleAS Certificate Authority

Oracle offers OracleAS Web Cache to help e-businesses manage Web site and Web-based application performance issues. OracleAS Web Cache is a content-aware server accelerator, or reverse proxy server, that improves the performance, scalability, and availability of Web sites that run on Oracle Application Server.

This Appendix explains how you can deploy OracleAS Web Cache to work with Oracle Application Server Certificate Authority. It provides key instructions and provides references for additional reading that you may find useful during configuration.

Perform the installation in the following stages:

- Install OracleAS WebCache
- Configure OracleAS WebCache for OracleAS Certificate Authority
- Configure OracleAS Certificate Authority Virtual Hosts for OracleAS WebCache
- Enable OracleAS WebCache for OracleAS Certificate Authority

Install OracleAS WebCache

Install OracleAS Web Cache by installing an instance of Oracle Application Server with the "J2EE and Webcache" component option. Although in practice you can install this instance on the same machine where OracleAS Certificate Authority resides, for testing purposes it is preferable to install OracleAS Web Cache on a different machine with a different hostname.

For more information, see the following:

- To download free release notes, installation documentation, white papers, or other collateral for OracleAS Web Cache, please visit the Oracle Technology Network (OTN) at `http://www.oracle.com/technology/index.html`.
- For configuration details, see the *Oracle Application Server Web Cache Administrator's Guide*, specifically "Part II, Configuration and Administration of OracleAS Web Cache".

Configure OracleAS WebCache for OracleAS Certificate Authority

Configure OracleAS Web Cache for OracleAS Certificate Authority and OracleAS Single Sign-On. Use these steps:

1. Obtain an SSL server wallet for the machine on which OracleAS Web Cache resides. Use Oracle Wallet Manager for this task.

   Note: The CN is the Web Cache host name.

   See the *Oracle HTTP Server Administrator's Guide* for details.

2. Use Oracle Enterprise Manager 10*g* to configure OracleAS Web Cache. From the Application Server Admin Control:
   - Go to **webcache** -> **Administration**
   - Under **Webcache** -> **Ports**, create Web Cache listener ports. There should be one port for each server, namely Web Cache listener port 4600 for OracleAS Certificate Authority port 6600 (server auth), Web Cache listener port 4601 for OracleAS Certificate Authority port 6601 (mutual auth), and Web Cache listener port 7778 for SSO port 7777 (non-SSL).

     For each port you configure, make sure to check HTTPS, specify Web Cache SSL wallet for SSL server (for example, the OracleAS Certificate Authority server), and client certificate for mutual authentication port, if required. For example, port 4601 should be HTTPS and requires a client certificate.
   - Under **Application** -> **Origin Servers**, create origin servers.

     The origin server is the description of the web server (host, port and protocol). There are two origin servers for OracleAS Certificate Authority, the first for https://*hostname*:6600 and the second for https://*hostname*:6601.

     There is also an origin server for OracleAS Single Sign-On.
   - Under **Application** -> **Sites**, create sites. A site contains the Web Cache hostname, above Listener port and protocol. Click on **Advanced** to choose HTTPS and Required Client Cert if necessary.

     The site is also mapped to the origin server.
   - Set the SSL wallet for the Web Cache. To accomplish this, go to **Webcache** -> **Security** and enter the SSL wallet location.

3. Restart OracleAS Web Cache.

   Note: The Web Cache restart may fail if it runs out of file descriptors. To resolve this problem, see the *Oracle Application Server Web Cache Administrator's Guide*.

For additional OracleAS Web Cache configuration details, see *Oracle Application Server Web Cache Administrator's Guide*

For information about configuring OracleAS Single Sign-On for OracleAS Web Cache, see the *Oracle Application Server Single Sign-On Administrator's Guide*, "Deploying OracleAS Single Sign-On with a Proxy Server".

Configure OracleAS Certificate Authority Virtual Hosts for OracleAS WebCache

Follow these steps to configure OracleAS Certificate Authority virtual hosts for OracleAS WebCache host and port:

1. Edit the `ocm_apache.conf` file, in the server auth virtual host section, as follows:
    1. Change the `ServerName` to Web Cache hostname (instead of the actual OracleAS Certificate Authority host name)
    2. Add a Port directive with Web Cache port for this virtual host. (for example, Port 4600).
    3. Add the following lines:

```
LoadModule certheaders_module libexec/mod_certheaders.so
AddCertHeader HTTPS
AddCertHeader SSL_CLIENT_CERT
```

    4. Comment out the following line:

```
SSLOptions +FakeBasicAuth +ExportCertData +CompatEnvVars +StrictRequire
```

2. Execute the command:

```
dcmctl updateconfig -ct ohs
```

3. Restart Oracle HTTP Server:

```
opmnctl restartproc type=ohs
```

4. On Internet Explorer, you may encounter a bug which you can work around by modifying the `$ORACLE_HOME/webcache/internal.xml` file. Insert IEHOSTHEADERBUG=*SSO_WEBC_PORT* in the <MISCELLANEOUS/> tag, where *SSO_WEBC_PORT* is the Web Cache port mapped to the SSO port.
5. Restart OracleAS WebCache.

> **Note:** The Web Cache restart may fail if it runs out of file descriptors. To resolve this problem, see the *Oracle Application Server Web Cache Administrator's Guide*.

Enable OracleAS WebCache for OracleAS Certificate Authority

Enable OracleAS Web Cache for OracleAS Certificate Authority by executing the following command:

```
$ORACLE_HOME/bin/sqlplus oca/ocadbpass
    @$ORACLE_HOME/oca/sql/ocabigipon.sql
```

If you wish to change the host and port of Web Cache sites for OracleAS Certificate Authority, execute this command:

```
$ORACLE_HOME/oca/sql/ocabigipoff.sql
```

followed by:

```
$ORACLE_HOME/oca/sql/ocabigipon.sql
```

If you need to disable OracleAS Web Cache for OracleAS Certificate Authority, execute this command:

```
$ORACLE_HOME/bin/sqlplus oca/ocadbpass
    @$ORACLE_HOME/oca/sql/ocabigipoff.sql
```

The Oracle Application Server Certificate Authority Web Interface

This appendix lists and describes the various windows, fields, and control devices in the Oracle Application Server Certificate Authority Web interface. It contains these sections:

- Windows and Fields in the Administration Interface
- Windows and Fields in the End-User Interface

Windows and Fields in the Administration Interface

This section lists and describes the windows and fields in the Web administrative interface.

Web Administrator Enrollment--Advanced DN

Use this page to enroll as a Web administrator if you know your full distinguished name (DN)--if it already exists--and understand how to enter it in LDIF format. This feature is a shortcut for the Distinguished Name Information heading on the Web Administrator Enrollment page, where it appears as the link Advanced DN. The DN is the location of your user entry in Oracle Internet Directory. Oracle Application Server Certificate Authority stores your certificate in and retrieves it from your directory entry.

For enrollment instructions, see Requesting the Administrator Certificate.

Advanced Screen

Use the Advanced screen to narrow or refine your search for certificate requests or existing certificates. The Advanced screen offers the following five search methods:

1. Search Certificate Requests Using Request Status

   Choose from three certificate search categories: Pending, Rejected, and Certified. You must use this option to search for rejected certificates

2. Search Using Distinguished Name (DN)

   Choose Certificate or Certificate Request from the Search list to specify certificate holders or certificate requesters. Enter the components of the certificate holder's or certificate requester's distinguished name. All fields are optional.

3. Search Using Advanced Distinguished Name

Choose this option if you know the certificate requester or certificate holder's full DN and understand how to enter it in LDIF format. In the following example, note that spaces between attributes are optional:

cn=Margarita Redmond,ou=sales,o=yourcorp,l=Bismarck,st=SD,c=US

4. Search Using Serial Number/Request ID Range

   Choose this option to retrieve information about certificates that fall within a given range. Use the Certificate list to toggle between requested certificates and existing certificates. Both serial number fields are mandatory.

5. Search Certificates Using Certificate Status

   Choose from three certificate status categories: Valid, Revoked, or Expired.

To learn how to search using these methods, please see the Advanced Search Screen.

Certificate Details

Use this page to obtain a complete description of a certificate, including its BASE64 encoding. The non modifiable fields on this page are as follows:

Field	Description
Status	This value is `VALID`, `REVOKED`, or `EXPIRED`
Serial Number	The certificate's serial number assigned by Oracle Application Server Certificate Authority.
Signature Algorithm	The algorithm used, such as MD5 with RSA encryption, which is indicated by the object identification number (OID).
Usage	The certificate function.
Issuing Authority	The CA that issued the certificate.
Subject DN	Distinguished name of the certificate holder.
Not Valid Before	Date and time certificate became valid.
Not Valid After	Date and time certificate expires.
BASE64-Encoded Certificate with CA certificate chain in PKCS#7 format	The encoded certificate plus its tree of trusted authorities, in PKCS#7 format. This form allows a single operation to transport all certificates in the trusted chain up through the root CA certificate.
BASE64 Encoded Certificate	The encoded certificate.

Choose one of the buttons located at the bottom of the page to perform your desired task:

Button Name	Description
OK	Returns you to the main certificate management page.
Revoke	Revokes the certificate. You must specify a revocation reason.
Renew	Renews the certificate. You must specify a new validity period.
Install in Browser	Installs the certificate into your browser.

Certificate Request Rejection

Use this page to reject a manual certificate request. You reject the request by choosing Submit. Choosing Cancel returns you to the Requests page.

The page contains nonmodifiable fields that constitute a profile of the certificate requester. A description of these fields follows:

Field	Description
Status	This value is always `PENDING`.
Certificate Type	This value is `client`, `server`, or `ca` (certificate authority)
Certificate Usage	The certificate function (SSL Client, Signing, or other)
Serial Number	The serial number used to reference the certificate request. OracleAS Certificate Authority assigns a new value when you approve a certificate request.
Subject DN	The distinguished name (DN) of the requester. The DN is the location of the requester's user entry in Oracle Internet Directory.
Request Date	The date and time that the user entered the request on the manual request form.
Algorithm	The algorithm used to encrypt the certificate.
Exponent	The public key exponent. The larger this number is, the longer clients take to encrypt messages.

Please see Certificate Request Rejection to learn how to perform this task.

Certificate Request Approval - Manual

Use this page to approve or reject a certificate request (by clicking either Approve or Reject). Choosing Cancel returns you to the Requests page.

This page displays details of the certificate request and lets you enable or edit the following features and fields:

- **Apply policy check while approving a certificate request**

  If policy checking is disabled (unchecked), then policy rules are not applied to the certificate request. This is useful when issuing special certificates that do not conform to policy rules.

- **Subject (Requester)**

  Administrators can edit the DN if users have entered it incorrectly.

- **Validity**

  Administrators can change the validity period before approving certificate requests.

The read-only Certificate Request Information fields are described as follows:

Field	Description
Status	This value is always `PENDING`.
Certificate Type	This value is either `client` or `server`.
Certificate Usage	This is one of eight values: Authentication; Encryption; Signing; Authentication and Encryption; Authentication and Signing; Encryption and Signing; Authentication, Encryption, and Signing; or Code Signing.
Serial Number	The serial number used to reference the certificate request when it is pending or once it has been granted. Oracle Application Server Certificate Authority assigns a new value when you approve a certificate request.
Subject DN	The distinguished name (DN) of the requester. The DN is the location of the requester's user entry in Oracle Internet Directory.
Request Date	The date and time that the user entered the request on the manual request form.
Algorithm	The algorithm used to encrypt the certificate and the exponent.
Exponent	The public key exponent. The larger this number is, the longer clients take to encrypt messages.

Please see Approving or Rejecting Certificate Requests to learn how to perform this task.

Requests Page

The Requests page displays a table listing all pending certificate requests that require administrator attention.

Use the buttons on this page to perform the following tasks:

- Search for and List Certificates and Certificate Requests
- Update the Certificate Revocation List

To perform one of the following tasks, select a request and click View Details:

- Approve Certificate Requests
- Reject Certificate Requests
- Revoke Certificates

Click a link to learn about a task.

Adding Custom Policies

The default policy plug-ins shipped with Oracle Application Server Certificate Authority are generic. You may need to enhance the default policy framework to suit your organization by writing custom policy plug-ins. Application programming interfaces (APIs) are provided to get information about certificate requests, certificates,

and other generic functions. Adding a policy is also referred to as registering a policy with Oracle Application Server Certificate Authority.

To add a custom policy:

1. Write a Java class that implements the OCACustomPolicyPlugin interface.
   - See the oracle.security.oca.policy package in the Javadoc provided with the documentation for descriptions of the classes and methods provided in OCACustomPolicyPlugin.
   - See Developing a Custom Policy Plug-in for information about writing a custom policy Java class
2. Package your custom policy Java class into a `.jar` file and place it in the following location, depending on your platform:
   - `$ORACLE_HOME/oca/policy` (UNIX)
   - `ORACLE_BASE\ORACLE_HOME\oca\policy` (Windows)

   If the policy subdirectory does not exist, then create it.
3. To register your custom policy with Oracle Application Server Certificate Authority, log in to the Web administrative interface.
4. On the main Policy page of the Configuration Management tab, select the **Operation** type for the custom policy you want to add and click **Go**. The Policy Rules page for that Operation appears.
5. On the Policy Rules page for the Operation type you selected, click **Add**, which is located on the right most side of the page. The Custom Policy Details page appears.
6. On the Custom Policy Details page, enter the information for your custom policy into the provided fields. The following describes the type of information each field requires:
   - **Name**: The name of your customer policy. For example, AuditCertDetails.
   - **Description**: A description of what your custom policy does.
   - **Class**: The name of the Java class that implements your custom policy. See Steps 1 and 2.
7. Check **Enable this policy** to activate the custom policy and click **OK**. A message appears confirming that a new policy has been added.
8. Check that the policy precedence is what you want for this policy. See Policy Actions for details on reordering policy precedence.
9. Restart the Oracle Application Server Certificate Authority server for your custom policy to take effect. See Starting and Stopping Oracle Application Server Certificate Authority.

Related Topics

See Policy Actions for the following topics:

- Viewing Policies
- Editing Policies
- Enabling Policies
- Disabling Policies

- Deleting Policies
- Reordering Policy Precedence
- Policy Management

Edit RenewalRequestConstraint

Use this page to set the default values and restrictions for the RenewalRequestConstraint policy. This policy applies to client or server/sub ca certificate renewal requests and restricts whether expired certificates can be renewed. It can be applied to SSL users.

You can modify the following default values and restrictions for this policy:

Parameter Details

Parameter	Default Value	Description
Allow Renewal	Checked (TRUE)	Specifies whether certificates can be renewed. When it is checked (TRUE), certificates can be renewed.
Days before expiration date	15 days	Specifies how many days before a certificate's expiration it can be renewed. If you specify 0 (zero), then certificates cannot be renewed before their expiration date. If you specify * (asterisk), then certificates can be renewed anytime before their expiration date.
Days after expiration date	15 days	Specifies how many days after a certificate's expiration it can be renewed. If you specify 0 (zero), then certificates cannot be renewed after their expiration date. If you specify * (asterisk), then certificates can be renewed anytime after their expiration date.
Duration of renewal (days)	365 days	Specifies how long renewed certificates are valid.

Predicate Details

Predicate expressions are optional, altering application of this policy when an incoming request matches a predicate expression specified here. A policy with no predicates applies to all incoming requests. For example, the following predicate expression specifies that all client renewal requests coming from the Acme Company marketing department that is located in Japan (ou=marketing,o=acme,c=japan) are subject to the parameter settings chosen for this predicate:

Type=="client" AND DN=="ou=marketing,o=acme,c=japan"

For detailed information about predicate expression syntax, see "Predicates in Policy Rules".

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters and predicates to their previous values.

Button	Description
Apply	Click this button to apply any changes made to this page.
Cancel	Click this button to cancel any changes made to this page and return to the main Policy page.
OK	Click this button to return to the main Policy page. Any changes made to this page are automatically saved.

