Oracle® Application Server Certificate Authority

Administrator’s Guide

10*g* (9.0.4)

Part No. B10663-01

September 2003

ORACLE

Oracle Application Server Certificate Authority Administrator's Guide, 10*g* (9.0.4)

Part No. B10663-01

Primary Author: Jeff Levinger

Contributors: Lakshmi Kethana, Mode Nalini, Paul Needham, Shreedhar Patwari, Deepak Ramakrishnan, Gary Truong, Miranda Zhai

Contents

2 Identity Management and OracleAS Certificate Authority Features

3 Introduction to OCA Administration and Certificate Management

5 Managing Policies in Oracle Application Server Certificate Authority

6 OracleAS Certificate Authority Administration: Advanced Topics

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

B Setting up a CA Hierarchy

C Known Troubleshooting Tips

D Extensions

E Glossary

Index

Send Us Your Comments

Oracle Application Server Certificate Authority Administrator’s Guide, 10*g* (9.0.4)

Part No. B10663-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager
- Postal service:

  Oracle Corporation
  Server Technologies Documentation
  500 Oracle Parkway, Mailstop 4op11
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Oracle Application Server Certificate Authority 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 (OCA) enables end-entities (users and servers) to authenticate themselves using certificates that OCA issues based on SSO, SSL, or other pre-existing authentication methods. Use of these certificates makes authentication a speedier and more secure process, relying on certificate identification. Each certificates is published to OID when it is issued and removed when it expires or is revoked. Users can access the OCA web interface to request issuance, revocation, or renewal of their own certificates. No special privilege is required for end-users to access the OCA web interface. However, certificate issuance, revocation, or renewal requires either prior authentication by SSO, SSL, or OCA or, in the absence of these, manual authentication by the OCA administrator.

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
- Documentation Accessibility
- Oracle Identity Management
- Organization
- Related Documentation
- Conventions

Audience

This guide is intended for Oracle Application Server Certificate Authority administrators who will manage certificate requests and certificate-related operations.

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. Standards will continue to evolve over time, and Oracle Corporation 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 additional information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Oracle Identity Management

The Oracle Application Server Certificate Authority (OCA) 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** (OID), 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** (SSO), 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, etc.

  In addition to its use of SSL, OC4J, and HTTP Server, Oracle Application Server Certificate Authority has a built-in reliance on SSO and OID. OCA publishes each valid certificate in an OID entry for the DN in use. SSO and other components can rely on these OID entries because OCA removes revoked and expired certificates from OID on a regular basis. The administrator also has the option of configuring OCA to publish its URL through SSO. This configuration choice causes every SSO-authenticated user who lacks a certificate to see the OCA page for requesting one. OCA certificates can be used to authenticate to any Oracle component or to authorize use of any application that is SSO-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 10g Installation Guide*. For the recommended deployment configuration and installation procedure, see section 11.9 of that Guide.

Organization

Each chapter in this document is described in the following table.

Chapter #	Chapter Reference	Description
1	**Public Key Infrastructure and OracleAS**	Briefly describes public key infrastructure and its Oracle implementation
2	**Identity Management and OracleAS Certificate Authority Features**	Describes the key features & interface (scalable, web-browser) to administer industry-standard certificates, integrate with LDAP directories and Single Sign-On, and apply policies
3	**Introduction to OCA Administration and Certificate Management**	Describes using the web administrator interface to accomplish OCA administration and certificate management
4	**Configuring Oracle Application Server Certificate Authority**	Describes the OCA user interface to request renew, or revoke certificates
5	**Managing Policies in Oracle Application Server Certificate Authority**	Describes how to manage or modify policies delivered with OCA, and how to make and manage new ones, for handling requests to issue, renew, or revoke certificates. The Administrator can modify policies using the web interface.
6	**OracleAS Certificate Authority Administration: Advanced Topics**	Describes Oracle Application Server Certificate Authority's requirements and interactions with Oracle® Application Server High Availability features and standard back-up-and-recovery procedures
7	**End-User Interface of the Oracle Application Server Certificate Authority**	Describes the web interface for end-users to request, acquire, renew, or revoke certificates

Chapter #	Chapter Reference	Description
A	**Command-Line Administration**	Presents syntax & examples for all uses of the `ocactl` command line tool for administration and certificate management
B	**Setting up a CA Hierarchy**	Describes how to acquire and import a subordinate certificate authority, which is a CA whose certificate is signed by some higher CA authority.
C	**Known Troubleshooting Tips**	Presents workarounds and other suggestions for handling certain issues or error messages that can arise while installing, administering, or using Oracle Application Server Certificate Authority.
D	**Glossary**	Describes X.509 V3 and IETF's PKIX standard extensions, with which Oracle Application Server Certificate Authority is compliant

Related Documentation

- *Oracle Application Server 10g Installation Guide*
- *Oracle Application Server 10g Administrator's Guide*
- *Oracle Application Server 10g Security Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle Application Server 10g 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.

In North America, printed documentation is available for sale in the Oracle Store at

```
http://oraclestore.oracle.com/
```

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

```
http://www.oraclebookshop.com/
```

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://technet.oracle.com/membership/index.htm`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://technet.oracle.com/docs/index.htm`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle10i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
`UPPERCASE monospace (fixed-width font)`	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a `NUMBER` column. You can back up the database by using the `BACKUP` command. Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view. Use the `DBMS_STATS.GENERATE_STATS` procedure.

Convention	Meaning	Example
`lowercase monospace (fixed-width font)`	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to open SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true.` Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase monospace (fixed-width font) italic`	Lowercase monospace italic font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`*` [ , `*`precision`*` ])`
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`

|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}`
		`[COMPRESS | NOCOMPRESS]`

Convention	Meaning	Example

. . .	Horizontal ellipsis points indicate either:
	■ That we have omitted parts of the code that are not directly related to the example
	■ That you can repeat a portion of the code
		CREATE TABLE ... AS *subquery*;
		SELECT *col1*, *col2*, ... , *coln* FROM employees;

. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	

Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2);
		acct CONSTANT NUMBER(4) := 3;

Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*

UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees;
		SELECT * FROM USER_TABLES;
		DROP TABLE hr.employees;

lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		SELECT last_name, employee_id FROM employees;
		sqlplus hr/hr
		CREATE USER mjones IDENTIFIED BY ty3MU9;

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 Oracle Application Server Certificate Authority 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 the covers the following topics:

- What Is a PKI?
- Benefits of a PKI
- Introduction to the OracleAS PKI

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:

- Key Pairs
- Certification Authority (CA) and Digital Certificates
- 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 pairs, 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

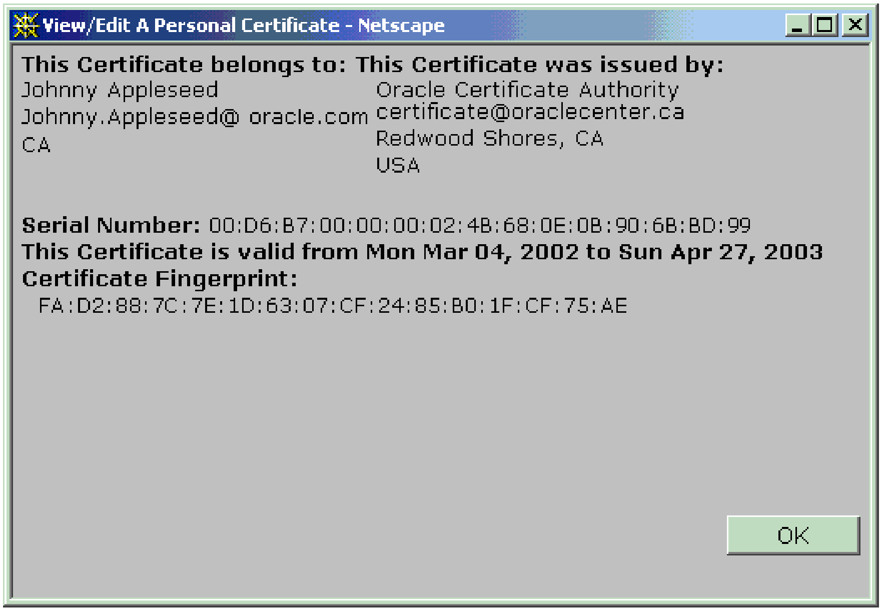
Digital certificates issued by Oracle Application Server Certificate Authority comply with the X.509, Version 3, ISO standard and with RFC 2459, promulgated by the PKIX working group of The Internet Engineering Task Force, `http://www.ietf.org/`.

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.
- 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 contain all of these elements.

Figure 1–1 A Certificate Issued by Oracle Application Server Certificate Authority



This Certificate belongs to:
Johnny Appleseed
Johnny.Appleseed@ oracle.com
CA

This Certificate was issued by:
Oracle Certificate Authority
certificate@oraclecenter.ca
Redwood Shores, CA
USA

Serial Number: 00:D6:B7:00:00:00:02:4B:68:0E:0B:90:6B:BD:99
This Certificate is valid from Mon Mar 04, 2002 to Sun Apr 27, 2003
Certificate Fingerprint:
FA:D2:88:7C:7E:1D:63:07:CF:24:85:B0:1F:CF:75:AE

Oracle Application Server Certificate Authority issues and works with X.509 certificates, supporting multiple certificate types.

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 RA 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
- Benefits of the OracleAS PKI
- Components of the OracleAS PKI

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 Oracle Application Server Certificate Authority 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 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.

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.

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 Oracle Application Server Certificate Authority's Server/SubCA form for requesting an OCA certificate. See Server/SubCA Certificates Tab in Chapter 7, "End-User Interface of the Oracle Application Server Certificate Authority".

These wallets conform to the PKCS #12 standard, and are the containers used by Oracle Application Server Certificate Authority. 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 Oracle Application Server Certificate Authority. 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, Oracle Application Server 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 Oracle Application Server Certificate Authority, a few clicks generates, submits, and stores a certificate. As a result, credential verification and authentication is simple and fast.

Oracle Application Server Certificate Authority is an optional infrastructure component in Oracle Application Server.

2

Identity Management and OracleAS Certificate Authority Features

Oracle Application Server Certificate Authority (OCA) provides secure mechanisms whereby it creates and signs X.509 v3 digital certificates for clients and servers. OCA enforces policies chosen or created by its administrator, as described in Chapter 5, and is controlled by that administrator through the scalable web-based interface described in Chapter 4. OCA provides a secure infrastructure for supporting and managing such certificates, including the web-based user interface described in Chapter 7.

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

- Identity Management Components and Architecture
- Key Features of Oracle Application Server Certificate Authority
- Automatic or Conventional Provisioning
- Hierarchical Certificate Authority Support
- Deployments and Installations

Identity Management Components and Architecture

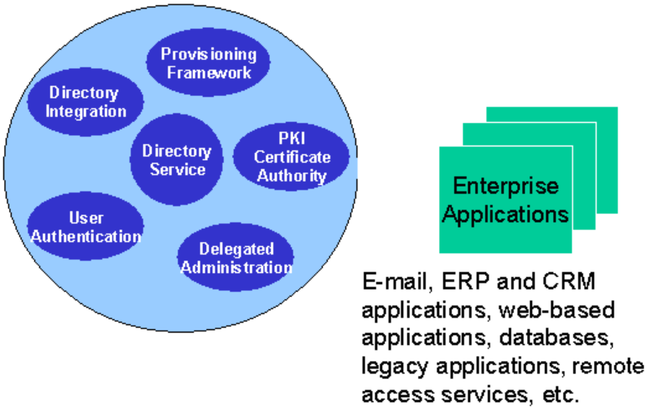
A complete identity management solution includes the following components:

- 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 stand-alone.

- 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


Provisioning Framework
Directory Integration
Directory Service
PKI Certificate Authority
User Authentication
Delegated Administration
Enterprise Applications
E-mail, ERP and CRM applications, web-based applications, databases, legacy applications, remote access services, etc.

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

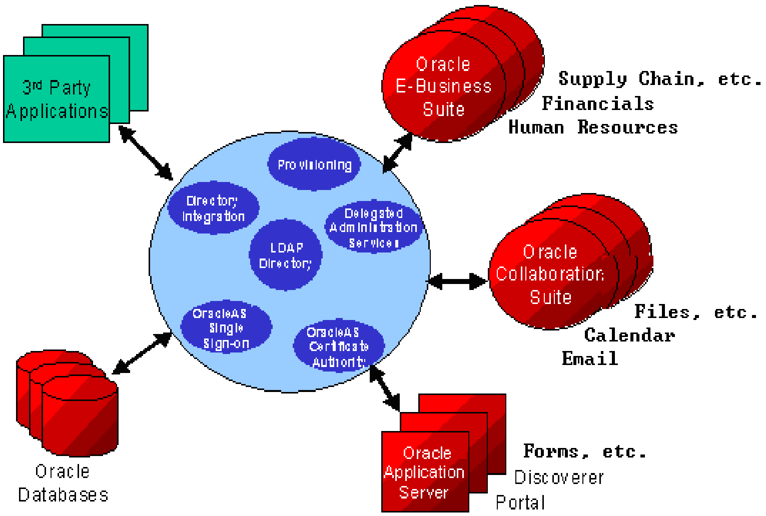
- Oracle Identity Management
- Leveraging Oracle Identity Management in the Enterprise
- Role of Oracle Identity Management in the Oracle Security Architecture
- Role of OracleAS Certificate Authority in Oracle Identity Management
- Simplified Provisioning through SSO Integration

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.
- OracleAS 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

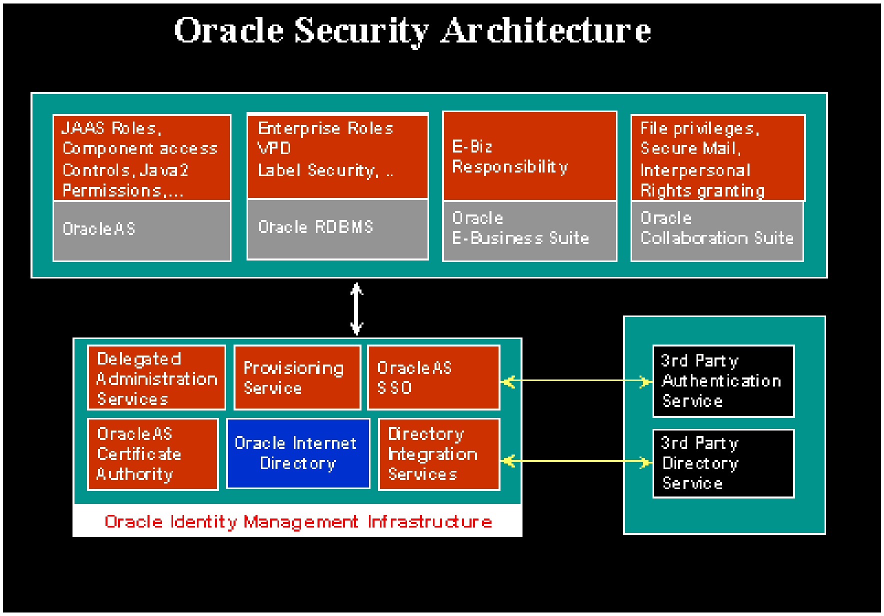
While Oracle Identity Management is designed to provide an enterprise infrastructure for Oracle products, it also serves as a robust and scalable identity management solution for custom and third-party enterprise applications, hardware and network operating systems of the enterprise.

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, the RDBMS, the Application Server, the E-business Suite, and the 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


Oracle Security Architecture
JAAS Roles, Component access Controls, Java2 Permissions,...
OracleAS
Enterprise Roles VPD Label Security, ..
Oracle RDBMS
E-Biz Responsibility
Oracle E-Business Suite
File privileges, Secure Mail, Interpersonal Rights granting
Oracle Collaboration Suite
Delegated Administration Services
Provisioning Service
OracleAS SSO
OracleAS Certificate Authority
Oracle Internet Directory
Directory Integration Services
Oracle Identity Management Infrastructure
3rd Party Authentication Service
3rd Party Directory Service

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 above, 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, Collaboration Suite components such as E-Mail and Voice mail use the Oracle Internet Directory to manage product-specific user preferences, user personal contacts and address book etc. These components rely on Oracle Application Server Certificate Authority 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 Oracle Application Server Certificate Authority support the secure authentication needed for simple, fast, consistent identity management.

Simplified Provisioning through SSO Integration

An application user authenticating to the OracleAS Single Sign-On (SSO) Server 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 SSO, providing increased security. The issued PKI certificate is automatically published in the Oracle Internet Directory (OID). 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 Server.

The Oracle Application Server Certificate Authority (OCA) administrator can optionally configure OCA to broadcast its URL through SSO. Doing so enables users authenticating through SSO to use OCA’s easy graphical interface to apply for a certificate. Having such a certificate makes future SSO authentication even easier, because SSO can then use OID to validate the certificate automatically supplied by the user’s browser. SSO can rely on the information in the directory because OCA automatically deletes revoked and expired certificates from the directory on a regular basis.