Related Topics

See Predicates in Policy Rules for the following topics:

- Editing Policies
- Adding Predicates to Policy Rules
- Reordering Predicate Precedence in Policy Rules
- Deleting Predicates from Policy Rules

Edit RevocationConstraintRule

Use this page to specify whether expired certificates can be revoked. Note: When enabled, this policy applies to all certificate revocation requests from both clients and servers. To provide different limits for certificate revocation requests from particular DNs, use predicates as described in the following section.

You can modify the following parameters and restrictions for this policy:

Parameter Details

allow revocation of expired certificates

Check this parameter to turn it on. When checked, expired certificates can be revoked, and when unchecked they cannot. By default this parameter is checked.

Predicate Details

Predicate expressions are optional, altering application of this policy when an incoming request matches a predicate expression. A policy with no predicates applies to all requests. For example, the following predicate expression specifies that client certificates from the United Kingdom Acme Company sales department (ou=sales,o=acme,c=uk) are subject to the parameter setting chosen for this predicate:

Type=="client" AND DN=="ou=sales,o=acme,c=uk"

For detailed information about predicate expression syntax, see "Predicates in Policy Rules".

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters and predicates to their previous values.

Button	Description
Apply	Click this button to apply any changes made to this page.
Cancel	Click this button to cancel any changes made to this page and return to the main Policy page.
OK	Click this button to return to the main Policy page. Any changes made to this page are automatically saved.

Related Topics

See Predicates in Policy Rules for the following topics:

- Editing Policies
- Adding Predicates to Policy Rules
- Reordering Predicate Precedence in Policy Rules
- Deleting Predicates from Policy Rules

Edit RSAKeyConstraints

Use this page to set the default values and restrictions for the RSAKeyConstraints policy. This policy specifies the minimum and maximum values for the length, in bits, of a public or private key. The drop-down lists show the selectable choices for certificate requests that do not meet any specified predicates. The corresponding limits for certificate requests that do meet a specified predicate are shown on the same line as that predicate.

> **Note:** When enabled, the default values for this policy apply to all certificate requests from both clients and servers. To provide different limits for certificate requests from particular **DN**s or for particular certificate types or usages, use predicates as described in the following section.

You can modify the following default values and restrictions for this policy:

Parameter Details

Parameter	Default Value	Description
maxsize Default Value	2048	Maximum key length
minsize Default Value	1024	Minimum key length. Use this parameter to ensure a minimum level of security.

Predicate Details

Predicate expressions are optional, altering application of this policy when an incoming request matches a predicate expression. A policy with no predicates applies to all requests. For example, the following predicate expression requires that client SSL certificate requests use the key lengths specified with this predicate:

OCMCert.Type=="client" AND OCMCert.Usage=="ssl"

For detailed information about predicate expression syntax, see "Predicates in Policy Rules".

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters and predicates to their previous values.
Apply	Click this button to apply any changes made to this page.
Cancel	Click this button to cancel any changes made to this page and return to the main Policy page.
OK	Click this button to return to the main Policy page. Any changes made to this page are automatically saved.

Related Topics

See Predicates in Policy Rules for the following topics:

- Editing Policies
- Adding Predicates to Policy Rules
- Reordering Predicate Precedence in Policy Rules
- Deleting Predicates from Policy Rules

Edit TrustPointDNCustomRule

Use this page to enable or disable the TrustPointDNCustomRule policy, an example of a custom plug-in policy you can develop by using the application programming interfaces (APIs) that OracleAS Certificate Authority provides. See "Developing a Custom Policy Plug-in" for more information.

When enabled, the TrustPointDNCustomRule policy checks the DN in every certificate request against all of the CA and subCA certificates' DNs in the certificate chain. If the DN specified in the certificate request matches any CA's DN, then OracleAS Certificate Authority rejects the request. (The certificate chain is an ordered list of certificates containing an end entity certificate and its corresponding CA certificates.)

Related Topic

- Adding Custom Policies

Edit UniqueCertificateConstraints

Use this page to enable and set the default values and restrictions for the UniqueCertificateConstraints policy, which limits each user to a single certificate for each specific usage or allows a user to have multiple certificates for each usage. If enabled, this policy verifies whether there are certificates in the repository that match the subject DN of the incoming certificate request. If a certificate with a matching DN is found and **Allow multiple certificates** is unchecked (FALSE), then the server also verifies whether certificate usage is the same. If a certificate with the same usage is

found, then OracleAS Certificate Authority will not issue another certificate with the same usage to the same subject DN.

You can modify the following parameters and restrictions for this policy:

Parameter Details

Allow multiple certificates Default Value

If this parameter is checked (TRUE), then OracleAS Certificate Authority will issue a certificate although there may be multiple certificates with the same subject DN and the same usage. If this parameter is left unchecked (FALSE), then the server will not issue multiple certificates with the same usage to the same subject DN.

Predicate Details

Predicate expressions are optional, altering application of this policy when an incoming request matches a predicate expression. A policy with no predicates applies to all requests. For example, the following predicate expression specifies that client certificate requests from the Acme Company's accounts payable department, located in Trenton, New Jersey, USA (ou=acct_pay,loc=trenton,o=acme,c=us) can get multiple certificates for the same DN and usage:

Type=="client" AND DN=="ou=acct_pay,loc=trenton,o=acme,c=us"

Allow multiple certificates value set to TRUE.

For detailed information about predicate expression syntax, see "Predicates in Policy Rules".

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters and predicates to their previous values.
Apply	Click this button to apply any changes made to this page.
Cancel	Click this button to cancel any changes made to this page and return to the main Policy page.
OK	Click this button to return to the main Policy page. Any changes made to this page are automatically saved.

Related Topics

- Edit (Editing Policies)
- Adding Predicates
- Reordering Predicates
- Delete (Deleting Predicates)

Edit ValidityRule

Use this page to set the default values and restrictions for the ValidityRule policy, which enforces a maximum and minimum period of time that manually authenticated requests can specify for certificate validity. For example, if the default maximum

validity period is set to 1825 days (5 years) and a certificate request asks for a 3650 day (10 year) validity period, then this policy will reject this request. All values must be specified in days for this parameter.

For SSL or OracleAS Single Sign-On (SSO) authenticated users, you can set the Default Validity period parameter, which automatically populates for those type of requests.

Note: When enabled, this policy applies to all certificate requests from both clients and servers. To provide different limits for certificate requests from particularDNs use predicates as described in the following section.

You can modify the following parameters and restrictions for this policy:

Parameter Details

Parameter	Default Value	Description
Maximum Validity period	3650	The maximum period in days that certificates are valid.
Minimum Validity period	90	The minimum period in days that certificates are valid.
Default Validity period	365	The validity period in days for SSO or SSL authenticated certificate requests.

Predicate Details

Predicate expressions are optional, altering application of this policy when an incoming request matches a predicate expression. A policy with no predicates applies to all requests. For example, the following predicate expression specifies that client SSL certificate requests use the maximum and minimum validity periods selected with this predicate:

Type=="client" AND Usage=="ssl"

For detailed information about predicate expression syntax, see "Predicates in Policy Rules".

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters and predicates to their previous values.
Apply	Click this button to apply any changes made to this page.
Cancel	Click this button to cancel any changes made to this page and return to the main Policy page.
OK	Click this button to return to the main Policy page. Any changes made to this page are automatically saved.

Related Topics

- Edit (Editing Policies)
- Adding Predicates
- Reordering Predicates
- Delete (Deleting Predicates)

Configuration Management -- General

Use the **General** page of the **Configuration Management** tab to view database and directory information, enable publishing certificates to the directory, enable logging and tracing, and to specify default values for distinguished name (DN) components.

You can view or configure the following parameters:

Certificate Publishing

When "Publish" is checked, OracleAS Certificate Authority automatically stores certificates in the directory when they are issued, and automatically deletes them when revoked. OracleAS Certificate Authority connects to the directory using SSL.

SSL and SSO Authentication

By default, users who are authenticated by SSL or OracleAS Single Sign-On can automatically issue, revoke, or renew their own certificates. You can disable this feature by unchecking Enable SSL Authentication or Enable SSO Authentication.

Default usage for client certificates

The value you choose here appears as the selected usage when a client requests a certificate. This does not prevent the user from selecting a different usage from the drop-down list, which includes authentication, encryption, signing, and combinations of these, plus CA signing, and code signing.

Subject Alternate Name Extension

For SSO users, the value chosen for this extension appears in the certificate to enable e-mail encryption, signing, or use by other applications. Your choices are shown in Extension Content Choice.

Extension Content Choice

Choose from None, Email, Principal Name (UID), or Email/Principal Name (UID). The choice made here appears in the certificate as the subject alternate name, enabling e-mail encryption, signing, or use by other applications. (UID means user identifier or unique identifier.) Choosing "Email/Principal Name (UID)" causes both to be listed in the certificate.

Mandatory

If this box is checked, the Subject Alternate Name Extension is required for all SSO-authenticated certificates. If an e-mail address or Principal Name cannot be found in Oracle Internet Directory for the user named in an SSO-authenticated certificate request, that request will be denied. An error message will state that an SSO-authenticated certificate could not be issued because an e-mail account was not found in the Oracle Internet Directory, and that the requestor should contact the administrator.

Logging and Tracing

Allows you to enable logging or tracing. OracleAS Certificate Authority server logs error information for all components it manages. By default, logging is enabled.

Choose Enable Logging to write system events and error messages to the Certificate Authority log table, viewable from the View Logs tab of the administrator web interface.

Choose Enable Tracing to record debugging messages for Oracle Support to `ORACLE_HOME/oca/logs/admin.trc` (for tracing command line actions) and `ORACLE_HOME/oca/logs/oca.trc` (for tracing web actions). This information is not intended for administrator use.

Default Base DN Components

If most of the DNs specified in enrollment requests have identical components (except the unique identifier component), then you can specify them here. Then manual enrollment request form fields generated by OracleAS Certificate Authority will pre populate with these default components, which users can overwrite if necessary. All fields are optional.

Database Settings

Displays the database connect string used to connect to the OracleAS Certificate Authority repository. This field is read-only.

In this section, you can also specify settings for the Database Pool Size and Database Pool Scheme, as follows:

Database Pool Size: Enter here the number of connections to the database (default: 10) that represents your expectation of how many users will be accessing OracleAS Certificate Authority concurrently. When a user in that first group exits OracleAS Certificate Authority, his connection becomes available to the next new user. For each user beyond that number, a new connection will be opened, to be closed as soon as that user has exited OracleAS Certificate Authority.

Database Pool Scheme: The default, "Fixed wait scheme", means that after 10 (the default pool size, or the number you specify) users are connected to OracleAS Certificate Authority, every subsequent user attempting to connect simply waits until one of the original 10 exits. The "dynamic" choice causes a new connection to be opened immediately for the new user, and after that user exits OracleAS Certificate Authority, that connection is closed. "Fixed Increment" means that after the original pool size limit is reached, a new connection is opened for each new user, up to a secondary limit, after which no new user can connect until an existing OracleAS Certificate Authority user exits.

Directory Settings

Displays directory host machine, listener port, and the bind DN that has privileges on the directory host port (the OracleAS Certificate Authority LDAP agent that publishes users' certificates to Oracle Internet Directory). All fields are read-only.

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters to their previous values.

Button	Description
Cancel	Click this button to cancel any changes made to this page and return to the main Configuration Management page (Notification).
OK	Click this button to save any changes made to this page. A message confirming that the configuration file was updated appears at the top of the screen. Restart the server as described in Starting and Stopping Oracle Application Server Certificate Authority for the changes to take effect.

Related Topics

- Configuration Management -- General
- Certificate Publishing
- Logging and Tracing in Configuration Management -- General
- Setting the Default Base DN Components in Configuration Management -- General

Configuration Management -- Notification

Use the **Notification** page of the **Configuration Management** tab to configure e-mail server host name and e-mail templates for automatic notifications. You can also use this page to enable administrator alerts, schedule timed jobs to automatically generate CRLs (Certificate Revocation Lists), and to synchronize OracleAS Certificate Authority with the directory.

Notifications are sent to users after OracleAS Certificate Authority processing events, such as certificate requests, revocations, or renewals. Alerts are sent to administrators for events like CRL generation failure, or when the pending request queue size is greater than the specified threshold.

You can configure the following parameters:

Mail Details

Use this region to specify your outgoing e-mail (SMTP) server host name, the administrator's name and e-mail address to display on notification e-mails, and the e-mail address to which administrator alerts should be sent. You can also enable secure MIME protocol use and e-mail message body templates, as explained in "Email Templates".

Note: The e-mails sent by OracleAS Certificate Authority are signed using the server's S/MIME wallet, which is stored in `ORACLE_HOME/oca/wallet/smime`.

Alerts

Note: to enable alerts, Mail Details information must be specified.

Use this region to enable administrator alerts as follows:

- To send administrator alerts for certificate processing events, check Enable Alerts. To enable other types of alerts, you must check this box and one or both of the following:
- To send administrator alerts when the request queue reaches a specified size, check Pending Requests Queue over Threshold. Specify that size (number of

certificate requests) in Queue Size Threshold. When you enable this alert, you must also specify the first time the server should check the queue size and how often thereafter, as follows:

  - In Queue Size Check Start Time, enter the start time using a 24-hour clock time (default is midnight), for example 2 hours 30 minutes for 2:30 in the morning, or 14 hours 30 minutes for 2:30 in the afternoon.
  - In Interval Between Queue Size Checks, enter the interval (default one day) to be added to that time to specify the time of the next check; it must be nonzero.
  - Changes survive restarts.

- To send administrator alerts if automatic CRL generation fails, check CRL Auto Generation Failure.

Scheduled Jobs

Use this region to schedule timed automatic jobs as follows:

- To Enable Automatic Generation of CRL, click the check box next to that label. Specify the first such generation and the intervals thereafter as follows:
  - In CRL Auto Generation Start Time, enter the start time using a 24-hour clock time (default is midnight), for example 2 hours 30 minutes for 2:30 in the morning, or 14 hours 30 minutes for 2:30 in the afternoon.
  - In CRL Auto Generation Interval, enter the interval (default one day) to be added to that time to specify the time of the next CRL generation; it must be nonzero.
  - In CRL Auto Generation Validity, enter the number of days each CRL will be considered valid.
  - Changes survive restarts.
- To enable automatic synchronization with the directory, check Synchronize Directory and specify the start time for the first such synchronization and subsequent intervals as follows:
  - In Synchronize Directory Start Time, enter the start time using a 24-hour clock time (default is midnight), for example 2 hours 30 minutes for 2:30 in the morning, or 14 hours 30 minutes for 2:30 in the afternoon.
  - In Synchronize Directory Interval, enter the interval (default one day) to be added to that time to specify the time of the next CRL generation; it must be nonzero.
  - Changes survive restarts.

  Synchronizing with the directory deletes all expired certificates from the directory, and publishes all certificates and CRLs to the directory, which may have failed due to system error, such as the directory being temporarily unavailable.

The buttons at the bottom of the page perform the functions described in the following table:

Button	Description
Revert	Click this button to reset all changed parameters to their previous values.

Button	Description
Cancel	Click this button to cancel any changes made to this page and refresh the Notification page.
OK	Click this button to save any changes made to this page. A message confirming that the configuration file was updated appears at the top of the screen. Restart the server as described in Starting and Stopping Oracle Application Server Certificate Authority for the changes to take effect.

Configuration Management -- Policy

Use the **Policy** page of the **Configuration Management** tab to manage policies. Policies displayed on this page apply to the operation shown in the **View Policies for** list located in the upper left corner. When you select Requests, Renewals, or Revocations from the list, all policies that apply to your chosen operation will display. Each policy's **Type**, **Status**, and **Description** are displayed. A **Default Policy** is one which ships with OracleAS Certificate Authority; a **Custom Policy** is one you write yourself. Default policies cannot be deleted (only disabled).