Key Features of Oracle Application Server Certificate Authority

Oracle Application Server 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
- National Language Support (NLS) for OCA Screens
- Scalability, Performance, and High Availability
- Automatic or Conventional Provisioning

Support for Open Standards

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

- X.509 version 3 certificates and certificate revocation list (CRL)
- 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, etc.)
- Multiple enrollment protocols for certificate requests such as Signed Public Key and Challenge (SPKAC) and Public Key Cryptography Standard (PKCS) #10 for certificate requests

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 Oracle Application Server Certificate Authority 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 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, e.g., if security has been compromised. (Stopping and starting OCA require the administrator to use the command-line tool `ocactl`, which requires his password.)

The end-user web interface for Oracle Application Server Certificate Authority 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 SSO authentication and click **Submit**, an SSO window appears in which you can enter your SSO 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, download the CRL (Certificate Revocation List), or change your method of authentication.

When you click the Server/SubCA Certificates tab, you can request a new Server/SubCA certificate, download the CRL, or download the CA certificate. You can also search for particular certificates or certificate requests by ID/Serial number or by common name.

National Language Support (NLS) 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 OCA supports; otherwise English is the language used. While OCA's command line tool, `ocactl`, uses only commands in English, messages (informational, error messages, etc.) are displayed in the language of the server locale, if supported; otherwise English appears.

> **See Also:** "Configuring National Language Support (NLS) for OCA Screens" in Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics"

Scalability, Performance, and High Availability

Oracle Application Server 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

Automatic or Conventional Provisioning

Conventional provisioning has an administrator issuing certificates to users. The automatic provisioning provided by Oracle Application Server Certificate Authority using SSO and SSL reduces the costs and delays of conventional methods for supporting PKI.

For SSO authentication, Oracle Application Server Certificate Authority uses mod_osso and Oracle Single Sign-on server. These methods simplify certificate management by helping Oracle Application Server Certificate Authority issue certificates to users who have been authenticated automatically by SSO.

A user who has previously been issued an X.509v3 certificate can submit that certificate over HTTPS as a means of authenticating to the Oracle Certificate Authority. Assuming the certificate was issued by the same Oracle 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.

OCA 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.

Oracle Application Server Certificate Authority supports the following authentication methods, explained in the following sections:

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

Oracle Single Sign-on Authentication

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

Oracle Application Server 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 the OracleAS Single Sign-On Server can choose to request a digital certificate from the Oracle Certificate Authority. The OracleAS Single Sign-On Server can make this easy by displaying a "get certificate" pop-up page, if OCA 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 Server 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 SSO authentication will then fail.

Secure Socket Layer (SSL-based) Authentication

Oracle Application Server Certificate Authority supports certificate-based authentication, so a user's prior, unrevoked X.509 v3 certificate will authenticate that user to Oracle Application Server Certificate Authority over HTTPS. Having thus authenticated the user, Oracle Application Server 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. Oracle 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 that Oracle Application Server Certificate Authority accepts 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 download existing certificates. Server and subordinate CA's can also request certificates using this manual process.

Hierarchical Certificate Authority Support

Oracle Application Server 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.

Oracle Application Server 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 Oracle 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.

Deployments and Installations

Oracle Application Server Certificate Authority (OCA) can work with several different deployment strategies for the following components that OCA needs:

- Oracle HTTP Server (OHS) (must be on the same machine as OCA)
- OC4J for OCA (must be on the same machine as OCA)
- Infrastructure metadata repository
- Oracle Internet Directory (OID)
- and optionally Oracle Single Sign-on Server (SSO)

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

In the *default* deployment, all these components are on the same machine and in the same Oracle Home, as shown in Figure 2–4. This configuration is ideal for development and non-production environments, and is the default installation configuration. The installation instructions for this default deployment configuration of OCA appear in Section 6.14 of the *Oracle Application Server 10g Installation Guide*.

In the *recommended* production deployment, OHS, OC4J, OCA and the infrastructure metadata repository will be on one machine, in one Oracle Home. The remaining components like SSO and OID will be 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 OCA in a very secure location. Since OCA is at the top of the trust chain for certificates, these additional protections are prudent in a production environment, as is illustrated in Figure 2–5. Similarly, it is better for Oracle Application Server Certificate Authority security reasons not to use Enterprise Manager for starting or stopping these components.

Installation instructions for this recommended deployment configuration appear in Section 6.20 of the *Oracle Application Server 10g Installation Guide*.

Figure 2–4 Oracle Application Server Certificate Authority Default Installation

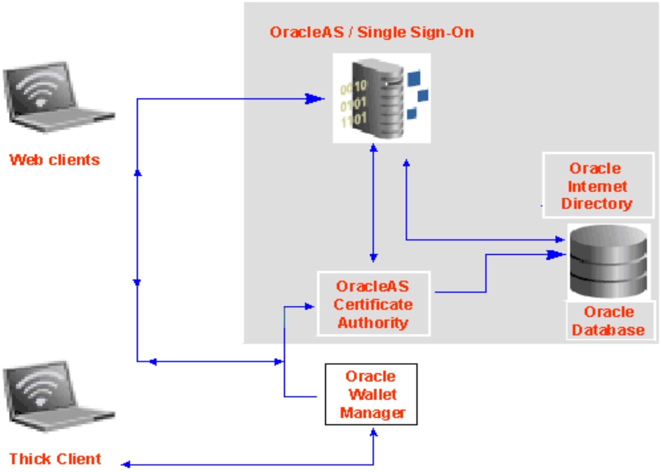
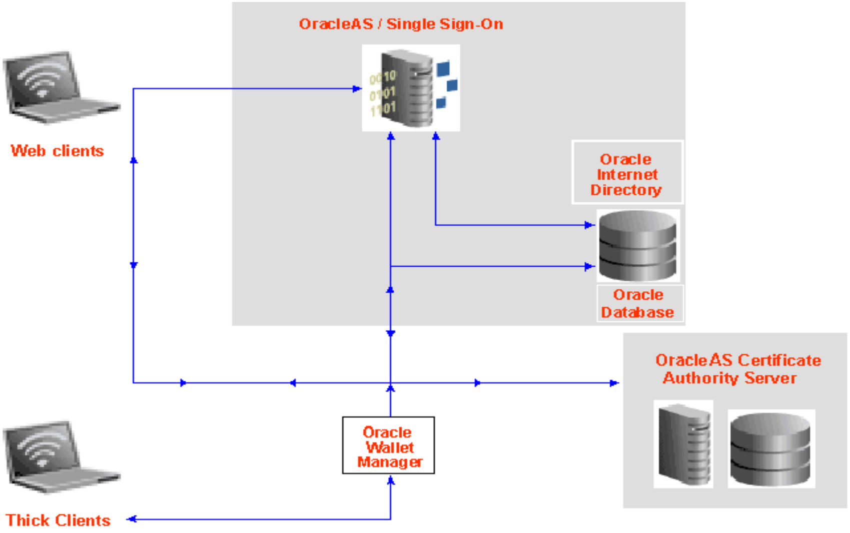

OracleAS / Single Sign-On
Web clients
Oracle Internet Directory
OracleAS Certificate Authority
Oracle Database
Oracle Wallet Manager
Thick Client

Figure 2–5 Oracle Application Server Certificate Authority Recommended Production Installation


OracleAS / Single Sign-On
Web clients
Oracle Internet Directory
Oracle Database
OracleAS Certificate Authority Server
Oracle Wallet Manager
Thick Clients

3

Introduction to OCA 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 Oracle Application Server Certificate Authority activity

This chapter describes the first of those three areas: certificate management. The other two are described in Chapter 4, "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 Oracle Application Server Certificate Authority, as explained in later sections, along with requesting or replacing the administrator’s certificate.

For end-user interactions with Oracle Application Server Certificate Authority, a separate web interface presents forms enabling personal certificate-related operations: see Chapter 7, "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
- Requesting the Administrator Certificate
- Replacing the Administrator Certificate

- Overview of the OracleAS Certificate Authority Administration Interface
- Managing Certificates
- Updating the Certificate Revocation List (CRL)
- Single Sign-on (SSO) and OracleAS Certificate Authority (OCA)
- Default Install Values for OracleAS Certificate Authority

Starting and Stopping Oracle Application Server Certificate Authority

For security reasons, OCA'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 on page 3-6. 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 Oracle Single Sign-on Server (SSO)
- Oracle HTTP Server (OHS)
- OC4J for OCA

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

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

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

To start, stop, or restart Oracle Application Server Certificate Authority, enter the corresponding command from those shown below, on the command line:

1. To stop Oracle Application Server Certificate Authority, use this command:

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

2. To start (or restart) Oracle Application Server Certificate Authority, use this command:

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

3. 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 Oracle Application Server 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 Oracle Application Server Certificate Authority is started for the first time. You must be accessing Oracle Application Server 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 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: 1024) and the validity period (default: 1 year).