Select a policy to edit, enable, disable, or delete by clicking the radio button to its left and then clicking the desired action on the upper right above the policy list. You can view policies by clicking **Edit** to display its details. To change the order in which the displayed policies are applied, or to add a new one, click the **Reorder** or **Add** buttons.

For detailed information about policies, see Chapter 6.

Changes to the policy configuration do not take effect until you restart the server as described inStarting and Stopping Oracle Application Server Certificate Authority.

To learn more about the policy management tasks you can perform on this page, please see the following pages:

Related Topics

- Edit Policies
- Enable or Disable Policies
- Adding Custom Policies
- Delete Policies
- Reordering Policies

Update Certificate Revocation List

Generates a new certificate revocation list (CRL). CRLs, defined by the X.509 standard, are signed data structures containing a list of all revoked certificates. Before granting users access, applications use this list to identify whether certificates are valid.

A description of the fields and buttons on this page follows:

Field or Button	Description
CRL Validity	Set the number of days that the new CRL will be valid

Field or Button	Description
Signature Algorithm	Choose the algorithm used to sign the CRL, for example, SHA1 with RSA or MD5 with RSA.
Cancel	Exit without making changes
OK	Update the CRL with all certificates revoked since the last CRL update

To learn how to revoke a certificate, please see Revoking Certificates.

Welcome to the OracleAS Certificate Authority Administration Pages

The tabs in this window help you to navigate around the OracleAS Certificate Authority Administrative pages:

- The Home tab returns you to this page.
- The Certificate Management tab enables you to approve or reject certificate requests.
- The Configuration Management tab enables you to set up notifications, alerts, certificate revocation list generation, logging/tracing, and manage certificate policies.
- The View Logs tab enables you to search and view error logs.

Shortcuts for the tabs are located along the page bottom. Click Practice Statement to view your site's certification practice statement, which you can add by editing the `ORACLE_HOME/j2ee/oca/applications/ocaapp/oca/helpsets/Help/ocaadmin_cs_practicestmt.html` file (UNIX) or `ORACLE_HOME\j2ee\oca\applications\ocaapp\oca\helpsets\Help\ocaadmin_cs_practicestmt.html` (Windows).

Web Administrator Enrollment

Use this page to request a certificate. An administrator who performs certificate management functions must be a certificate holder.Use this page to request a certificate by following these steps:

1. Enter Distinguished Name Information

   The distinguished name (DN) is the location of a user's entry in Oracle Internet Directory. OracleAS Certificate Authority uses the directory entry to store and retrieve the user's certificate. A blue asterisk identifies the fields required under this heading. These are as follows:

   - Common Name

     The name of the OracleAS Certificate Authority administrator

   - Organization

     The company to which the administrator belongs

2. Enter the Admin Password

Enter the administrator password for OracleAS Certificate Authority.

3. Enter Certificate Information

   The fields under this heading enable you to specify the how strong the certificate key is and how long the certificate is valid. If you are using Internet Explorer, you designate the storage mechanism instead of the key strength.

   - Certificate Key Size (Netscape Communicator/Mozilla/Safari users)

     The length of the private key that will be generated by your browser. Choose a key strength from the available options, typically 512 (low grade), 1024 (medium grade), or 2048 (high grade). *Note*: Not all options are available on all browsers.

   - Cryptographic Service Provider (Internet Explorer users)

     The type of certificate storage or the key size. Click the drop-down list box to choose one of: **Microsoft Base Cryptographic Provider, Microsoft Enhanced Cryptographic Provider**, or **Microsoft Strong Cryptographic Provider**.

     Choose a smart card only if you have a corresponding smart card device installed on your system. If, for example, you have a Gemplus smart card reader installed, you may choose Gemplus GemSAFE Card CSP. Please note that this option is not appropriate without that reader.

   - Validity Period (all browsers)

     The length of time the certificate is valid. Click the drop-down list box to choose one of four alternatives.

4. Click **Submit** to process your request. Click **Reset** to start over.

   The Approved Certificate Information page appears. It contains detailed information about the certificate.

5. Click **Install in Browser** to install the certificate to your browser. Please note that this installation process differs between browsers:

   - Netscape Communicator/Mozilla

     When you click **Install in Browser**, the certificate is installed with the corresponding CA's certificate. No message appears informing you that the process is complete, although the browser status bar displays "Document:Done."

   - Internet Explorer

     When you click **Install in Browser**, the certificate is installed with the corresponding CA's certificate. The browser displays the message "Certificate has been imported successfully." After installation, you are asked whether the signer's certificate must be imported. Internet Explorer displays a window that contains details about the CA being imported. Use this window to choose whether to import the signer.

   - Safari

     You cannot install the certificate directly in the browser. Follow these steps to install the certificate:

     – Go to the web user interface https://hostname:port/oca/user.

     – Go to **User Certificates > Manual Authentication**.

     – Search for the web administrator's certificate using the serial number you noted earlier.

- Select the certificate and click **View Details**.
- Copy the BASE64 encoded certificate (not the BASE64 encoded certificate with the CA chain in PKCS#7 format), and save it into a file with the appropriate extension (`.pem/.der/.cer`).
- Double click the file. The keychain access utility opens up with a pop-up dialog, asking you if you want to import the certificate in the keychain. (*Note*: Your system will have more than one keychain, but be sure to import it into the default "login" keychain which is in an unlocked state.)
- There is a button to view the certificate. View it to verify if it is the web administrator's certificate. Click OK to import the certificate into the keychain.

After installing the administrator's certificate, you should see the Certificate Management and the Configuration Management tabs in the administrative Web interface.

For more detailed enrollment instructions, please see Chapter 4. If you need to change the administrator, then again use Chapter 4 or see the following help topic:

Web Administrator Enrollment

View Logs

Use this page to search Oracle Application Server Certificate Authority error logs. The Certificate Authority server logs error messages for all components it manages. After you have entered your search criteria, the table displays all messages that match it. Enter your search criteria as follows:

1. Choose to search by Client Address (IP address) or Message content. To search by message content, enter information such as a DN or username.
2. Click Go.

The most recent messages that match your search criteria display in the View Logs table ten messages on each page.

Related Topic

Logging and Tracing in Configuration Management -- General

Windows and Fields in the End-User Interface

This section lists and describes the windows and fields in the Web user interface.

Advanced Search Screen

Use the Advanced screen to narrow or refine your search for certificate requests or existing certificates. The Advanced screen offers the following three search methods:

1. **Search Using Distinguished Name (DN)**

   Enter the components of the certificate requester's or certificate holder's distinguished name. The Search list enables you to toggle between certificate requesters and certificate holders.

2. **Search Using Advanced Distinguished Name**

   Choose this option if you know the certificate requester or certificate holder's full DN and understand how to enter it in LDIF format. You need to enter a

contiguous DN to get results. For example, "cn=Margarita Redmond,ou=sales,o=yourcorp" is acceptable but "cn=Margarita Redmond,,o=yourcorp" is not. Please note in the following example that spaces between attributes are optional.

cn=Margarita Redmond,ou=sales,o=yourcorp,l=Bismarck,st=SD,c=US

3. **Search Using Serial No./Request ID Range**

   Choose this option to retrieve information about certificates that fall within a given serial number range. Use the Search list to toggle between requested certificates and existing certificates.

To learn how to search using these methods, see the Advanced Search Screen.

Authentication Page

Use the Authentication page to identify yourself to the OracleAS Certificate Authority server. The mode that you choose is dictated by your existing OracleAS credentials. The modes, represented as radio buttons, are as follows:

- Use your OracleAS Single Sign-On name and password

  Use this option if you are an OracleAS Single Sign-On user and need to obtain or revoke a digital certificate.

- Use your existing certificate

  Use this option if you have a valid certificate issued by the current OracleAS Certificate Authority. If you have such a certificate, then you can identify yourself to the Certificate Authority using the Secure Sockets Layer (SSL) protocol.

- Use manual approval / authentication

  Use this option if you are not using OracleAS Single Sign-On or the SSL protocol for identification and need to obtain a digital certificate. The administrator will manually verify your identity before issuing your certificate.

To learn how to perform the tasks introduced on this page, see the following topics:

- Certificate Request Form - SSL Authentication
- SSO Certificate Request Form
- Certificate Request Form (Manual Request)

CA Certificate Details

This page displays the certificate authority (CA) certificate in BASE64 format as well as the "BASE64-Encoded Certificate with CA certificate chain in PKCS#7 format". You can copy and paste the encoded CA certificate into Oracle Wallet Manager when using that tool to create a PKCS#10 certificate request. You must use this method to request a server certificate or a subordinate CA certificate.

Save CA Certificate

Use this screen to install the certificate authority (CA) certificate into your browser. To see the CA certificate in BASE64 or PKCS #7 encoding, click Advanced. To install the CA certificate into your browser, click Install in Browser. If the current CA is a subordinate CA, its ancestor CA certificates will also be present. You can use this form to import the CA certificates into Oracle Wallet Manager (OWM). The PKCS #7 encoding contains the whole certificate chain.

This page also displays the following CA certificate details:

Field	Description
Status	VALID indicates that the certificate can still be trusted. This is the only value that will appear here.
Serial Number	The number used to reference the certificate.
Signature Algorithm	The algorithm used, which is indicated by the object identification number (OID).
Usage	The certificate's function. In the case of a CA certificate, these values are always "Certificate Signing" and "CRL Signing."
Issuing Authority	The CA that issued the certificate. If the root CA issued the certificate, the requester and the issuer are the same.
Subject DN	Distinguished name of the certificate holder.
Not Valid Before	Date and time certificate became valid.
Not Valid After	Date and time certificate expires.

Related Topic

- Save CA Certificate

Certificate Approval--Single Sign-On, SSL

Use this page to view details about your new certificate and to install it to your browser (by clicking Install in Browser). Clicking OK again after you install the certificate returns you to the User Certificates page. The new certificate appears on the Certificates bar of this page. The Certificate Approval page has the following fields:

Field	Description
Status	This value is always VALID.
Serial Number	This is the certificate's serial number.
Signature Algorithm	The algorithm used, which is indicated by the object identification number (OID).
Usage	The certificate function.
Issuing Authority	The CA that issued the certificate.
Subject DN	Distinguished name of the certificate holder.
Not Valid Before	Data and time certificate became valid.
Not Valid After	Date and time certificate expires.

Choose one of the buttons located at the bottom of the page to perform your desired task:

Button Name	Function Description
OK	Returns you to the main page of the User Certificates tab.
Install in Browser	Installs the certificate into your browser.
Save to Disk	Saves the certificate to a file on your local system.

Certificate Details

Use this page to obtain a complete description of a certificate, including its BASE64 encoding. The non-modifiable fields on this page are as follows:

Field	Description
Status	This value is `VALID`, `REVOKED`, or `EXPIRED`
Serial Number	This is the certificate's serial number assigned by OracleAS Certificate Authority.
Signature Algorithm	The object identification number (OID) representing the algorithm used, such as MD5 with RSA encryption.
Usage	The certificate function.
Issuing Authority	The CA that issued the certificate.
Subject DN	Distinguished name of the certificate holder.
Not Valid Before	Data and time certificate became valid.
Not Valid After	Date and time certificate expires.
BASE64 Encoded Certificate	The encoded certificate.
BASE64-Encoded Certificate with CA certificate chain in PKCS#7 format	The encoded certificate plus its tree of trusted authorities, in PKCS#7 format. This form allows a single operation to transport all certificates in the trusted chain up through the root CA certificate.

Choose one of the buttons located at the bottom of the page to perform your desired task:

Button Name	Function Description
OK	Returns you to the main page of the User Certificates tab.
Revoke	Revokes the certificate. You must specify a revocation reason.

Button Name	Function Description
Renew	Renews the certificate. You must specify a new validity period.
Install in Browser	Installs the certificate into your browser.

Certificate Request Form

Use this form to request a certificate manually. The Certificate Request form has the following headings:

Distinguished Name Information

The distinguished name (DN) is the location of a user's entry in Oracle Internet Directory. OracleAS Certificate Authority uses the directory entry to store and retrieve the user's certificate. A blue asterisk identifies the fields required under this heading. These are as follows:

- **Common Name**

  The name of the certificate requester

- **Organization**

  The company to which the certificate requester belongs

Contact Information

The certificate requester's name, e-mail address, and phone number. Please note that the Name field and either the E-Mail ID or the Phone No. field require input.

Certificate Information

Use the fields under this heading to specify the certificate key size or storage mechanism, the certificate function, and the certificate's life span. A description of these fields follows.

- **Certificate Key Size** (for Netscape Communicator/Mozilla/Safari users)

  The length of the private key that will be generated by your browser. Choose a key strength from the available options, typically 512 (low grade), 1024 (medium grade), or 2048 (high grade). *Note*: Not all options are available on all browsers.

- **Cryptographic Service Provider** (for Internet Explorer users)

  The type of certificate storage. Click the list to choose one of several storage methods, which determines the key strength. Choose between **Microsoft Base Cryptographic Provider**, **Microsoft Enhanced Cryptographic Provider**, and **Microsoft Strong Cryptographic Provider**. Choose a smart card only if you have a corresponding smart card device installed on your system. If, for example, you have a Gemplus smart card reader installed, you may choose **Gemplus GemSAFE Card CSP**. Please note that this option is not appropriate without that reader.

- **Certificate Usage** (for all browser types)

  The function of the certificate. Choose a usage that fits your intended applications and your enterprise policies; if unsure, choose "Authentication, Encryption, and Signing." (The default for your site is preselected.) The following list shows your possible choices:

Function	Description
Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.)
Encryption	Enables encrypting and decrypting electronic documents.
Signing	Enables verifiable signature for (and assures non-tampering of) electronic documents, including e-mail (using S/MIME, the Secure Multipurpose Internet Mail Extension)
Authentication, Encryption	Certificate can be used for both purposes.
Authentication, Signing	Certificate can be used for both purposes.
Authentication, Encryption, and Signing	Certificate can be used for all three purposes.
Encryption, Signing	Certificate can be used for both purposes.
CA Signing	Enables requesting subordinate CA certificates
Code Signing	Provides verifiable signature for the provider of (and assures non-tampering of) Java code, JavaScript, and other signed files.

- **Validity Period** (for users of all browser types)

  The length of time the certificate is valid. Click the list to choose one of four alternatives. A certificate is valid up to 10 years.

Related Topic

- Certificate Request Form (Manual Request)

Certificate Revocation List

This page displays the current certificate revocation list. It indicates when the list was last updated. The list shows the serial number and revocation date for each revoked certificate.

Use the buttons on this page to install the CRL into your browser or to save it either as a binary file or as a BASE64-encoded text file. (BASE64 encoding text files make it easier to copy, paste, or e-mail the information.)

After installing or saving the CRL as you choose, click **OK** to return to the **User Certificates** page.

Revocation Reason

Use this page to choose a revocation reason. Here is a description of the available options.

Revocation Reason	Description
Key Compromise	The user's private key has been lost or has been exposed.

Revocation Reason	Description
Affiliation Change	The organization has decided to use a different root CA.
CA Compromise	The CA has been replaced by a sub-CA or the CA certificate has been compromised.
Certificate Hold	The certificate is temporarily suspended.
Cessation of Operation	The existing root CA has ceased operations. A new root CA is required.
Remove from CRL	The certificate has been removed from the certificate revocation list (CRL).
Superseded	The root CA's certificate has been replaced. The old certificate must be removed and the new one installed
Unspecified	No reason available or provided.

To learn how to revoke a certificate, please see Revoking Certificates.

Certificate Request Form--Advanced

Use this form to request a certificate if you know your full distinguished name (DN)--if it already exists--and understand how to enter it in LDIF format. This feature is a shortcut for the **Distinguished Name Information** heading on the standard Certificate Request form, where it appears as the link **Advanced DN**. The Advanced form supports the same DN components that the standard form supports. The DN is the location of your user entry in Oracle Internet Directory. OracleAS Certificate Authority stores your certificate in and retrieves it from your directory entry.

Related Topic

Certificate Request Form (Manual Request)

Server/SubCA Certificates

Use this page to search for and display information about certificates and certificate requests, or to request a new server or SubCA certificate. Clicking **Request a Certificate** brings up the **Server/SubCA Certificates** form for you to fill in.

When a search you specified brings up a list of certificates or certificate requests, you can see more details for a particular entry by clicking its **Select** button (far left) and then clicking **View Details**. To show search results beyond the first 25, you can click **Next 25** or click in the drop-down list to select the range you wish to display.

Related Topics

The functions you can select using the buttons on the Server/SubCA Certificates page are explained at the following links:

- Listing a Single Certificate Request or Issued Certificate
- Server/SubCA Certificates Tab
- Update Certificate Revocation List
- Save CA Certificate

Server/SubCA Certificate Request

Use this form to request a certificate for a Web server or a subordinate certificate authority. The Server/SubCA Certificate Request form has the following headings:

Certificate Request

You request a certificate by using the openSSL reqtool or Oracle Wallet Manager to generate a certificate in PKCS#10 encoding in BASE64 format. Then paste the encoded certificate request in the PKCS#10 Request field

Contact Information

The certificate requester's name, e-mail address, and phone number. Note that the **Name** field and either the **E-Mail ID** or the **Phone No.** field require input.

Certificate Information

Use the fields under this heading to specify the certificate function and life span. A description of these fields follows.

- **Certificate Usage**

  The function of the certificate. Choose a usage that fits your intended applications and your enterprise policies; if unsure, choose "Authentication, Encryption, and Signing." (The default for your site is preselected.) The following list shows your possible choices:

Function	Description
Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.)
Encryption	Enables encrypting and decrypting electronic documents.
Signing	Enables verifiable signature for (and assures non-tampering of) electronic documents, including e-mail (using S/MIME, the Secure Multipurpose Internet Mail Extension)
Authentication, Encryption	Certificate can be used for both purposes.
Authentication, Signing	Certificate can be used for both purposes.
Authentication, Encryption, and Signing	Certificate can be used for all three purposes.
Encryption, Signing	Certificate can be used for both purposes.
CA Signing	Enables requesting subordinate CA certificates
Code Signing	Provides verifiable signature for the provider of (and assures non-tampering of) Java code, JavaScript, and other signed files.

- **Validity Period**

  The length of time the certificate is valid. Click the list to choose 6 months, one year, or five years for the validity period.

Related Topic

- Server/SubCA Certificates Tab

Certificate Request Form - SSL Authentication

Use this form if you already have a certificate and want to request another one--either because you want a different key size or storage mechanism or because you want to use the certificate for a different purpose. This form has the following headings and fields:

Distinguished Name Information

The **User DN** field under the **Distinguished Name Information** heading, displays the DN under which the first certificate was assigned. You cannot modify this field.

Certificate Information

Use the fields under this heading to specify the certificate key size or storage mechanism, the certificate function, and the certificate's life span. A description of these fields follows.

- **Certificate Key Size** (Netscape Communicator/Mozilla/Safari users)

  The length of the private key that will be generated by your browser. Choose a key strength from the available options, typically 512 (low grade), 1024 (medium grade), or 2048 (high grade). *Note*: Not all options are available on all browsers.

- **Cryptographic Service Provider** (Internet Explorer users)

  The type of certificate storage or the key size. Click the drop-down list box to choose one of: **Microsoft Base Cryptographic Provider, Microsoft Enhanced Cryptographic Provider**, or **Microsoft Strong Cryptographic Provider**. Choose a smart card only if you have a corresponding smart card device installed on your system. If, for example, you have a Gemplus smart card reader installed, you may choose **Gemplus GemSAFE Card CSP**. Please note that this option is not appropriate without that reader.

- **Certificate Usage** (users of all browser types)

  The function of the certificate. Choose a usage that fits your intended applications and your enterprise policies; if unsure, choose "Authentication, Encryption, and Signing." (The default for your site is preselected.) The following list shows your possible choices:

Function	Description
Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.)
Encryption	Enables encrypting and decrypting electronic documents.
Signing	Enables verifiable signature for (and assures non-tampering of) electronic documents, including e-mail (using S/MIME, the Secure Multipurpose Internet Mail Extension)
Authentication, Encryption	Certificate can be used for both purposes.
Authentication, Signing	Certificate can be used for both purposes.

Function	Description
Authentication, Encryption, and Signing	Certificate can be used for all three purposes.
Encryption, Signing	Certificate can be used for both purposes.
CA Signing	Enables requesting subordinate CA certificates
Code Signing	Provides verifiable signature for the provider of (and assures non-tampering of) Java code, JavaScript, and other signed files.

Related Topic

- Certificate Request Form - SSL Authentication

SSO Certificate Request Form

Use this form if you have been authenticated by an OracleAS Single Sign-On server and want to request a new certificate or an additional one. The SSO Certificate Request form has the following headings and fields:

Distinguished Name Information

The **User DN** field under the **Distinguished Name Information** heading, displays the DN under which your certificates are issued. You cannot modify this field.

Certificate Information

Use the fields under this heading to specify the certificate key size or storage mechanism, the certificate function, and the certificate's life span. A description of these fields follows.

- **Certificate Key Size** (Netscape Communicator/Mozilla/Safari users)

  The length of the private key that will be generated by your browser. Choose a key strength from the available options, typically 512 (low grade), 1024 (medium grade), or 2048 (high grade). *Note*: Not all options are available on all browsers.

- **Cryptographic Service Provider** (Internet Explorer users)

  The type of certificate storage or the key size. Click the drop-down list box to choose one of: **Microsoft Base Cryptographic Provider, Microsoft Enhanced Cryptographic Provider**, or **Microsoft Strong Cryptographic Provider**. Choose a smart card only if you have a corresponding smart card device installed on your system. If, for example, you have a Gemplus smart card reader installed, you may choose **Gemplus GemSAFE Card CSP**. Please note that this option is not appropriate without that reader.

- **Certificate Usage** (users of both browser types)

  The function of the certificate. Choose a usage that fits your intended applications and your enterprise policies; if unsure, choose "Authentication, Encryption, and Signing." (The default for your site is preselected.) The following list shows your possible choices:

Function	Description
Authentication	Enables secure identification when requesting or providing access or services, such as when logging into an enterprise portal. (Typically, SSL protocol is used.)

Function	Description
Encryption	Enables encrypting and decrypting electronic documents.
Signing	Enables verifiable signature for (and assures non-tampering of) electronic documents, including e-mail (using S/MIME, the Secure Multipurpose Internet Mail Extension)
Authentication, Encryption	Certificate can be used for both purposes.
Authentication, Signing	Certificate can be used for both purposes.
Authentication, Encryption, and Signing	Certificate can be used for all three purposes.
Encryption, Signing	Certificate can be used for both purposes.
CA Signing	Enables requesting subordinate CA certificates
Code Signing	Provides verifiable signature for the provider of (and assures non-tampering of) Java code, JavaScript, and other signed files.

Related Topic

To learn how to use your single sign-on user name and password to request a certificate, see

- SSO Certificate Request Form

User Certificates - Manual Authentication

You can use the **Request a Certificate** button on this page to request a certificate. This page also has buttons that enable you to save the Certificate Authority's (CA) certificate or Certificate Revocation List (CRL). In addition, you can change the authentication mode that you use to request a certificate. Clicking **Change Authentication** returns you to the Authentication page, where you can make another choice.

After you have submitted your certificate request, you can use search features on the User Certificates - Manual Authentication page to check the status of your request.

Related Topics

- Listing a Single Certificate Request or Issued Certificate
- Server/SubCA Certificates Tab
- Update Certificate Revocation List
- Save CA Certificate
- User Certificates - Manual Authentication

User Certificates - SSL Authentication

If you gained access to OracleAS Certificate Authority with an existing certificate, then you can use the **Get Certificate** button on this page to add a certificate. The form also has buttons that enable you to save a certificate revocation list (CRL) or change the authentication mode that you use to request a certificate. Clicking **Change**

Authentication returns you to the Authentication page where you can make the change.

All of your valid certificates are displayed in the master table under the **Certificates** bar. Each row contains information about a particular certificate, including serial number, life span and usage type. Click the button in the far left column of a row to view additional details about a certificate or to revoke it. You should revoke your certificate if your private key is lost, corrupted, or stolen.

Related Topics

- Certificate Request Form - SSL Authentication to learn how to use an existing certificate to add a certificate
- Certificate Revocation List to learn how to save a CRL

User Certificates - SSO Authentication

Use the **Get Certificate** button on this page to request or add a new certificate. This page also has buttons that enable you to save the Certificate Revocation List (CRL) or change the authentication mode that you use to request a certificate. Clicking **Change Authentication** returns you to the Authentication page, where you can make your choice.

All of your valid certificates are displayed in the master table under the **Certificates** bar. Each row contains information about a particular certificate, including serial number, life span and usage type. To view additional details about a certificate, or to revoke it, click the button in the **Select** column of that certificate's row and then click **View Details**. You should revoke your certificate if your private key is lost, corrupted, or stolen.

Related Topics

- SSO Certificate Request Form to learn how to use your single sign-on user name and password to request a certificate
- Certificate Revocation List to learn how to save a CRL

Welcome to the OracleAS Certificate Authority User Pages

The user pages for Oracle Certificate Authority possess features that enable you to request, view, and revoke X.509 certificates and to save certificate revocation lists. Use the Authentication page to access these tabs. The tabs in this window help you to navigate around the Oracle Certificate Authority user pages:

- The **Home** tab returns you to this page.
- The **User Certificates** tab enables you to view and revoke your certificates, create certificate requests, change your authentication method, and save certificate revocation lists.
- The **Server/SubCA** tab enables you to search for certificates and certificate requests by using the Server/SubCA Certificate form. You can also use this form to request PKCS#10-encoded certificates for Web servers or subordinate certificate authorities, and to save or install certificate authority certificates from Oracle Certificate Authority.
- The **View Logs** tab enables you to search error, warning, and audit logs.

This page also enables you to do any combination of the following four tasks:

- **Install the certificate authority certificate into your browser:**

  - In Netscape, the **New Certificate Authority** dialog box appears:

    Click through the dialog boxes, which help you decide whether to accept the certificate. In the last one, to accept the certificate, click **Finish**; to reject the certificate or to postpone acceptance, click **Cancel**.

  - In Internet Explorer, a warning dialog box tells you the file's name, type, and source, and asks if you want to open or save it. Click **Save** and choose a file system destination.

  - In Safari, the certificate cannot be imported into the browser directly. Follow the instructions in **Save the certificate authority certificate to your file system** to install the certificate.

- **Install certificate revocation lists into your browser:**

  - In Netscape, installing the CRL into your browser brings up a dialog box stating that the CRL was successfully imported It also tells you who issued it, when the next update is, and offers to enable automatic CRL generation if you say so. (If you do, you can specify when and how often the update occurs.)

  - In Internet Explorer, a warning dialog box tells you the file's name, type, and source, and asks if you want to open or save it. Click **Save** and choose a file system destination.

- **Save the certificate authority certificate to your file system:**

  - In Netscape, saving the certificate authority certificate to your file system brings up a dialog box asking what to do with this file (OCABase64.cert). Ensure that **Save it to disk** is selected, and then click **OK**. In the "save" dialog that appears, select where you want to store it and click **Save**.

  - In Internet Explorer, a warning dialog box tells you the file's name, type, and source, and asks if you want to open or save it. Click **Save** and choose a file system destination.

  - In Safari, a page appears which shows the BASE64 encoded certificate of the Certificate Authority. Copy it and paste it into a `.der/.pem/.cer` file. Double click the file. The keychain access utility opens up with a pop-up dialog, asking you if you want to import the certificate in the keychain. (*Note*: Your system will have more than one keychain, but be sure to import it into the default "login" keychain which is in an unlocked state.)

- **Save certificate revocation list to your file system:**

  - In Netscape, saving the Certificate Revocation List to your file system brings up a dialog box asking what to do with this file (OCAcrlBase64.txt). Ensure that **Save it to disk** is selected, and then click **OK**. In the "save" dialog that appears, select where you want to store it and click **Save**.

  - In Internet Explorer, a warning dialog box tells you the file's name, type, and source, and asks if you want to open or save it. Click **Save** and choose a file system destination.

Glossary

3DES

See **Triple Data Encryption Standard (3DES)**.

access control item (ACI)

Access control information represents the permissions that various entities or subjects have to perform operations on a given object in the directory. This information is stored in Oracle Internet Directory as user-modifiable operational **attribute**s, each of which is called an access control item (ACI). An ACI determines user access rights to directory data. It contains a set of rules for controlling access to entries (structural access items) and attributes (content access items). Access to both structural and content access items may be granted to one or more users or groups.

access control list (ACL)

A list of resources and the usernames of people who are permitted access to those resources within a computer system. In Oracle Internet Directory, an ACL is a list of **access control item (ACI) attribute value**s that is associated with directory objects. The attribute values on that list represent the permissions that various directory user entities (or subjects) have on a given object.

access control policy point (ACP)

A directory entry that contains access control policy information that applies downward to all entries at lower positions in the **directory information tree (DIT)**. This information affects the entry itself and all entries below it. In Oracle Internet Directory, you can create ACPs to apply an access control policy throughout a **subtree** of your directory.

account lockout

A security feature that locks a user account if repeated failed logon attempts occur within a specified amount of time, based on security policy settings. Account lockout occurs in OracleAS Single Sign-On when a user submits an account and password combination from any number of workstations more times than is permitted by Oracle Internet Directory. The default lockout period is 24 hours.

ACI

See **access control item (ACI)**.

ACL

See **access control list (ACL)**.

ACP

See **access control policy point (ACP)**.

administrative area

A **subtree** on a directory server whose entries are under the control of a single administrative authority. The designated administrator controls each **entry** in that administrative area, as well as the directory **schema**, **access control list (ACL)**, and **attribute**s for those entries.

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a **symmetric cryptography** algorithm that is intended to replace **Data Encryption Standard (DES)**. AES is a Federal Information Processing Standard (FIPS) for the encryption of commercial and government data.

advanced replication

See **Oracle Database Advanced Replication**.

advanced symmetric replication (ASR)

See **Oracle Database Advanced Replication**.

AES

See **Advanced Encryption Standard (AES)**.

anonymous authentication

The process by which a directory authenticates a user without requiring a user name and password combination. Each anonymous user then exercises the privileges specified for anonymous users.

API

See **application programming interface (API)**.

application programming interface (API)

A series of software routines and development tools that comprise an interface between a computer application and lower-level services and functions (such as the operating system, device drivers, and other software applications). APIs serve as building blocks for programmers putting together software applications. For example, LDAP-enabled clients access Oracle Internet Directory information through programmatic calls available in the LDAP API.

application service provider

Application Service Providers (ASPs) are third-party entities that manage and distribute software-based services and solutions to customers across a wide area network from a central data center. In essence, ASPs are a way for companies to outsource some or almost all aspects of their information technology needs.

ASN.1

Abstract Syntax Notation One (ASN.1) is an International Telecommunication Union (ITU) notation used to define the syntax of information data. ASN.1 is used to describe structured information, typically information that is to be conveyed across some communications medium. It is widely used in the specification of Internet protocols.

ASR

See **Oracle Database Advanced Replication**.

asymmetric algorithm

A **cryptographic algorithm** that uses different **key**s for **encryption** and **decryption**.