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

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

To request your certificate, perform the following seven steps:

1. Access the Oracle Application Server 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` as the host on which OCA 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`.

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.

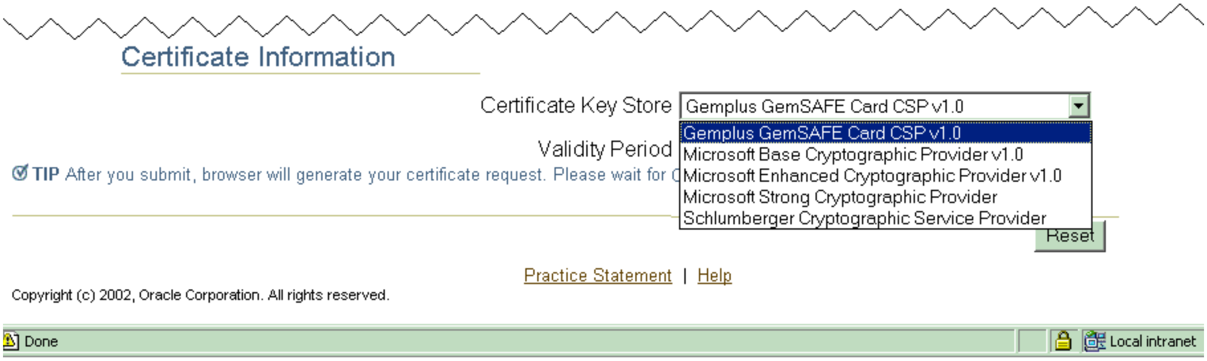
Table 3–1 DN Information for the Administrator's Certificate

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

- **Certificate Authority Administrator Password:** Only the Oracle Application Server Certificate Authority administrator can do certificate and configuration management. This person is initially authenticated by entering here the password as entered during OCA installation, in the screen named "Specify OCA Administrator Password".
- **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 **Key Size** appears, referring to the size in bits of the key-pair to be generated: 512, 1024, ... Choose the size appropriate to your site: 1024 is a standard default, providing good security. Higher numbers improve the security at some price in performance.

  * In Internet Explorer, the phrase **Key Store** appears, referring to a choice of providers for cryptography service. Standard choices include Microsoft Basic Crypto Provider, Microsoft Enhanced Crypto Provider, and Microsoft Strong Cryptographic Provider, for which the key sizes are fixed at 512, 1024, and 2048 bits, respectively. Other choices may also be present, such as Gemplus for smart card usage. This section of the form will look like this:



Certificate Information

Certificate Key Store: Gemplus GemSAFE Card CSP v1.0

Validity Period

TIP After you submit, browser will generate your certificate request. Please wait for (

Copyright (c) 2002, Oracle Corporation. All rights reserved.

Gemplus GemSAFE Card CSP v1.0
Microsoft Base Cryptographic Provider v1.0
Microsoft Enhanced Cryptographic Provider v1.0
Microsoft Strong Cryptographic Provider
Schlumberger Cryptographic Service Provider
Reset
Practice Statement | Help
Done
Local intranet

Oracle Application Server Certificate Authority recommends using Microsoft Enhanced Cryptographic Provider for the Administrator Certificate. However, if readers for smartcards like Gemplus are available, they should be used; if no reader is installed, selecting smartcard suppliers like Gemplus or Schlumberger 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 the **Submit** button. (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 **Import Certificate**. (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 Oracle Application Server Certificate Authority, as described in Chapter 4, "Configuring Oracle Application Server 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 OCA Administrator password. For security reasons, these commands are only enabled on the command line.

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 above in Requesting the Administrator Certificate.

Here are the three relevant command-line tasks:

1. To stop the Oracle Application Server 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 Oracle Application Server 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 on page 3-3 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, "Known Troubleshooting Tips", in section 1. Prerequisite Issues and Warnings, item b. Issue: Cannot Log in as Administrator after Logging in as Normal User.

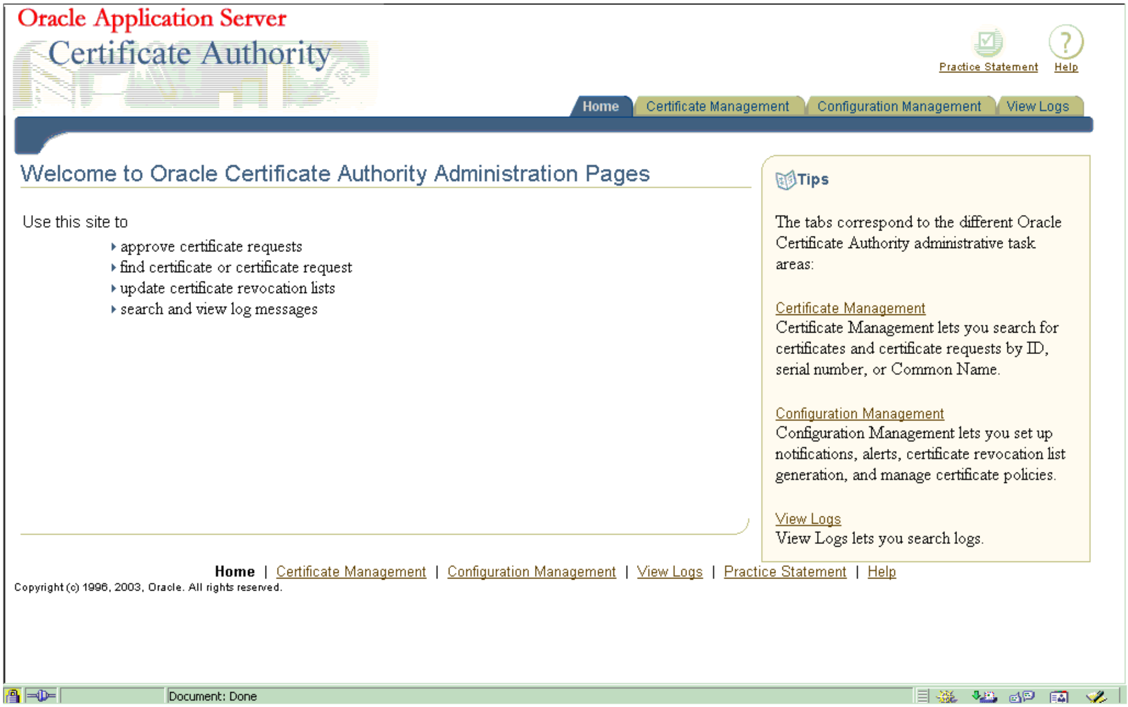
To access the Oracle Application Server Certificate Authority administration interface, launch your web browser. Enter the URL and port number of the administration server as they were displayed at the end of installation:

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

where `oracle_HTTP_HOST` as the host on which OCA 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.

The Oracle Application Server Certificate Authority home page appears, presenting three additional subtabs, as the following figure shows:



Welcome to Oracle Certificate Authority Administration Pages

Use this site to

- approve certificate requests
- find certificate or certificate request
- update certificate revocation lists
- search and view log messages

Tips

The tabs correspond to the different Oracle Certificate Authority administrative task areas:

Certificate Management
Certificate Management lets you search for certificates and certificate requests by ID, serial number, or Common Name.

Configuration Management
Configuration Management lets you set up notifications, alerts, certificate revocation list generation, and manage certificate policies.

View Logs
View Logs lets you search logs.

Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

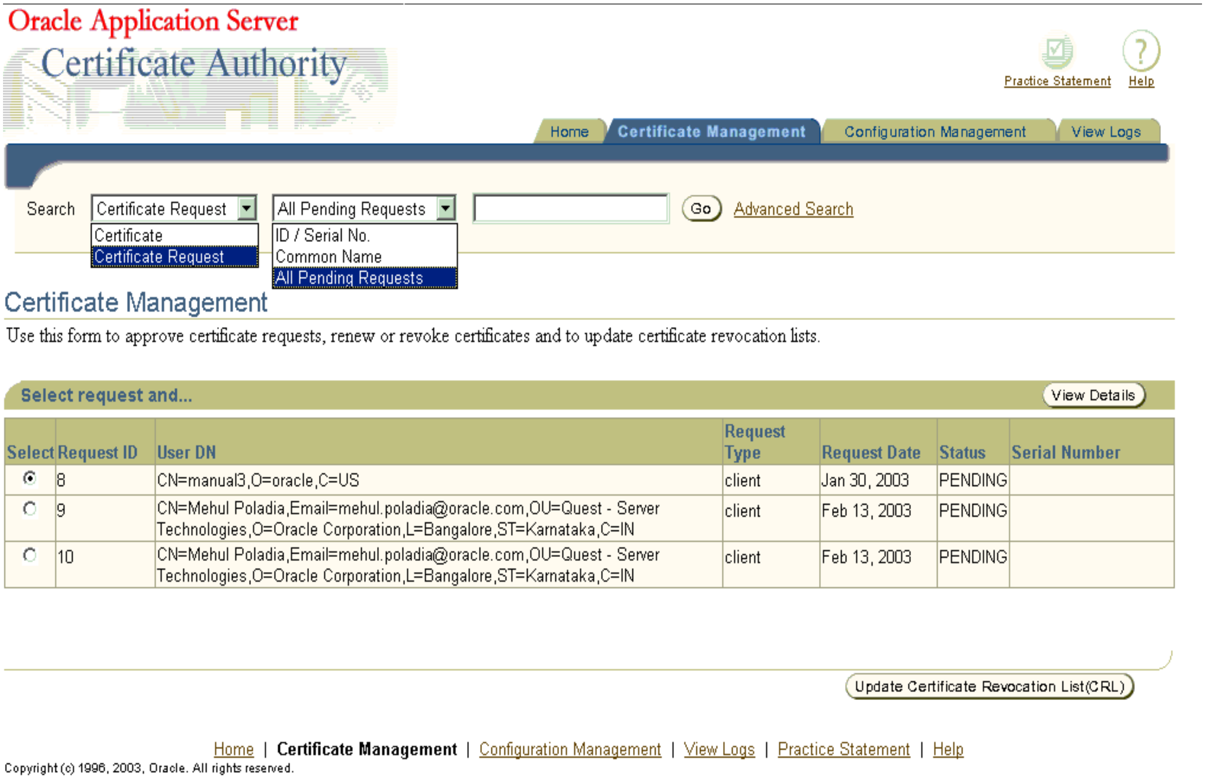
Document: Done

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 4
- View Logs Tab, described in Chapter 4, "Configuring Oracle Application Server Certificate Authority"

Certificate Management Tab

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



Certificate Management

Use this form to approve certificate requests, renew or revoke certificates and to update certificate revocation lists.

Select	Request ID	User DN	Request Type	Request Date	Status	Serial Number
●	8	CN=manual3,O=oracle,C=US	client	Jan 30, 2003	PENDING	
○	9	CN=Mehul Poladia,Email=mehul.poladia@oracle.com,OU=Quest - Server Technologies,O=Oracle Corporation,L=Bangalore,ST=Karnataka,C=IN	client	Feb 13, 2003	PENDING	
○	10	CN=Mehul Poladia,Email=mehul.poladia@oracle.com,OU=Quest - Server Technologies,O=Oracle Corporation,L=Bangalore,ST=Karnataka,C=IN	client	Feb 13, 2003	PENDING	

Update Certificate Revocation List(CRL)
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

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 5, "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 one, follow the steps in the corresponding section below.

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 import 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" on page 3-14.

To display a list of certificates, see "Using Advanced Search" on page 3-15.

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 import the selected certificate.)

Revoking Certificates

As the administrator, you can revoke certificates, 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.

To find the target certificate, follow the instructions in "Listing a Single Certificate Request or Issued Certificate" on page 3-14 or "Using Advanced Search" on page 3-15. 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 SSO or SSL authentication can also revoke their own certificates, as described in Certificate Revocation in Chapter 7, "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 Oracle Application Server 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 Appendix A, "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 OCA Web Administrator's Certificate in Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics".

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 SSO- 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 allows you to 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 **Import to 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, the latest database list of revoked certificates is placed in your browser to prevent entities with revoked certificates from being authenticated.

See also: Updating the Certificate Revocation List (CRL)

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, etc. 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 import 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. Oracle Application Server 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 below, 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, Oracle Application Server 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.

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 owner’s organizational unit or division
Organization	Name of the company or organization to which the owner belongs
Location	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: The section entitled Domain Component Attributes in Appendix E, "Glossary".

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**.

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.

You generate an updated CRL 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.

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

You can retrieve it for review or saving by choosing **Download CRL** then **Import to Browser** or **Download to your local disk**.

> **See also:** Downloading Certificate Revocation Lists into Your File System in Chapter 7, "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.

Single Sign-on (SSO) and OracleAS Certificate Authority (OCA)

OCA and SSO complement each other in simplifying the provisioning of user certificates and using them to enable PKI authentication to all applications that use SSO. The two configuration choices described in this section can make this collaboration even easier:

- Broadcasting the OCA Certificate Request URL to SSO-Authenticated Users
- Bringing SSO-Authenticated Users to the OCA Certificate Request URL

The first configuration choice, broadcasting, makes it even easier for an SSO user to file a certificate request than it is using the default OCA configuration. OCA's default is to provide certificates when an SSO-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 7, "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 SSO / OCA certificate directly.

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

The last two subsections are

- Enabling PKI Authentication with SSO and OCA
- User Certificates and SSO Usage

They describe all the steps required for PKI authentication with OCA and SSO, and the process Single Sign-On uses for authentication.

Broadcasting the OCA Certificate Request URL to SSO-Authenticated Users

The URL at which SSO users can get an OCA 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.

where `oracle_HTTP_HOST` is the host on which OCA is installed, and oca_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.

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

Bringing SSO-Authenticated Users to the OCA Certificate Request URL

Although OCA is configured by default to act on SSO authentication, there are several steps. Users would still need to go to the OCA user interface, select SSO authentication, and then request the certificate. (See Chapter 7, "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, Oracle Application Server Certificate Authority has a mechanism to simplify the user experience, by sending users directly to the OCA Certificate Request URL after authentication by the SSO server.

Oracle Application Server Certificate Authority can be configured to provide this URL to the SSO server, for display whenever SSO is not using a certificate to authenticate a user. After SSO authenticates such a user, it then displays the OCA screen enabling that user to request a certificate. After that certificate is created and imported 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.)

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

```
ocactl linksso
```

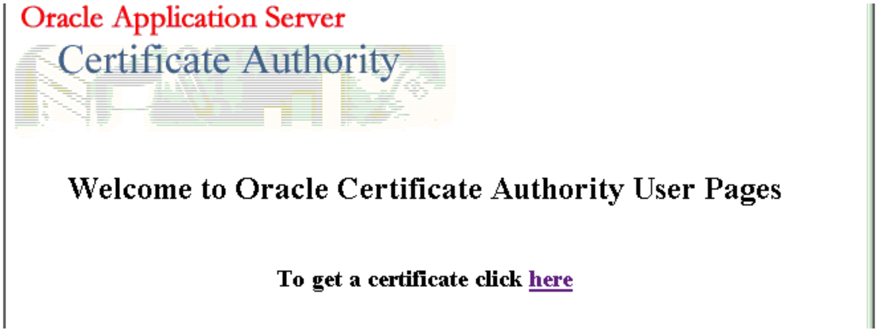
The administrator can also use the ocactl command-line tool (with the administrator password) to cancel the use of this URL through the SSO server, by issuing the following command:

```
ocactl unlinksso
```

Please note that these commands do not require OCA 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 SSO 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 SSO server is restarted, the OCA welcome page will be displayed whenever SSO is not using a certificate to authenticate a user. That page looks like the following illustration:



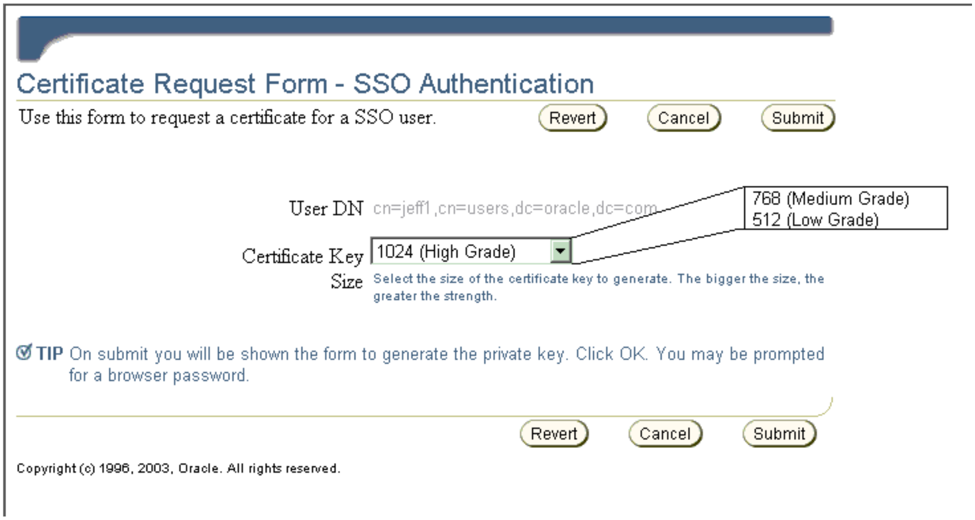
Oracle Application Server

Certificate Authority

Welcome to Oracle Certificate Authority User Pages

To get a certificate click here

When the SSO user clicks that "here" link, the OCA certificate request page appears:



Certificate Request Form - SSO Authentication

Use this form to request a certificate for a SSO user.

User DN cn=jeff1,cn=users,dc=oracle,dc=com

Certificate Key Size: 1024 (High Grade)

768 (Medium Grade)
512 (Low Grade)

Select the size of the certificate key to generate. The bigger the size, the greater the strength.

TIP On submit you will be shown the form to generate the private key. Click OK. You may be prompted for a browser password.

Copyright (c) 1996, 2003, Oracle. All rights reserved.

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 **Import to Browser**, the certificate is imported into the user's browser for automatic use.

Enabling PKI Authentication with SSO and OCA

After being PKI-enabled, the SSO server can use certificates to authenticate users for applications rather than requesting username and password. When a user chooses SSO authentication, the browser asks her to choose a certificate previously imported into the browser, after which the certificate enrollment form will show up directly.

You need to do certain steps to configure SSO to use certificates. These steps are fully described in the *Oracle Application Server Single Sign-On Administrator's Guide*:

1. Enable SSL as described in Chapter 9 of that book. Where there is a choice of Java or PL/SQL, follow the directions in the Java sections.
2. Configure SSO for certificates, as described in Chapter 7 of that book.
3. Re-register OCA's virtual host to the Single Sign-On Server, as explained in the next section.

Re-registering OCA's Virtual Host to SSO Server

Each time the administrator enables the SSO server to use SSL, the OCA virtual host must be re-registered with the SSL-enabled SSO server. All SSO-using applications must do so. Re-registration is done by using the single sign-on registration tool, `ossoreg.jar`. OCA'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*.

Running this tool on the machine hosting the SSO server generates OCA's mod_osso record in the osso.conf file, reflecting SSL settings on the single sign-on server, as follows:

Use a command of the following form (although on a single line) from the `$ORACLE_HOME` where OCA is installed:

```
$ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/sso/lib/ossoreg.jar
-oracle_home_path orcl_home_path
-site_name site_name
-config_mod_osso TRUE
-mod_osso_url mod_osso_url
-u userid
[-virtualhost virtual_host_name]
[-update_mode CREATE | DELETE | MODIFY]
[-config_file config_file_path]
```

```
[-admin_info admin_info]
[-admin_id adminid]
```

Example Suppose that the OCA host name is `myoca.mysite.com` and the `OCA` server authentication port is 4400. 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):

   ```
   setenv ORACLE_HOME /sso_server/oracle_home
   setenv LD_LIBRARY_PATH $ORACLE_HOME/lib
   ```

2. Using these variables as set, the actual command would be as follows (although on a single line):

   ```
   $ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/sso/lib/ossoreg.jar
   -oracle_home_path $ORACLE_HOME -site_name "my_oca_site_name"
   -config_mod_osso TRUE  -mod_osso_url https://myoca.mysite.com:4400
   -u root -config_file $ORACLE_HOME/Apache/Apache/conf/osso/oca/osso.conf
   ```