See also: **public key cryptography**.

asymmetric cryptography

See **public key cryptography**.

attribute

Directory attributes hold a specific data element such as a name, phone number, or job title. Each directory **entry** is comprised of a set of attributes, each of which belongs to an **object class**. Moreover, each attribute has both a *type*, which describes the kind of information in the attribute, and a *value*, which contains the actual data.

attribute configuration file

In an Oracle Directory Integration and Provisioning environment, a file that specifies attributes of interest in a connected directory.

attribute type

Attribute types specify information about a data element, such as the data type, maximum length, and whether it is single-valued or multivalued. The attribute type provides the real-world meaning for a value, and specifies the rules for creating and storing specific pieces of data, such as a name or an e-mail address.

attribute uniqueness

An Oracle Internet Directory feature that ensures that no two specified **attribute**s have the same value. It enables applications synchronizing with the enterprise directory to use attributes as unique keys.

attribute value

Attribute values are the actual data contained within an **attribute** for a particular **entry**. For example, for the attribute type `email`, an attribute value might be `sally.jones@oracle.com`.

authentication

The process of verifying the identity claimed by an entity based on its credentials. Authentication of a user is generally based on something the user knows or has (for example, a password or a certificate).

Authentication of an electronic message involves the use of some kind of system (such as **public key cryptography**) to ensure that a file or message which claims to originate from a given individual or company actually does, and a check based on the contents of a message to ensure that it was not modified in transit.

authentication level

An OracleAS Single Sign-On parameter that enables you to specify a particular authentication behavior for an application. You can link this parameter with a specific **authentication plugin**.

authentication plugin

An implementation of a specific authentication method. OracleAS Single Sign-On has Java plugins for password authentication, digital certificates, Windows native authentication, and third-party access management.

authorization

The process of granting or denying access to a service or network resource. Most security systems are based on a two step process. The first stage is authentication, in which a user proves his or her identity. The second stage is authorization, in which a user is allowed to access various resources based on his or her identity and the defined **authorization policy**.

authorization policy

Authorization policy describes how access to a protected resource is governed. Policy maps identities and objects to collections of rights according to some system model. For example, a particular authorization policy might state that users can access a sales report only if they belong to the sales group.

basic authentication

An **authentication** protocol supported by most browsers in which a Web server authenticates an entity with an encoded user name and password passed via data transmissions. Basic authentication is sometimes called plaintext authentication because the base-64 encoding can be decoded by anyone with a freely available decoding utility. Note that encoding is not the same as **encryption**.

Basic Encoding Rules (BER)

Basic Encoding Rules (BER) are the standard rules for encoding data units set forth in **ASN.1**. BER is sometimes incorrectly paired with ASN.1, which applies only to the abstract syntax description language, not the encoding technique.

BER

See **Basic Encoding Rules (BER)**.

binding

In networking, binding is the establishment of a logical connection between communicating entities.

In the case of Oracle Internet Directory, binding refers to the process of authenticating to the directory.

The formal set of rules for carrying a **SOAP** message within or on top of another protocol (underlying protocol) for the purpose of exchange is also called a binding.

block cipher

Block ciphers are a type of **symmetric algorithm**. A block cipher encrypts a message by breaking it down into fixed-size blocks (often 64 bits) and encrypting each block with a key. Some well known block ciphers include **Blowfish**, **DES**, and **AES**.

See also: **stream cipher**.

Blowfish

Blowfish is a **symmetric cryptography** algorithm developed by Bruce Schneier in 1993 as a faster replacement for **DES**. It is a **block cipher** using 64-bit blocks and keys of up to 448 bits.

CA

See **Certificate Authority (CA)**.

CA certificate

A **Certificate Authority (CA)** signs all certificates that it issues with its **private key**. The corresponding Certificate Authority's **public key** is itself contained within a certificate, called a CA Certificate (also referred to as a root certificate). A browser must contain the CA Certificate in its list of trusted root certificates in order to trust messages signed by the CA's private key.

cache

Generally refers to an amount of quickly accessible memory in your computer. However, on the Web it more commonly refers to where the browser stores downloaded files and graphics on the user's computer.

CBC

See **cipher block chaining (CBC)**.

central directory

In an Oracle Directory Integration and Provisioning environment, the directory that acts as the central repository. In an Oracle Directory Integration and Provisioning environment, Oracle Internet Directory is the central directory.

certificate

A certificate is a specially formatted data structure that associates a **public key** with the identity of its owner. A certificate is issued by a **Certificate Authority (CA)**. It contains the name, serial number, expiration dates, and public key of a particular entity. The certificate is digitally signed by the issuing CA so that a recipient can verify that the certificate is real. Most digital certificates conform to the **X.509** standard.

Certificate Authority (CA)

A Certificate Authority (CA) is a trusted third party that issues, renews, and revokes digital **certificate**s. The CA essentially vouches for a entity's identity, and may delegate the verification of an applicant to a **Registration Authority (RA)**. Some well known Certificate Authorities (CAs) include Digital Signature Trust, Thawte, and VeriSign.

certificate chain

An ordered list of certificates containing one or more pairs of a user **certificate** and its associated **CA certificate**.

certificate management protocol (CMP)

Certificate Management Protocol (CMP) handles all relevant aspects of certificate creation and management. CMP supports interactions between **public key infrastructure (PKI)**) components, such as the **Certificate Authority (CA)**, **Registration Authority (RA)**, and the user or application that is issued a certificate.

certificate request message format (CRMF)

Certificate Request Message Format (CRMF) is a format used for messages related to the life-cycle management of **X.509** certificates, as described in the **RFC** 2511 specification.

certificate revocation list (CRL)

A Certificate Revocation List (CRL) is a list of digital **certificate**s which have been revoked by the **Certificate Authority (CA)** that issued them.

change logs

A database that records changes made to a directory server.

cipher

See **cryptographic algorithm**.

cipher block chaining (CBC)

Cipher block chaining (CBC) is a mode of operation for a **block cipher**. CBC uses what is known as an initialization vector (IV) of a certain length. One of its key characteristics is that it uses a chaining mechanism that causes the decryption of a block of ciphertext to depend on all the preceding ciphertext blocks. As a result, the entire validity of all preceding blocks is contained in the immediately previous ciphertext block.

cipher suite

In **Secure Sockets Layer (SSL)**, a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

ciphertext

Ciphertext is the result of applying a **cryptographic algorithm** to readable data (plaintext) in order to render the data unreadable by all entities except those in possession of the appropriate **key**.

circle of trust

A circle of trust is a **federation** of **service provider**s and **identity provider**s that have business relationships based on **Liberty Alliance** architecture and operational agreements, and with whom users can transact business in a secure and apparently seamless environment.

claim

A claim is a declaration made by an entity (for example, a name, identity, key, group, and so on).

client SSL certificates

A type of **certificate** used to identify a client machine to a server through **Secure Sockets Layer (SSL)** (client authentication).

cluster

A collection of interconnected usable whole computers that is used as a single computing resource. Hardware clusters provide high availability and scalability.

CMP

See **certificate management protocol (CMP)**.

CMS

See **Cryptographic Message Syntax (CMS)**.

code signing certificates

A type of **certificate** used to identify the entity who signed a Java program, Java Script, or other signed file.

cold backup

In Oracle Internet Directory, this refers to the procedure of adding a new **directory system agent (DSA)** node to an existing replicating system by using the database copy procedure.

concurrency

The ability to handle multiple requests simultaneously. Threads and processes are examples of concurrency mechanisms.

concurrent clients

The total number of clients that have established a session with Oracle Internet Directory.

concurrent operations

The number of operations that are being executed on Oracle Internet Directory from all of the **concurrent clients**. Note that this is not necessarily the same as the concurrent clients, because some of the clients may be keeping their sessions idle.

confidentiality

In cryptography, confidentiality (also known as privacy) is the ability to prevent unauthorized entities from reading data. This is typically achieved through **encryption**.

configset

See **configuration set entry**.

configuration set entry

An Oracle Internet Directory entry holding the configuration parameters for a specific instance of the directory server. Multiple configuration set entries can be stored and referenced at runtime. The configuration set entries are maintained in the subtree specified by the `subConfigsubEntry` attribute of the **directory-specific entry (DSE)**, which itself resides in the associated **directory information base (DIB)** against which the servers are started.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its service name for the Oracle Database or its Oracle System Identifier (SID) for Oracle release 8.0 or version 7 databases. The network route provides, at a minimum, the location of the listener through use of a network address.

connected directory

In an Oracle Directory Integration and Provisioning environment, an information repository requiring full synchronization of data between Oracle Application Server Certificate Authority and itself—for example, an Oracle human resources database.

consumer

A directory server that is the destination of replication updates. Sometimes called a slave.

contention

Competition for resources.

context prefix

The **distinguished name (DN)** of the root of a **naming context**.

CRL

See **certificate revocation list (CRL)**.

CRMF

See **certificate request message format (CRMF)**.

cryptographic algorithm

A cryptographic algorithm is a defined sequence of processes to convert readable data (plaintext) to unreadable data (ciphertext) and vice versa. These conversions require some secret knowledge, normally contained in a **key**. Examples of cryptographic algorithms include **DES**, **AES**, **Blowfish**, and **RSA**.

Cryptographic Message Syntax (CMS)

Cryptographic Message Syntax (CMS) is a syntax defined in **RFC** 3369 for signing, digesting, authenticating, and encrypting digital messages.

cryptography

The process of protecting information by transforming it into an unreadable format. The information is encrypted using a **key**, which makes the data unreadable, and is then decrypted later when the information needs to be used again. See also **public key cryptography** and **symmetric cryptography**.

dads.conf

A configuration file for Oracle HTTP Server that is used to configure a **database access descriptor (DAD)**.

DAS

See **Oracle Delegated Administration Services**. (DAS).

Data Encryption Standard (DES)

Data Encryption Standard (DES) is a widely used **symmetric cryptography** algorithm developed in 1974 by IBM. It applies a 56-bit key to each 64-bit block of data. DES and 3DES are typically used as encryption algorithms by **S/MIME**.

data integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

See also: **integrity**.

database access descriptor (DAD)

Database connection information for a particular Oracle Application Server component, such as the OracleAS Single Sign-On schema.

decryption

The process of converting the contents of an encrypted message (ciphertext) back into its original readable format (plaintext).

default identity management realm

In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple other enterprises and stores information for them. In such hosted environments, the enterprise performing the hosting is called the default identity management realm, and the enterprises that are hosted are each associated with their own identity management realm in the **directory information tree (DIT)**.

default knowledge reference

A **knowledge reference** that is returned when the base object is not in the directory, and the operation is performed in a **naming context** not held locally by the server. A default knowledge reference typically sends the user to a server that has more knowledge about the directory partitioning arrangement.

default realm location

An attribute in the **root Oracle Context** that identifies the root of the **default identity management realm**.

Delegated Administration Services

See **Oracle Delegated Administration Services**.

delegated administrator

In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple other enterprises and stores information for them. In such an environment, a global administrator performs activities that span the entire directory. Other administrators—called delegated administrators—may exercise roles in specific identity management realms, or for specific applications.

DER

See **Distinguished Encoding Rules (DER)**.

DES

See **Data Encryption Standard (DES)**.

DIB

See **directory information base (DIB)**.

Diffie-Hellman

Diffie-Hellman (DH) is a public key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. First published in 1976, it was the first workable public key cryptographic system.

See also: **symmetric algorithm**.

digest

See **message digest**.

digital certificate

See **certificate**.

digital signature

A digital signature is the result of a two-step process applied to a given block of data. First, a **hash function** is applied to the data to obtain a result. Second, that result is

encrypted using the signer's **private key**. Digital signatures can be used to ensure integrity, message authentication, and non-repudiation of data. Examples of digital signature algorithms include **DSA**, **RSA**, and **ECDSA**.

Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) is an **asymmetric algorithm** that is used as part of the Digital Signature Standard (DSS). It cannot be used for encryption, only for digital signatures. The algorithm produces a pair of large numbers that enable the authentication of the signatory, and consequently, the integrity of the data attached. DSA is used both in generating and verifying digital signatures.

See also: **Elliptic Curve Digital Signature Algorithm (ECDSA)**.

directory

See **Oracle Internet Directory**, **Lightweight Directory Access Protocol (LDAP)**, and **X.500**.

directory information base (DIB)

The complete set of all information held in the directory. The DIB consists of entries that are related to each other hierarchically in a **directory information tree (DIT)**.

directory information tree (DIT)

A hierarchical tree-like structure consisting of the **DN**s of the entries.

directory integration and provisioning server

In an Oracle Directory Integration and Provisioning environment, the server that drives the synchronization of data between Oracle Internet Directory and a **connected directory**.

directory integration profile

In an Oracle Directory Integration and Provisioning environment, an entry in Oracle Internet Directory that describes how Oracle Directory Integration and Provisioning communicates with external systems and what is communicated.

Directory Manager

See **Oracle Directory Manager**.

directory naming context

See **naming context**.

directory provisioning profile

A special kind of **directory integration profile** that describes the nature of provisioning-related notifications that Oracle Directory Integration and Provisioning sends to the directory-enabled applications.

directory replication group (DRG)

The directory servers participating in a **replication agreement**.

directory server instance

A discrete invocation of a directory server. Different invocations of a directory server, each started with the same or different configuration set entries and startup flags, are said to be different directory server instances.

directory synchronization profile

A special kind of **directory integration profile** that describes how synchronization is carried out between Oracle Internet Directory and an external system.

directory system agent (DSA)

The **X.500** term for a directory server.

directory-specific entry (DSE)

An entry specific to a directory server. Different directory servers may hold the same **directory information tree (DIT)** name, but have different contents—that is, the contents can be specific to the directory holding it. A DSE is an entry with contents specific to the directory server holding it.

directory user agent (DUA)

The software that accesses a directory service on behalf of the directory user. The directory user may be a person or another software element.

DIS

See **directory integration and provisioning server**.

Distinguished Encoding Rules (DER)

Distinguished Encoding Rules (DER) are a set of rules for encoding **ASN.1** objects in byte-sequences. DER is a special case of **Basic Encoding Rules (BER)**.

distinguished name (DN)

A **X.500** distinguished name (DN) is a unique name for a node in a directory tree. A DN is used to provide a unique name for a person or any other directory entry. A DN is a concatenation of selected **attribute**s from each node in the tree along the path from the root node to the named entry's node. For example, in LDAP notation, the DN for a person named John Smith working at Oracle's US office would be: "cn=John Smith, ou=People, o=Oracle, c=us".

DIT

See **directory information tree (DIT)**.

DN

See **distinguished name (DN)**.

Document Type Definition (DTD)

A Document Type Definition (DTD) is a document that specifies constraints on the tags and tag sequences that are valid for a given **XML** document. DTDs follow the rules of Simple Generalized Markup Language (SGML), the parent language of XML.

domain component attribute

The domain component (dc) attribute can be used in constructing a **distinguished name (DN)** from a domain name. For example, using a domain name such as "oracle.com", one could construct a DN beginning with "dc=oracle, dc=com", and then use this DN as the root of its subtree of directory information.

DRG

See **directory replication group (DRG)**.

DSA

See **Digital Signature Algorithm (DSA)** or **directory system agent (DSA)**.

DSE

See **directory-specific entry (DSE)**.

DTD

See **Document Type Definition (DTD)**.

ECC

See **Elliptic Curve Cryptography (ECC)**.

ECDSA

See **Elliptic Curve Digital Signature Algorithm (ECDSA)**.

EJB

See **Enterprise Java Bean (EJB)**.

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is an alternative to the **RSA** encryption system which is based on the difficulty of solving elliptic curve discrete logarithm problems rather than on factoring large numbers. Developed and marketed by Certicom, ECC is especially suitable for environments, such as wireless devices and PC cards, where computational power is limited and high speed is required. For any given key size (measured in bits) ECC provides more security (is harder to decrypt without the key) than RSA.

Elliptic Curve Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analog of the **Digital Signature Algorithm (DSA)** standard. The advantages of ECDSA compared to RSA-like schemes are shorter key lengths and faster signing and decryption. For example, a 160 (210) bit ECC key is expected to give the same security as a 1024 (2048) bit RSA key, and the advantage increases as level of security is raised.

encryption

Encryption is the process of converting plaintext to ciphertext by applying a **cryptographic algorithm**.

encryption certificate

An encryption certificate is a **certificate** containing a **public key** that is used to encrypt electronic messages, files, documents, or data transmission, or to establish or exchange a session key for these same purposes.

end-to-end security

This is a property of message-level security that is established when a message traverses multiple applications within and between business entities and is secure over its full route through and between the business entities.