3. After this command completes, you must restart Oracle HTTP Server by running the following command:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl restartproc type=ohs
   ```

User Certificates and SSO Usage

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

New users can provision their certificates by using the OCA Certificate Request URL for SSO, as described in the sections referenced above.

Once SSO can recognize a user by means of a certificate, she can access applications, including OCA, 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 imports it into the browser, she can thereafter authenticate herself to SSO through PKI.

When the browser of a user presents a certificate to SSO, wanting authentication to use some application, SSO 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.

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 3–2 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 Wallet from Oracle Application Server 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.

Table 3–2 Installation Values for Wallets, CRL, and OHS Port [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 [2]	2048 [2, 3]	3560 days	Default Path Length = 3	Database
CA SSL wallet	cn=<hostname> + CA's DN (except CA's CN)	1024 [4]	730 days		$OH/oca/wallet/ssl [5]
OHS Port for OCA virtual host	--	--	--	4400 and 4401 [6]	$OH/Apache/Apache/conf/ocm_apache.conf [7]
Certificate Revocation List	--	--	One day	--	--

Notes to Table 3–2:

1. For 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.

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 can be done to avoid the warning "CA certificate not trusted" when first connecting to OCA.

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 OCA, include 4402 through 4419.

7. $OH stands for $ORACLE_HOME, so the full location is $ORACLE_HOME/Apache/conf/ocm_apache.conf.

4

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 Oracle Application Server Certificate Authority actions and for implementation of certificate security policies
- Viewing logs of Oracle Application Server Certificate Authority activity

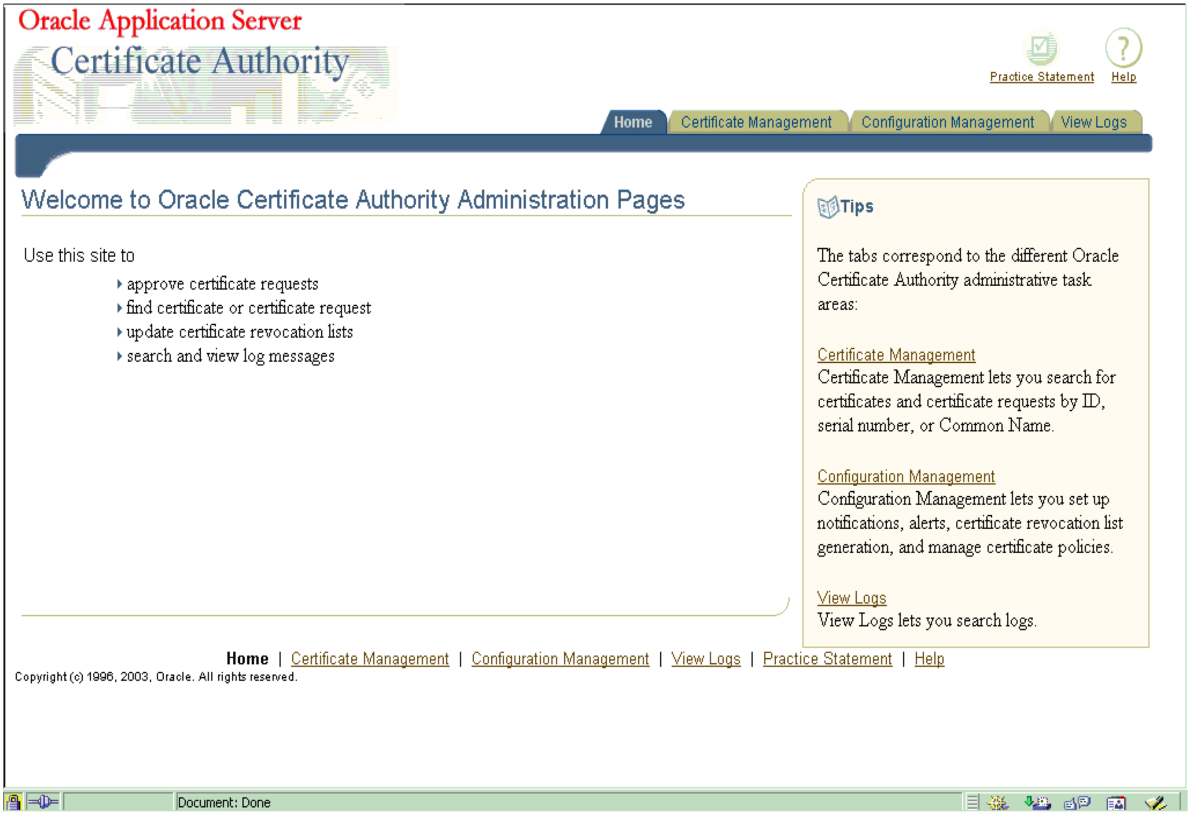
This chapter describes the second and third of those areas: configuration management and viewing logs, as well as describing the content you should provide in your certification practice statement.

It contains the following sections:

- Structure of the Administration Interface
- Configuration Management Tab
- View Logs Tab
- Creating and Updating Your Certification Practice Statement

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:



Welcome to Oracle Certificate Authority Administration Pages

Use this site to

- approve certificate requests
- find certificate or certificate request
- update certificate revocation lists
- search and view log messages

Tips

The tabs correspond to the different Oracle Certificate Authority administrative task areas:

Certificate Management
Certificate Management lets you search for certificates and certificate requests by ID, serial number, or Common Name.

Configuration Management
Configuration Management lets you set up notifications, alerts, certificate revocation list generation, and manage certificate policies.

View Logs
View Logs lets you search logs.

Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

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

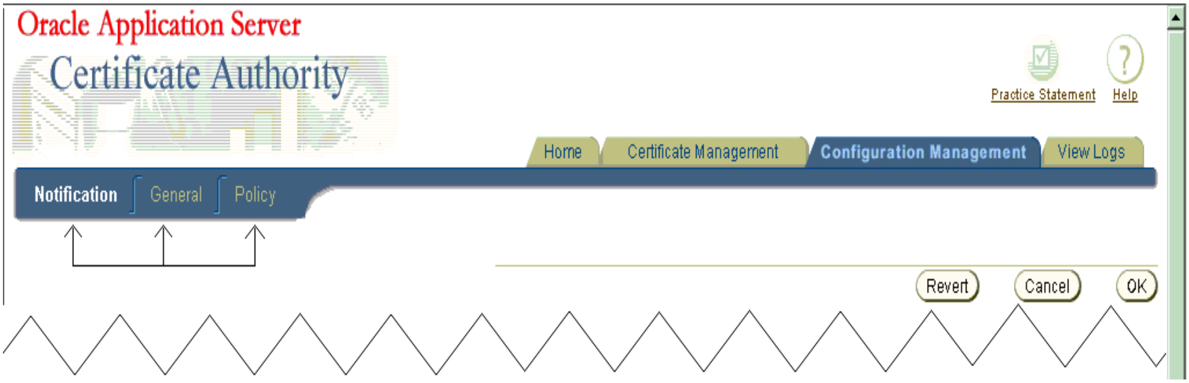
- Certificate Management Tab, described in Chapter 3, particularly in the section entitled Managing Certificates
- Configuration Management Tab, described in this chapter
- View Logs Tab, described in this chapter

Configuration Management Tab

The Configuration management tab is one of the four choices available when you first enter the Oracle Application Server 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 Oracle Application Server 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 5, "Managing Policies in Oracle Application Server Certificate Authority"



Summary of Configuration Tasks

Table 4–1, Table 4–2, and Table 4–3 list the tasks encompassed by the Notification, General, and Policy sub-tabs of Configuration Management and provide links to discussions of those tasks.

Table 4–1 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
Specify the interval between generating CRLs, the interval between validating CRLs, and the interval between directory synchronizations	■ Scheduled Jobs

Table 4–2 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 SSO authentication for certificate management.	■ SSL and SSO Authentication
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 4–3 Policy Sub-tab Tasks and Discussions in Configuration Management

Policy Sub-tab of Oracle Application Server Certificate Authority Tasks and Data (in Chapter 5)	*Links to Task Discussions*
See the policies applicable to available operations, such as certificate requests, revocations, or renewals.	■ Certificate Request Policies as Shipped ■ Certificate Revocation Policy as Shipped ■ Certificate Renewal Policy as Shipped
Edit, enable, disable, delete, add, or reorder policies.	■ Policy Actions

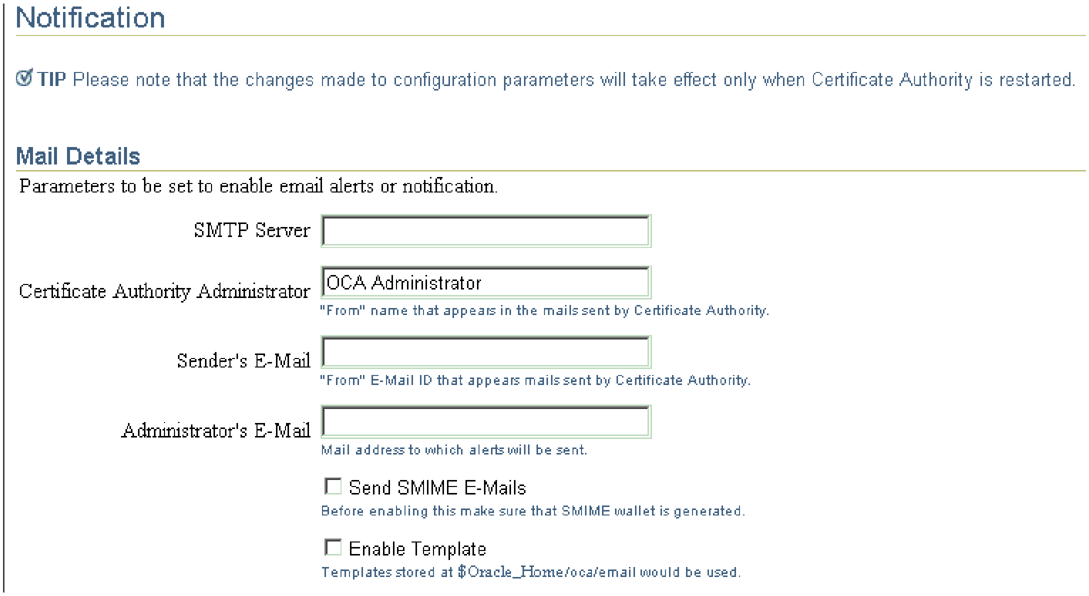
Notification Sub-tab

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 Oracle Application Server Certificate Authority is restarted.

Mail Details

Mail parameters enable email notifications to be sent, encrypted or clear, to the email address you specify for the administrator and to the OCA users when appropriate, using your specified server, sender, and template. You specify your choices in the following portion of the Notification subtab screen:



Notification

TIP Please note that the changes made to configuration parameters will take effect only when Certificate Authority is restarted.

Mail Details

Parameters to be set to enable email alerts or notification.

SMTP Server

Certificate Authority Administrator OCA Administrator
"From" name that appears in the mails sent by Certificate Authority.

Sender's E-Mail
"From" E-Mail ID that appears mails sent by Certificate Authority.

Administrator's E-Mail
Mail address to which alerts will be sent.

☐ Send SMIME E-Mails
Before enabling this make sure that SMIME wallet is generated.

☐ Enable Template
Templates stored at $Oracle_Home/oca/email would be used.

Note that the hint below **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/email."

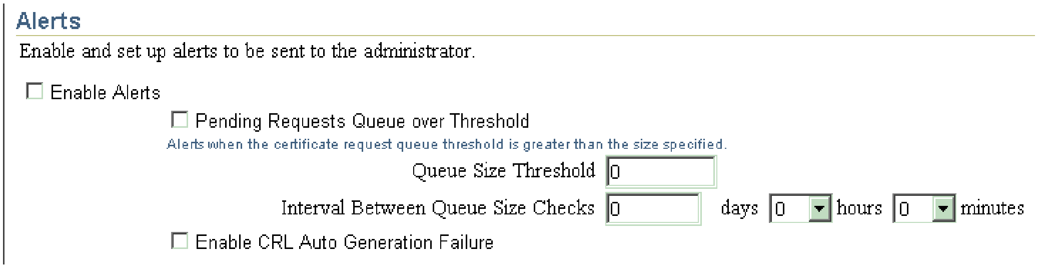
> **See Also:** Regenerating the CA SSL and CA SMIME Wallets in Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics"

Alerts

Alerts parameters enable you to specify whether you are to receive alerts in the following circumstances:

- When the number of pending certificate requests exceeds the queue threshold you specify here, to be checked on the schedule you specify here
- 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, input/output, or connectivity issues.