Enterprise Java Bean (EJB)

Enterprise JavaBeans (EJBs) are a Java API developed by Sun Microsystems that defines a component architecture for multi-tier client/server systems. Because EJB systems are written in Java, they are platform independent. Being object oriented, they

can be implemented into existing systems with little or no recompiling and configuring.

Enterprise Manager

See **Oracle Enterprise Manager**.

entry

An entry is a unique record in a directory that describes an object, such as a person. An entry consists of **attribute**s and their associated **attribute value**s, as dictated by the **object class** that describes that entry object. All entries in an LDAP directory structure are uniquely identified through their **distinguished name (DN)**.

export agent

In an Oracle Directory Integration and Provisioning environment, an agent that exports data out of Oracle Internet Directory.

export data file

In an Oracle Directory Integration and Provisioning environment, the file that contains data exported by an **export agent**.

export file

See **export data file**.

external agent

A directory integration agent that is independent of Oracle Directory Integration and Provisioning server. Oracle Directory Integration and Provisioning server does not provide scheduling, mapping, or error handling services for it. An external agent is typically used when a third party metadirectory solution is integrated with Oracle Directory Integration and Provisioning.

external application

Applications that do not delegate authentication to the OracleAS Single Sign-On server. Instead, they display HTML login forms that ask for application user names and passwords. At the first login, users can choose to have the OracleAS Single Sign-On server retrieve these credentials for them. Thereafter, they are logged in to these applications transparently.

failover

The process of failure recognition and recovery. In an Oracle Application Server Cold Failover Cluster (Identity Management), an application running on one cluster node is transparently migrated to another cluster node. During this migration, clients accessing the service on the cluster see a momentary outage and may need to reconnect once the failover is complete.

fan-out replication

Also called a point-to-point replication, a type of replication in which a supplier replicates directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

Federal Information Processing Standards (FIPS)

Federal Information Processing Standards (FIPS) are standards for information processing issued by the US government Department of Commerce's National Institute of Standards and Technology (NIST).

federated identity management (FIM)

The agreements, standards, and technologies that make identity and entitlements portable across autonomous domains. FIM makes it possible for an authenticated user to be recognized and take part in personalized services across multiple domains. It avoids pitfalls of centralized storage of personal information, while allowing users to link identity information between different accounts. Federated identity requires two key components: trust and standards. The trust model of federated identity management is based on **circle of trust**. The standards are defined by the **Liberty Alliance** Project.

federation

A federation is a group of entities (companies and organizations) that have a shared user base, and have agreed to provide identity and authorization tokens so that their users only have to logon once to access all of the services in their **circle of trust**. Within the federation, at least one entity serves as the **identity provider** who is responsible for authenticating users. Entities that provide services to the user are referred to as **service provider**s.

filter

A filter is an expression that defines the entries to be returned from a request or search on a directory. Filters are typically expressed as DNs, for example: `cn=susie smith,o=acme,c=us`.

FIM

See **federated identity management (FIM)**.

FIPS

See **Federal Information Processing Standards (FIPS)**.

forced authentication

The act of forcing a user to reauthenticate if he or she has been idle for a preconfigured amount of time. Oracle Application Server Single Sign-On enables you to specify a global user inactivity timeout. This feature is intended for installations that have sensitive applications.

GET

An authentication method whereby login credentials are submitted as part of the login URL.

global administrator

In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple other enterprises and stores information for them. In such an environment, a global administrator performs activities that span the entire directory.

global unique identifier (GUID)

An identifier generated by the system and inserted into an entry when the entry is added to the directory. In a multimaster replicated environment, the GUID, not the DN, uniquely identifies an entry. The GUID of an entry cannot be modified by a user.

global user inactivity timeout

An optional feature of Oracle Application Server Single Sign-On that forces users to reauthenticate if they have been idle for a preconfigured amount of time. The global user inactivity timeout is much shorter than the single sign-out session timeout.

globalization support

Multilanguage support for graphical user interfaces. Oracle Application Server Single Sign-On supports 29 languages.

globally unique user ID

A numeric string that uniquely identifies a user. A person may change or add user names, passwords, and distinguished names, but her globally unique user ID always remains the same.

grace login

A login occurring within the specified period before password expiration.

group search base

In the Oracle Internet Directory default **directory information tree (DIT)**, the node in the identity management realm under which all the groups can be found.

guest user

One who is not an anonymous user, and, at the same time, does not have a specific user entry.

GUID

See **global unique identifier (GUID)**.

handshake

A protocol two computers use to initiate a communication session.

hash

A number generated from a string of text with an algorithm. The hash value is substantially smaller than the text itself. Hash numbers are used for security and for faster access to data.

See also: **hash function**.

hash function

In cryptography, a hash function or one-way hash function is an algorithm that produces a given value when applied to a given block of data. The result of a hash function can be used to ensure the integrity of a given block of data. For a hash function to be considered secure, it must be very difficult, given a known data block and a known result, to produce another data block that produces the same result.

Hashed Message Authentication Code (HMAC)

Hashed Message Authentication Code (HMAC) is a hash function technique used to create a secret hash function output. This strengthens existing hash functions such as MD5 and SHA. It is used in transport layer security (TLS).

HMAC

See **Hashed Message Authentication Code (HMAC)**.

HTTP

The Hyper Text Transfer Protocol (HTTP) is the protocol used between a Web browser and a server to request a document and transfer its contents. The specification is maintained and developed by the World Wide Web Consortium.

HTTP Server

See **Oracle HTTP Server**.

httpd.conf

The file used to configure **Oracle HTTP Server**.

iASAdmins

The administrative group responsible for user and group management functions in Oracle Application Server. The OracleAS Single Sign-On administrator is a member of the group iASAdmins.

identity management

The process by which the complete security lifecycle for network entities is managed in an organization. It typically refers to the management of an organization's application users, where steps in the security life cycle include account creation, suspension, privilege modification, and account deletion. The network entities managed may also include devices, processes, applications, or anything else that needs to interact in a networked environment. Entities managed by an identity management process may also include users outside of the organization, for example customers, trading partners, or Web services.

identity management infrastructure database

The database that contains data for OracleAS Single Sign-On and Oracle Internet Directory.

identity management realm

A collection of identities, all of which are governed by the same administrative policies. In an enterprise, all employees having access to the intranet may belong to one realm, while all external users who access the public applications of the enterprise may belong to another realm. An identity management realm is represented in the directory by a specific **entry** with a special **object class** associated with it.

identity management realm-specific Oracle Context

An Oracle Context contained in each identity management realm. It stores the following information:

- User naming policy of the identity management realm—that is, how users are named and located.
- Mandatory authentication attributes.
- Location of groups in the identity management realm.
- Privilege assignments for the identity management realm—for example: who has privileges to add more users to the realm.
- Application specific data for that realm including authorizations.

identity provider

These are organizations recognized by the members of a **circle of trust** as the entity responsible for authenticating users and providing the digital identity information of

users to other parties in a **federation**. Identity providers enter into partnerships with service providers and provide services that follow agreed-upon practices set by all parties in a federation.

import agent

In an Oracle Directory Integration and Provisioning environment, an agent that imports data into Oracle Internet Directory.

import data file

In an Oracle Directory Integration and Provisioning environment, the file containing the data imported by an **import agent**.

infrastructure tier

The Oracle Application Server components responsible for identity management. These components are OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Internet Directory.

inherit

When an **object class** has been derived from another class, it also derives, or inherits, many of the characteristics of that other class. Similarly, an attribute subtype inherits the characteristics of its supertype.

instance

See **directory server instance**.

integrity

In cryptography, integrity is the ability to detect if data has been modified by entities that are not authorized to modify it.

Internet Directory

See **Oracle Internet Directory**.

Internet Engineering Task Force (IETF)

The principal body engaged in the development of new Internet standard specifications. It is an international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Internet Message Access Protocol (IMAP)

A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders, also called mailboxes, in a way that is functionally equivalent to local mailboxes.

J2EE

See **Java 2 Platform, Enterprise Edition (J2EE)**.

Java 2 Platform, Enterprise Edition (J2EE)

Java 2 Platform, Enterprise Edition (J2EE) is an environment for developing and deploying enterprise applications, defined by Sun Microsystems Inc. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multitiered, Web-based applications.

Java Server Page (JSP)

JavaServer Pages (JSP), a server-side technology, are an extension to the Java servlet technology that was developed by Sun Microsystems. JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements (the design and display of the page). Embedded in the HTML page, the Java source code and its extensions help make the HTML more functional, being used in dynamic database queries, for example.

JSP

See **Java Server Page (JSP)**.

key

A key is a data structure that contains some secret knowledge necessary to successfully encrypt or decrypt a given block of data. The larger the key, the harder it is to crack a block of encrypted data. For example, a 256-bit key is more secure than a 128-bit key.

key pair

A **public key** and its associated **private key**.

See also: **public/private key pair**.

knowledge reference

The access information (name and address) for a remote **directory system agent (DSA)** and the name of the **directory information tree (DIT)** subtree that the remote DSA holds. Knowledge references are also called referrals.

latency

The time a client has to wait for a given directory operation to complete. Latency can be defined as wasted time. In networking discussions, latency is defined as the travel time of a packet from source to destination.

LDAP

See **Lightweight Directory Access Protocol (LDAP)**.

LDAP connection cache

To improve throughput, the OracleAS Single Sign-On server caches and then reuses connections to Oracle Internet Directory.

LDAP Data Interchange Format (LDIF)

A common, text-based format for exchanging directory data between systems. The set of standards for formatting an input file for any of the LDAP command-line utilities.

LDIF

See **LDAP Data Interchange Format (LDIF)**.

legacy application

Older application that cannot be modified to delegate authentication to the OracleAS Single Sign-On server. Also known as an **external application**.

Liberty Alliance

The Liberty Alliance Project is an alliance of more than 150 companies, non-profit, and government organizations from around the globe. The consortium is committed to developing an open standard for federated network identity that supports all current

and emerging network devices. The Liberty Alliance is the only global body working to define and drive open technology standards, privacy, and business guidelines for **federated identity management (FIM)**.

Lightweight Directory Access Protocol (LDAP)

A set of protocols for accessing information in directories. LDAP supports TCP/IP, which is necessary for any type of Internet access. Its framework of design conventions supports industry-standard directory products, such as Oracle Internet Directory. Because it is a simpler version of the **X.500** standard, LDAP is sometimes called X.500 light.

load balancer

Hardware devices and software that balance connection requests between two or more servers, either due to heavy load or failover. BigIP, Alteon, or Local Director are all popular hardware devices. Oracle Application Server Web Cache is an example of load balancing software.

logical host

In an Oracle Application Server Cold Failover Cluster (Identity Management), one or more disk groups and pairs of host names and IP addresses. It is mapped to a physical host in the cluster. This physical host impersonates the host name and IP address of the logical host.

MAC

See **message authentication code (MAC)**.

man-in-the-middle

A security attack characterized by the third-party, surreptitious interception of a message. The third-party, the *man-in-the-middle*, decrypts the message, re-encrypts it (with or without alteration of the original message), and retransmits it to the originally-intended recipient—all without the knowledge of the legitimate sender and receiver. This type of security attack works only in the absence of **authentication**.

mapping rules file

In an Oracle Directory Integration and Provisioning environment, the file that specifies mappings between Oracle Internet Directory attributes and those in a **connected directory**.

master definition site (MDS)

In replication, a master definition site is the Oracle Internet Directory database from which the administrator runs the configuration scripts.

master site

In replication, a master site is any site other than the **master definition site (MDS)** that participates in LDAP replication.

matching rule

In a search or compare operation, determines equality between the attribute value sought and the attribute value stored. For example, matching rules associated with the `telephoneNumber` attribute could cause "(650) 123-4567" to be matched with either "(650) 123-4567" or "6501234567" or both. When you create an **attribute**, you associate a matching rule with it.

MD2

Message Digest Two (MD2) is a message digest **hash function**. The algorithm processes input text and creates a 128-bit **message digest** which is unique to the message and can be used to verify data integrity. MD2 was developed by Ron Rivest for RSA Security and is intended to be used in systems with limited memory, such as smart cards.

MD4

Message Digest Four (MD4) is similar to **MD2** but designed specifically for fast processing in software.

MD5

Message Digest Five (MD5) is a message digest **hash function**. The algorithm processes input text and creates a 128-bit **message digest** which is unique to the message and can be used to verify data integrity. MD5 was developed by Ron Rivest after potential weaknesses were reported in **MD4**. MD5 is similar to MD4 but slower because more manipulation is made to the original data.

MDS

See **master definition site (MDS)**.

message authentication

The process of verifying that a particular message came from a particular entity.

See also: **authentication**.

message authentication code (MAC)

The Message Authentication Code (MAC) is a result of a two-step process applied to a given block of data. First, the result of a **hash function** is obtained. Second, that result is encrypted using a **secret key**. The MAC can be used to authenticate the source of a given block of data.

message digest

The result of a **hash function**.

See also: **hash**.

metadirectory

A directory solution that shares information between all enterprise directories, integrating them into one virtual directory. It centralizes administration, thereby reducing administrative costs. It synchronizes data between directories, thereby ensuring that it is consistent and up-to-date across the enterprise.

middle tier

That portion of a OracleAS Single Sign-On instance that consists of the Oracle HTTP Server and OC4J. The OracleAS Single Sign-On middle tier is situated between the identity management infrastructure database and the client.

mod_osso

A module on the Oracle HTTP Server that enables applications protected by OracleAS Single Sign-On to accept HTTP headers in lieu of a user name and password once the user has logged into the OracleAS Single Sign-On server. The values for these headers are stored in the **mod_osso cookie**.

mod_osso cookie

User data stored on the HTTP server. The cookie is created when a user authenticates. When the same user requests another application, the Web server uses the information in the mod_osso cookie to log the user in to the application. This feature speeds server response time.

mod_proxy

A module on the Oracle HTTP Server that makes it possible to use **mod_osso** to enable single sign-on to legacy, or **external application**s.

MTS

See **shared server**.

multimaster replication

Also called peer-to-peer or *n*-way replication, a type of replication that enables multiple sites, acting as equals, to manage groups of replicated data. In a multimaster replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

naming attribute

The attribute used to compose the RDN of a new user entry created through Oracle Delegated Administration Services or Oracle Internet Directory Java APIs. The default value for this is `cn`.

naming context

A subtree that resides entirely on one server. It must be contiguous, that is, it must begin at an entry that serves as the top of the subtree, and extend downward to either leaf entries or **knowledge reference**s (also called referrals) to subordinate naming contexts. It can range in size from a single entry to the entire **directory information tree (DIT)**.

native agent

In an Oracle Directory Integration and Provisioning environment, an agent that runs under the control of the **directory integration and provisioning server**. It is in contrast to an **external agent**.

net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the service to which they wish to connect, for example:

```
CONNECT username/password@net_service_name
```

Depending on your needs, net service names can be stored in a variety of places, including:

- Local configuration file, `tnsnames.ora`, on each client
- Directory server
- Oracle Names server
- External naming service, such as NDS, NIS or CDS

Net Services

See **Oracle Net Services**.

nickname attribute

The attribute used to uniquely identify a user in the entire directory. The default value for this is `uid`. Applications use this to resolve a simple user name to the complete distinguished name. The user nickname attribute cannot be multi-valued—that is, a given user cannot have multiple nicknames stored under the same attribute name.

non-repudiation

In cryptography, the ability to prove that a given **digital signature** was produced with a given entity's **private key**, and that a message was sent untampered at a given point in time.

OASIS

Organization for the Advancement of Structured Information Standards. OASIS is a worldwide not-for-profit consortium that drives the development, convergence and adoption of e-business standards.

object class

In LDAP, object classes are used to group information. Typically an object class models a real-world object such as a person or a server. Each directory entry belongs to one or more object classes. The object class determines the attributes that make up an entry. One object class can be derived from another, thereby inheriting some of the characteristics of the other class.

OC4J

See **Oracle Containers for J2EE (OC4J)**.

OCA

See **Oracle Certificate Authority**.

OCI

See **Oracle Call Interface (OCI)**.

OCSP

See **Online Certificate Status Protocol (OCSP)**.

OEM

See **Oracle Enterprise Manager**.

OID

See **Oracle Internet Directory**.

OID Control Utility

A command-line tool for issuing run-server and stop-server commands. The commands are interpreted and executed by the **OID Monitor** process.

OID Database Password Utility

The utility used to change the password with which Oracle Internet Directory connects to an Oracle Database.

OID Monitor

The Oracle Internet Directory component that initiates, monitors, and terminates the Oracle Internet Directory Server processes. It also controls the replication server if one is installed, and Oracle Directory Integration and Provisioning Server.

Online Certificate Status Protocol (OCSP)

Online Certificate Status Protocol (OCSP) is one of two common schemes for checking the validity of digital certificates. The other, older method, which OCSP has superseded in some scenarios, is **certificate revocation list (CRL)**. OCSP is specified in **RFC** 2560.

one-way function

A function that is easy to compute in one direction but quite difficult to reverse compute, that is, to compute in the opposite direction.

one-way hash function

A **one-way function** that takes a variable sized input and creates a fixed size output.

See also: **hash function**.

Oracle Application Server Single Sign-On

OracleAS Single Sign-On consists of program logic that enables you to log in securely to applications such as expense reports, mail, and benefits. These applications take two forms: **partner application**s and **external application**s. In both cases, you gain access to several applications by authenticating only once.

Oracle Call Interface (OCI)

An application programming interface (API) that enables you to create applications that use the native procedures or function calls of a third-generation language to access an Oracle Database server and control all phases of SQL statement execution.

Oracle Certificate Authority

Oracle Application Server Certificate Authority is a **Certificate Authority (CA)** for use within your Oracle Application Server environment. OracleAS Certificate Authority uses Oracle Internet Directory as the storage repository for certificates. OracleAS Certificate Authority integration with OracleAS Single Sign-On and Oracle Internet Directory provides seamless certificate provisioning mechanisms for applications relying on them. A user provisioned in Oracle Internet Directory and authenticated in OracleAS Single Sign-On can choose to request a digital certificate from OracleAS Certificate Authority.

Oracle CMS

Oracle CMS implements the IETF **Cryptographic Message Syntax (CMS)** protocol. CMS defines data protection schemes that allow for secure message envelopes.

Oracle Containers for J2EE (OC4J)

A lightweight, scalable container for **Java 2 Platform, Enterprise Edition (J2EE)**.

Oracle Context

See **identity management realm-specific Oracle Context** and **root Oracle Context**.

Oracle Crypto

Oracle Crypto is a pure Java library that provides core cryptography algorithms.

Oracle Database Advanced Replication

A feature in the Oracle Database that enables database tables to be kept synchronized across two Oracle databases.

Oracle Delegated Administration Services

A set of individual, pre-defined services—called Oracle Delegated Administration Services units—for performing directory operations on behalf of a user. Oracle Internet Directory Self-Service Console makes it easier to develop and deploy administration solutions for both Oracle and third-party applications that use Oracle Internet Directory.

Oracle Directory Integration and Provisioning

A collection of interfaces and services for integrating multiple directories by using Oracle Internet Directory and several associated plug-ins and connectors. A feature of Oracle Internet Directory that enables an enterprise to use an external user repository to authenticate to Oracle products.

Oracle Directory Integration and Provisioning Server

In an Oracle Directory Integration and Provisioning environment, a daemon process that monitors Oracle Internet Directory for change events and takes action based on the information present in the **directory integration profile**.

Oracle Directory Integration Platform

A component of **Oracle Internet Directory**. It is a framework developed to integrate applications around a central LDAP directory like Oracle Internet Directory.

Oracle Directory Manager

A Java-based tool with a graphical user interface for administering Oracle Internet Directory.

Oracle Enterprise Manager

A separate Oracle product that combines a graphical console, agents, common services, and tools to provide an integrated and comprehensive systems management platform for managing Oracle products.

Oracle HTTP Server

Software that processes Web transactions that use the Hypertext Transfer Protocol (HTTP). Oracle uses HTTP software developed by the Apache Group.

Oracle Identity Management

An infrastructure enabling deployments to manage centrally and securely all enterprise identities and their access to various applications in the enterprise.

Oracle Internet Directory

A general purpose directory service that enables retrieval of information about dispersed users and network resources. It combines **Lightweight Directory Access Protocol (LDAP)** Version 3 with the high performance, scalability, robustness, and availability of the Oracle Database.

Oracle Liberty SDK

Oracle Liberty SDK implements the **Liberty Alliance** Project specifications enabling federated single sign-on between third-party Liberty-compliant applications.

Oracle Net Services

The foundation of the Oracle family of networking products, allowing services and their client applications to reside on different computers and communicate. The main function of Oracle Net Services is to establish network sessions and transfer data

between a client application and a server. Oracle Net Services is located on each computer in the network. Once a network session is established, Oracle Net Services acts as a data courier for the client and the server.

Oracle PKI certificate usages

Defines Oracle application types that a **certificate** supports.

Oracle PKI SDK

Oracle PKI SDK implements the security protocols that are necessary within **public key infrastructure (PKI)** implementations.

Oracle SAML

Oracle SAML provides a framework for the exchange of security credentials among disparate systems and applications in an XML-based format as outlined in the **OASIS** specification for the **Security Assertions Markup Language (SAML)**.

Oracle Security Engine

Oracle Security Engine extends Oracle Crypto by offering X.509 based certificate management functions. Oracle Security Engine is a superset of Oracle Crypto.

Oracle S/MIME

Oracle S/MIME implements the **Secure/Multipurpose Internet Mail Extension (S/MIME)** specifications from the **Internet Engineering Task Force (IETF)** for secure e-mail.

Oracle Wallet Manager

A Java-based application that security administrators use to manage public-key security credentials on clients and servers.

See also: *Oracle Advanced Security Administrator's Guide*.

Oracle Web Services Security

Oracle Web Services Security provides a framework for authentication and authorization using existing security technologies as outlined in the **OASIS** specification for Web Services Security.

Oracle XML Security

Oracle XML Security implements the W3C specifications for XML Encryption and XML Signature.

OracleAS Portal

An OracleAS Single Sign-On **partner application** that provides a mechanism for integrating files, images, applications, and Web sites. The External Applications portlet provides access to external applications.

other information repository

In an Oracle Directory Integration and Provisioning environment, in which Oracle Internet Directory serves as the **central directory**, any information repository except Oracle Internet Directory.

OWM

See **Oracle Wallet Manager**.

partition

A unique, non-overlapping directory naming context that is stored on one directory server.

partner application

An Oracle Application Server application or non-Oracle application that delegates the authentication function to the OracleAS Single Sign-On server. This type of application spares users from reauthenticating by accepting **mod_osso** headers.

peer-to-peer replication

Also called multimaster replication or *n*-way replication. A type of replication that enables multiple sites, acting as equals, to manage groups of replicated data. In such a replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

PKCS#1

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS#1 provides recommendations for the implementation of public-key cryptography based on the RSA algorithm, covering the following aspects: cryptographic primitives; encryption schemes; signature schemes; ASN.1 syntax for representing keys and for identifying the schemes.

PKCS#5

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS#5 provides recommendations for the implementation of password-based cryptography.

PKCS#7

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #7 describes general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes.

PKCS#8

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #8 describes syntax for private key information, including a private key for some public key algorithms and a set of attributes. The standard also describes syntax for encrypted private keys.

PKCS#10

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #10 describes syntax for a request for certification of a public key, a name, and possibly a set of attributes.

PKCS#12

The Public Key Cryptography Standards (PKCS) are specifications produced by RSA Laboratories. PKCS #12 describes a transfer syntax for personal identity information, including private keys, certificates, miscellaneous secrets, and extensions. Systems (such as browsers or operating systems) that support this standard allow a user to import, export, and exercise a single set of personal identity information—typically in a format called a **wallet**.

PKI

See **public key infrastructure (PKI)**.

plaintext

Plaintext is readable data prior to a transformation to ciphertext using encryption, or readable data that is the result of a transformation from ciphertext using decryption.

point-to-point replication

Also called fan-out replication is a type of replication in which a supplier replicates directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

policy precedence

In Oracle Application Server Certificate Authority (OCA), policies are applied to incoming requests in the order that they are displayed on the main policy page. When the OCA policy processor module parses policies, those that appear toward the top of the policy list are applied to requests first. Those that appear toward the bottom of the list are applied last and take precedence over the others. Only enabled policies are applied to incoming requests.

policy.properties

A multipurpose configuration file for Oracle Application Server Single Sign-On that contains basic parameters required by the single sign-on server. Also used to configure advanced features of OracleAS Single Sign-On, such as multilevel authentication.

POSIX

Portable Operating System Interface for UNIX. A set of programming interface standards governing how to write application source code so that the applications are portable between operating systems. A series of standards being developed by the **Internet Engineering Task Force (IETF)**.

POST

An authentication method whereby login credentials are submitted within the body of the login form.

predicates

In Oracle Application Server Certificate Authority (OCA), a policy predicate is a logical expression that can be applied to a policy to limit how it is applied to incoming certificate requests or revocations. For example, the following predicate expression specifies that the policy in which it appears can have a different effect for requests or revocations from clients with DNs that include "ou=sales,o=acme,c=us":

```
Type=="client" AND DN=="ou=sales,o=acme,c=us"
```

primary node

In an Oracle Application Server Cold Failover Cluster (Identity Management), the cluster node on which the application runs at any given time.

See also: **secondary node**.

private key

A private key is the secret key in a **public/private key pair** used in **public key cryptography**. An entity uses its private key to decrypt data that has been encrypted with its **public key**. The entity can also use its private key to create **digital signature**s. The security of data encrypted with the entity's public key as well as signatures created by the private key depends on the private key remaining secret.

private key cryptography

See **symmetric cryptography**.

profile

See **directory integration profile**.

provisioned applications

Applications in an environment where user and group information is centralized in Oracle Internet Directory. These applications are typically interested in changes to that information in Oracle Internet Directory.

provisioning agent

An application or process that translates Oracle-specific provisioning events to external or third-party application-specific events.

proxy server

A server between a client application, such as a Web browser, and a real server. It intercepts all requests to the real server to see if it can fulfil the requests itself. If not, it forwards the request to the real server. In OracleAS Single Sign-On, proxies are used for load balancing and as an extra layer of security.

See also: **load balancer**.

proxy user

A kind of user typically employed in an environment with a middle tier such as a firewall. In such an environment, the end user authenticates to the middle tier. The middle tier then logs into the directory on the end user's behalf. A proxy user has the privilege to switch identities and, once it has logged into the directory, switches to the end user's identity. It then performs operations on the end user's behalf, using the authorization appropriate to that particular end user.

public key

A public key is the non-secret key in a **public/private key pair** used in **public key cryptography**. A public key allows entities to encrypt data that can only then be decrypted with the public key's owner using the corresponding **private key**. A public key can also be used to verify digital signatures created with the corresponding private key.

public key certificate

See **certificate**.

public key cryptography

Public key cryptography (also known as asymmetric cryptography) uses two keys, one public and the other private. These keys are called a key pair. The private key must be kept secret, while the public key can be transmitted to any party. The private key and the public key are mathematically related. A message that is signed by a private key can be verified by the corresponding public key. Similarly, a message encrypted by the public key can be decrypted by the private key. This method ensures privacy because only the owner of the private key can decrypt the message.

public key encryption

The process in which the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using the recipient's private key.

public key infrastructure (PKI)

A public key infrastructure (PKI) is a system that manages the issuing, distribution, and authentication of **public key**s and **private key**s. A PKI typically comprises the following components:

- A **Certificate Authority (CA)** that is responsible for generating, issuing, publishing and revoking digital certificates.
- A **Registration Authority (RA)** that is responsible for verifying the information supplied in requests for certificates made to the CA.
- A directory service where a **certificate** or **certificate revocation list (CRL)** gets published by the CA and where they can be retrieved by relying third parties.
- Relying third parties that use the certificates issued by the CA and the **public key**s contained therein to verify **digital signature**s and encrypt data.

public/private key pair

A mathematically related set of two numbers where one is called the private key and the other is called the public key. Public keys are typically made widely available, while private keys are available only to their owners. Data encrypted with a public key can only be decrypted with its associated private key and vice versa. Data encrypted with a public key cannot be decrypted with the same public key.

RC2

Rivest Cipher Two (RC2) is a 64-bit **block cipher** developed by Ronald Rivest for RSA Security, and was designed as a replacement for **Data Encryption Standard (DES)**.

RC4

Rivest Cipher Four (RC4) is a **stream cipher** developed by Ronald Rivest for RSA Security. RC4 allows variable key lengths up to 1024 bits. RC4 is most commonly used to secure data communications by encrypting traffic between Web sites that use the **Secure Sockets Layer (SSL)** protocol.

RDN

See **relative distinguished name (RDN)**.

readable data

Data prior to a transformation to ciphertext via encryption or data that is the result of a transformation from ciphertext via decryption.

realm

See **identity management realm**.

realm search base

An attribute in the **root Oracle Context** that identifies the entry in the **directory information tree (DIT)** that contains all **identity management realm**s. This attribute is used when mapping a simple realm name to the corresponding entry in the directory.

referral

Information that a directory server provides to a client and which points to other servers the client must contact to find the information it is requesting.

See also: **knowledge reference**.

Registration Authority (RA)

The Registration Authority (RA) is responsible for verifying and enrolling users before a certificate is issued by a **Certificate Authority (CA)**. The RA may assign each applicant a relative distinguished value or name for the new certificate applied. The RA does not sign or issue certificates.

registry entry

An entry containing runtime information associated with invocations of Oracle Internet Directory servers, called a **directory server instance**. Registry entries are stored in the directory itself, and remain there until the corresponding directory server instance stops.

relational database

A structured collection of data that stores data in tables consisting of one or more rows, each containing the same set of columns. Oracle makes it very easy to link the data in multiple tables. This is what makes Oracle a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the tables. The link is based on one or more fields common to both tables.

relative distinguished name (RDN)

The local, most granular level entry name. It has no other qualifying entry names that would serve to uniquely address the entry. In the example, `cn=Smith,o=acme,c=US`, the RDN is `cn=Smith`.

remote master site (RMS)

In a replicated environment, any site, other than the **master definition site (MDS)**, that participates in **Oracle Database Advanced Replication**.

replica

Each copy of a **naming context** that is contained within a single server.

replication agreement

A special directory entry that represents the replication relationship among the directory servers in a **directory replication group (DRG)**.

response time

The time between the submission of a request and the completion of the response.

RFC

The Internet Request For Comments (or RFC) documents are the written definitions of the protocols and policies of the Internet. The Internet Engineering Task Force (IETF) facilitates the discussion, development, and establishment of new standards. A standard is published using the RFC acronym and a reference number. For example, the official standard for e-mail is RFC 822.

root CA

In a hierarchical **public key infrastructure (PKI)**, the root **Certificate Authority (CA)** is the CA whose **public key** serves as the most trusted datum for a security domain.

root directory specific entry (DSE)

An entry storing operational information about the directory. The information is stored in a number of attributes.

root DSE

See **root directory specific entry (DSE)**.

root Oracle Context

In the Oracle Identity Management infrastructure, the root Oracle Context is an entry in Oracle Internet Directory containing a pointer to the default identity management realm in the infrastructure. It also contains information on how to locate an identity management realm given a simple name of the realm.

RSA

RSA is a **public key cryptography** algorithm named after its inventors (Rivest, Shamir, and Adelman). The RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browsers from Netscape and Microsoft, and many other products.

RSAES-OAEP

The RSA Encryption Scheme - Optimal Asymmetric Encryption Padding (RSAES-OAEP) is a public key encryption scheme combining the **RSA** algorithm with the OAEP method. Optimal Asymmetric Encryption Padding (OAEP) is a method for encoding messages developed by Mihir Bellare and Phil Rogaway.

S/MIME

See **Secure/Multipurpose Internet Mail Extension (S/MIME)**.

SAML

See **Security Assertions Markup Language (SAML)**.

SASL

See **Simple Authentication and Security Layer (SASL)**.

scalability

The ability of a system to provide throughput in proportion to, and limited only by, available hardware resources.

schema

The collection of **attribute**s, **object class**es, and their corresponding **matching rule**s.

secondary node

In an Oracle Application Server Cold Failover Cluster (Identity Management), the cluster node to which an application is moved during a failover.

See also: **primary node**.

secret key

A secret key is the **key** used in a **symmetric algorithm**. Since a secret key is used for both encryption and decryption, it must be shared between parties that are transmitting ciphertext to one another but must be kept secret from all unauthorized entities.

secret key cryptography

See **symmetric cryptography**.

Secure Hash Algorithm (SHA)

Secure Hash Algorithm (SHA) is a **hash function** algorithm that produces a 160-bit **message digest** based upon the input. The algorithm is used in the Digital Signature Standard (DSS). With the introduction of the Advanced Encryption Standard (AES) which offers three key sizes: 128, 192 and 256 bits, there has been a need for a companion hash algorithm with a similar level of security. The newer SHA-256, SHA-284 and SHA-512 hash algorithms comply with these enhanced requirements.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) is a protocol designed by Netscape Communications to enable encrypted, authenticated communications across networks (such as the Internet). SSL uses the **public key encryption** system from RSA, which also includes the use of a digital certificate. SSL provides three elements of secure communications: **confidentiality**, **authentication**, and **integrity**.

SSL has evolved into **Transport Layer Security (TLS)**. TLS and SSL are not interoperable. However, a message sent with TLS can be handled by a client that handles SSL.

Secure/Multipurpose Internet Mail Extension (S/MIME)

Secure/Multipurpose Internet Mail Extension (S/MIME) is an Internet Engineering Task Force (IETF) standard for securing MIME data through the use of **digital signature**s and **encryption**.

Security Assertions Markup Language (SAML)

Security Assertions Markup Language (SAML) is an **XML**-based framework for exchanging security information over the Internet. SAML enables the exchange of **authentication** and **authorization** information between various security services systems that otherwise would not be able to interoperate. The SAML 1.0 specification was adopted by **OASIS** in 2002.

server certificate

A **certificate** that attests to the identity of an organization that uses a secure Web server to serve data. A server certificate must be associated with a **public/private key pair** issued by a mutually trusted **Certificate Authority (CA)**. Server certificates are required for secure communications between a browser and a Web server.

service provider

These are organizations recognized by the members of a **circle of trust** as the entities that provide Web-based services to users. Service providers enter into partnerships with other service providers and identity providers with the goal of providing their common users with secure single sign-on between all parties of the **federation**.

service time

The time between the initiation of a request and the completion of the response to the request.

session key

A **secret key** that is used for the duration of one message or communication session.

SGA

See **System Global Area (SGA)**.

SHA

See **Secure Hash Algorithm (SHA)**.

shared server

A server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configuration, many user processes connect to a dispatcher. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means a small pool of server processes can server a large amount of clients. Contrast with dedicated server.

sibling

An entry that has the same parent as one or more other entries.

Signed Public Key And Challenge (SPKAC)

Signed Public Key And Challenge (SPKAC) is a proprietary protocol used by the Netscape Navigator browser to request certificates.

simple authentication

The process by which the client identifies itself to the server by means of a DN and a password which are not encrypted when sent over the network. In the simple authentication option, the server verifies that the DN and password sent by the client match the DN and password stored in the directory.

Simple Authentication and Security Layer (SASL)

A method for adding authentication support to connection-based protocols. To use this specification, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating a security layer for subsequent protocol interactions. The command has a required argument identifying a SASL mechanism.

single key-pair wallet

A **PKCS#12**-format wallet that contains a single user **certificate** and its associated **private key**. The **public key** is imbedded in the certificate.

single sign-off

The process by which you terminate an OracleAS Single Sign-On session and log out of all active partner applications simultaneously. You can do this by logging out of the application that you are working in.

single sign-on (SSO)

A process or system that enables a user to access multiple computer platforms or application systems after being authenticated only once.

single sign-on SDK

Legacy APIs to enable OracleAS Single Sign-On partner applications for single sign-on. The SDK consists of PL/SQL and Java APIs as well as sample code that demonstrates how these APIs are implemented. This SDK is now deprecated and **mod_osso** is used instead.

single sign-on server

Program logic that enables users to log in securely to single sign-on applications such as expense reports, mail, and benefits.

SLAPD

Standalone LDAP daemon. An LDAP directory server service that is responsible for most functions of a directory except replication.

slave

See **consumer**.

smart knowledge reference

A **knowledge reference** that is returned when the knowledge reference entry is in the scope of the search. It points the user to the server that stores the requested information.

SOAP

Simple Object Access Protocol (SOAP) is an **XML**-based protocol that defines a framework for passing messages between systems over the Internet via HTTP. A SOAP message consists of three parts — an envelope that describes the message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.

specific administrative area

Administrative areas control:

- Subschema administration
- Access control administration
- Collective attribute administration

A *specific* administrative area controls one of these aspects of administration. A specific administrative area is part of an autonomous administrative area.

SPKAC

See **Signed Public Key And Challenge (SPKAC)**.

sponsor node

In replication, the node that is used to provide initial data to a new node.

SSL

See **Secure Sockets Layer (SSL)**.

stream cipher

Stream ciphers are a type of **symmetric algorithm**. A stream cipher encrypts in small units, often a bit or a byte at a time, and implements some form of feedback mechanism so that the key is constantly changing. **RC4** is an example of a stream cipher.

See also: **block cipher**.

subACLSubentry

A specific type of **subentry** that contains **access control list (ACL)** information.

subclass

An object class derived from another object class. The object class from which it is derived is called its **superclass**.

subentry

A type of entry containing information applicable to a group of entries in a subtree. The information can be of these types:

- Access control policy points
- Schema rules
- Collective attributes

Subentries are located immediately below the root of an administrative area.

subordinate CA

In a hierarchical **public key infrastructure (PKI)**, the subordinate **Certificate Authority (CA)** is a CA whose certificate signature key is certified by another CA, and whose activities are constrained by that other CA.

subordinate reference

A **knowledge reference** pointing downward in the **directory information tree (DIT)** to a **naming context** that starts immediately below an entry

subschema DN

The list of **directory information tree (DIT)** areas having independent **schema** definitions.

subSchemaSubentry

A specific type of **subentry** containing **schema** information.

subtree

A section of a directory hierarchy, which is also called a **directory information tree (DIT)**. The subtree typically starts at a particular directory node and includes all subdirectories and objects below that node in the directory hierarchy.

subtype

An attribute with one or more options, in contrast to that same attribute without the options. For example, a `commonName` (`cn`) attribute with American English as an option is a subtype of the `commonName` (`cn`) attribute without that option. Conversely, the `commonName` (`cn`) attribute without an option is the **supertype** of the same attribute with an option.

success URL

When using Oracle Application Server Single Sign-On, the URL to the routine responsible for establishing the session and session cookies for an application.

super user

A special directory administrator who typically has full access to directory information.

superclass

The **object class** from which another object class is derived. For example, the object class `person` is the superclass of the object class `organizationalPerson`. The latter, namely, `organizationalPerson`, is a **subclass** of `person` and inherits the attributes contained in `person`.

superior reference

A **knowledge reference** pointing upward to a **directory system agent (DSA)** that holds a naming context higher in the **directory information tree (DIT)** than all the naming contexts held by the referencing DSA.

supertype

An attribute without options, in contrast to the same attribute with one or more options. For example, the `commonName` (`cn`) attribute without an option is the supertype of the same attribute with an option. Conversely, a `commonName` (`cn`) attribute with American English as an option is a **subtype** of the `commonName` (`cn`) attribute without that option.

supplier

In replication, the server that holds the master copy of the **naming context**. It supplies updates from the master copy to the **consumer** server.

symmetric algorithm

A symmetric algorithm is a cryptographic algorithm that uses the same key for encryption and decryption. There are essentially two types of symmetric (or secret key) algorithms — **stream cipher**s and **block cipher**s.

symmetric cryptography

Symmetric cryptography (or shared secret cryptography) systems use the same key to encipher and decipher data. The problem with symmetric cryptography is ensuring a secure method by which the sender and recipient can agree on the secret key. If a third party were to intercept the secret key in transit, they could then use it to decipher anything it was used to encipher. Symmetric cryptography is usually faster than asymmetric cryptography, and is often used when large quantities of data need to be exchanged. **DES**, **RC2**, and **RC4** are examples of symmetric cryptography algorithms.

symmetric key

See **secret key**.

System Global Area (SGA)

A group of shared memory structures that contain data and control information for one Oracle database instance. If multiple users are concurrently connected to the same instance, the data in the instance SGA is shared among the users. Consequently, the SGA is sometimes referred to as the "shared global area." The combination of the background processes and memory buffers is called an Oracle instance.

system operational attribute

An attribute holding information that pertains to the operation of the directory itself. Some operational information is specified by the directory to control the server, for example, the time stamp for an entry. Other operational information, such as access information, is defined by administrators and is used by the directory program in its processing.

think time

The time the user is not engaged in actual use of the processor.

third-party access management system

Non-Oracle single sign-on system that can be modified to use OracleAS Single Sign-On to gain access to Oracle Application Server applications.

throughput

The number of requests processed byOracle Internet Directory for each unit of time. This is typically represented as "operations per second."

Time Stamp Protocol (TSP)

Time Stamp Protocol (TSP), as specified in RFC 3161, defines the participating entities, the message formats, and the transport protocol involved in time stamping a digital message. In a TSP system, a trusted third-party Time Stamp Authority (TSA) issues time stamps for messages.

TLS

See **Transport Layer Security (TLS)**.

Transport Layer Security (TLS)

A protocol providing communications privacy over the Internet. The protocol enables client/server applications to communicate in a way that prevents eavesdropping, tampering, or message forgery.

Triple Data Encryption Standard (3DES)

Triple Data Encryption Standard (3DES) is based on the **Data Encryption Standard (DES)** algorithm developed by IBM in 1974, and was adopted as a national standard in 1977. 3DES uses three 64-bit long keys (overall key length is 192 bits, although actual key length is 56 bits). Data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key. This makes 3DES three times slower than standard DES but also three times more secure.

trusted certificate

A third party identity that is qualified with a level of trust. The trust is used when an identity is being validated as the entity it claims to be. Typically, trusted certificates come from a **Certificate Authority (CA)** you trust to issue user certificates.

trustpoint

See **trusted certificate**.

TSP

See **Time Stamp Protocol (TSP)**.

Unicode

A type of universal character set, a collection of 64K characters encoded in a 16-bit space. It encodes nearly every character in just about every existing character set standard, covering most written scripts used in the world. It is owned and defined by Unicode Inc. Unicode is canonical encoding which means its value can be passed around in different locales. But it does not guarantee a round-trip conversion between it and every Oracle character set without information loss.

UNIX Crypt

The UNIX encryption algorithm.

URI

Uniform Resource Identifier (URI). A way to identify any point of content on the Web, whether it be a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form or subset of URI called a **URL**.

URL

Uniform Resource Locator (URL). The address of a file accessible on the Internet. The file can be a text file, HTML page, image file, a program, or any other file supported by HTTP. The URL contains the name of the protocol required to access the resource, a domain name that identifies a specific computer on the Internet, and a hierarchical description of the file location on the computer.

URLC token

The OracleAS Single Sign-On code that passes authenticated user information to the **partner application**. The partner application uses this information to construct the session cookie.

user name mapping module

A OracleAS Single Sign-On Java module that maps a user **certificate** to the user's nickname. The nickname is then passed to an authentication module, which uses this nickname to retrieve the user's certificate from the directory.

user search base

In the Oracle Internet Directory default **directory information tree (DIT)**, the node in the identity management realm under which all the users are placed.

UTC (Coordinated Universal Time)

The standard time common to every place in the world. Formerly and still widely called Greenwich Mean Time (GMT) and also World Time, UTC nominally reflects the mean solar time along the Earth's prime meridian. UTC is indicated by a z at the end of the value, for example, 200011281010z.

UTF-8

A variable-width 8-bit encoding of **Unicode** that uses sequences of 1, 2, 3, or 4 bytes for each character. Characters from 0-127 (the 7-bit ASCII characters) are encoded with one byte, characters from 128-2047 require two bytes, characters from 2048-65535 require three bytes, and characters beyond 65535 require four bytes. The Oracle character set name for this is AL32UTF8 (for the Unicode 3.1 standard).

UTF-16

16-bit encoding of **Unicode**.The Latin-1 characters are the first 256 code points in this standard.

verification

Verification is the process of ensuring that a given **digital signature** is valid, given the **public key** that corresponds to the **private key** purported to create the signature and the data block to which the signature purportedly applies.

virtual host

A single physical Web server machine that is hosting one or more Web sites or domains, or a server that is acting as a proxy to other machines (accepts incoming requests and reroutes them to the appropriate server).

In the case of OracleAS Single Sign-On, virtual hosts are used for load balancing between two or more OracleAS Single Sign-On servers. They also provide an extra layer of security.

virtual host name

In an Oracle Application Server Cold Failover Cluster (Identity Management), the host name corresponding to a particular virtual IP address.

virtual IP address

In an Oracle Application Server Cold Failover Cluster (Identity Management), each physical node has its own physical IP address and physical host name. To present a single system image to the outside world, the cluster uses a dynamic IP address that can be moved to any physical node in the cluster. This is called the virtual IP address.

wait time

The time between the submission of the request and initiation of the response.

wallet

An abstraction used to store and manage security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services. A wallet resource locator (WRL) provides all the necessary information to locate the wallet.

Wallet Manager

See **Oracle Wallet Manager**.

Web service

A Web service is application or business logic that is accessible using standard Internet protocols, such as **HTTP**, **XML**, and **SOAP**. Web Services combine the best aspects of component-based development and the World Wide Web. Like components, Web Services represent black-box functionality that can be used and reused without regard to how the service is implemented.

Web Services Description Language (WSDL)

Web Services Description Language (WSDL) is the standard format for describing a Web service using **XML**. A WSDL definition describes how to access a Web service and what operations it will perform.

WSDL

See **Web Services Description Language (WSDL)**.

WS-Federation

Web Services Federation Language (WS-Federation) is a specification developed by Microsoft, IBM, BEA, VeriSign, and RSA Security. It defines mechanisms to allow **federation** between entities using different or like mechanisms by allowing and brokering trust of identities, attributes, and authentication between participating **Web service**s.

See also: **Liberty Alliance**.

X.500

X.500 is a standard from the International Telecommunication Union (ITU) that defines how global directories should be structured. X.500 directories are hierarchical with different levels for each category of information, such as country, state, and city.

X.509

X.509 is the most widely used standard for defining digital certificates. A standard from the International Telecommunication Union (ITU), for hierarchical directories

with authentication services, used in many **public key infrastructure (PKI)** implementations.

XML

Extensible Markup Language (XML) is a specification developed by the World Wide Web Consortium (W3C). XML is a pared-down version of Standard Generalized Mark-Up Language (SGML), designed especially for Web documents. XML is a metalanguage (a way to define tag sets) that allows developers to define their own customized markup language for many classes of documents.

XML canonicalization (C14N)

This is a process by which two logically equivalent XML documents can be resolved to the same physical representation. This has significance for digital signatures because a signature can only verify against the same physical representation of the data against which it was originally computed. For more information, see the W3C's XML Canonicalization specification.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X