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



Alerts

Enable and set up alerts to be sent to the administrator.

☐ Enable Alerts

☐ Pending Requests Queue over Threshold

Alerts when the certificate request queue threshold is greater than the size specified.

Queue Size Threshold 0

Interval Between Queue Size Checks 0 days 0 hours 0 minutes

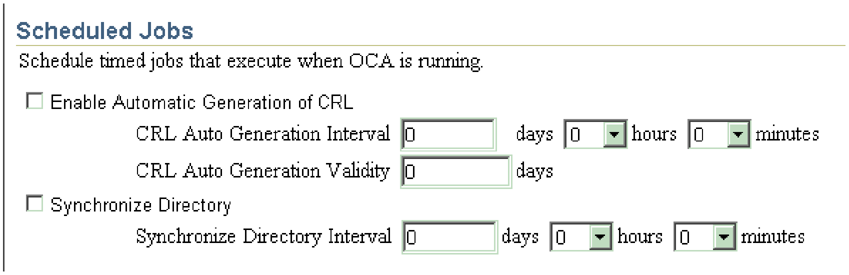
☐ Enable CRL Auto Generation Failure

Scheduled Jobs

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

- Whether a CRL is to be generated automatically, and how often. This feature establishes a reliable, timely, and regular process supporting applications that depend on the CRL to detect revoked or expired certificates.
- Whether directories are to be synchronized, and how often. 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.

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



Scheduled Jobs

Schedule timed jobs that execute when OCA is running.

☐ Enable Automatic Generation of CRL

CRL Auto Generation Interval 0 days 0 hours 0 minutes

CRL Auto Generation Validity 0 days

☐ Synchronize Directory

Synchronize Directory Interval 0 days 0 hours 0 minutes

General Sub-tab

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

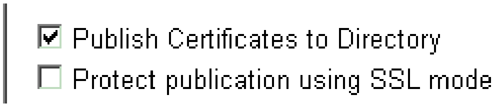
- Certificate Publishing
- SSL and SSO Authentication

- Logging and Tracing
- Default Base DN Components
- Database Settings
- Directory Settings

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

Certificate Publishing

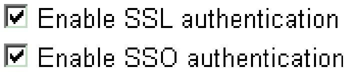
The choices in this section enable you to publish certificates to the directory and to choose SSL protection for messages that tell the directory about those certificates.



☑ Publish Certificates to Directory
□ Protect publication using SSL mode

SSL and SSO Authentication

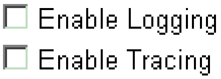
The choices in this section let you specify that SSL or SSO users can be recognized automatically, meaning that their existing certificates (or SSO authentication) are accepted as authenticating their identities. Enabled by default, such acceptance means Oracle Application Server Certificate Authority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



☑ Enable SSL authentication
☑ Enable SSO authentication

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.

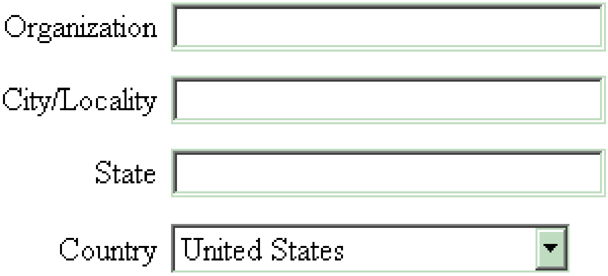


□ Enable Logging
□ Enable Tracing

Logs are stored in the OCA 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.



Organization

City/Locality

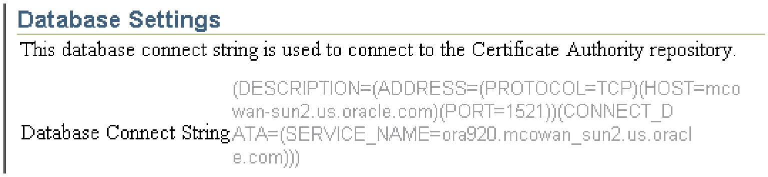
State

Country United States

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 simply tell you the database connect string that is being used to connect to the Oracle Application Server Certificate Authority repository.



Database Settings

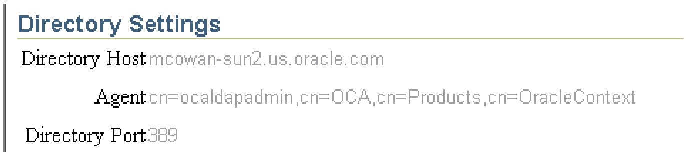
This database connect string is used to connect to the Certificate Authority repository.

Database Connect String (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=mcowan-sun2.us.oracle.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ora920.mcowan_sun2.us.oracle.com)))

These settings only change if Oracle Application Server Certificate Authority's repository moves to a new location. You can use the `ocactl updateconnection` command in that case to update the repository connection settings.

Directory Settings

The settings shown here simply tell you the host, agent, and port being used to connect with Oracle Internet Directory.



Directory Settings

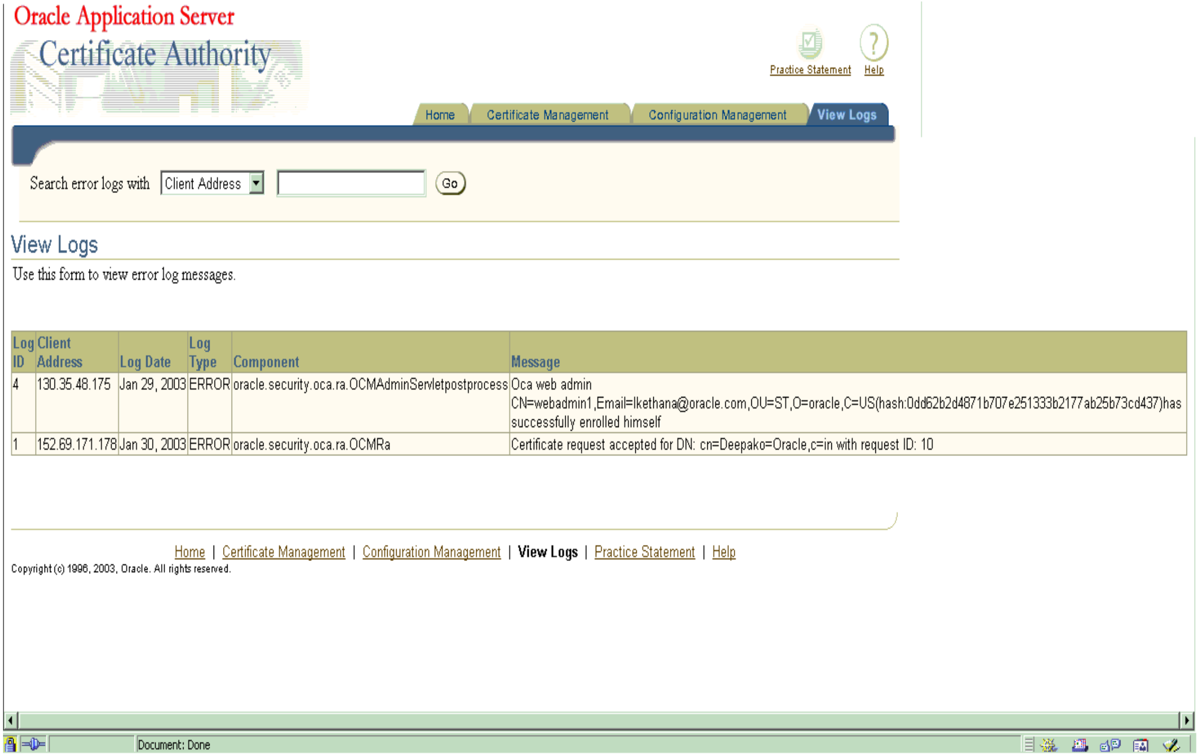
Directory Host mcowan-sun2.us.oracle.com

Agent cn=ocaldapadmin,cn=OCA,cn=Products,cn=OracleContext

Directory Port 389

View Logs Tab

This configuration management page enables you to view logs that record messages regarding transactions or errors occurring during use of Oracle Application Server Certificate Authority. Such a screen would look like this:



Search error logs with Client Address Go

View Logs

Use this form to view error log messages.

Log ID	Client Address	Log Date	Log Type	Component	Message
4	130.35.48.175	Jan 29, 2003	ERROR	oracle.security.oca.ra.OCMAdminServletpostprocess	Oca web admin CN=webadmin1,Email=lkethana@oracle.com,OU=ST,O=oracle,C=US(hash:0dd62b2d4871b707e251333b2177ab25b73cd437)has successfully enrolled himself
1	152.69.171.178	Jan 30, 2003	ERROR	oracle.security.oca.ra.OCMRa	Certificate request accepted for DN: cn=Deepako=Oracle,c=in with request ID: 10

Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

Document: Done

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 Oracle Application Server Certificate Authority generating the entry, and the component's message about the activity.

Creating and Updating Your Certification Practice Statement

A certification practice statement describes the policies and procedures your site and certification authority follow, and thus often contains the following information:

- Legal notices, obligations, and liability
- Warnings or cautionary notes about using certificates
- Public Key Infrastructure knowledge requirements
- Standards or protocols used
- Certificate-specific data:
  - life cycle details
  - limitations
  - key strengths and related security consequences
- Hierarchy of certificate authorities at a site
- Services provided
- How to acquire, revoke, or renew a certificate
- Contact information

You can add or alter your certification practice statement (CPS) by editing the $ORACLE_HOME / oca / help / Help / oca_cps.html file.

After Oracle Application Server Certificate Authority is restarted, your changes will appear on the Practice page when any user clicks the Practice Statement icon appearing on every page.

Note: The Certificate Practice Statement created by the OCA administrator using the above procedure is not internationalization (i18n) compliant. This fact means that clients in a language different from the OCA server language will see the practice statement only in the server's language.

Certificate Practice Statements described by the OCA administrator using the above procedure is not internationalization (i18n) compliant. That means, the clients in a different language than the OCA server language will see the practise statement in server’s language only."

5 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 Oracle Application Server Certificate Authority support standard needs. They are, however, configurable by the administrator, using the Configuration Management tab of the Oracle Application Server 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 the Oracle Application Server Certificate Authority, including the tools for developing custom policy plug-ins.

Topics in this chapter include:

- Definitions
- Overview of Policy Management
- Oracle Application Server Certificate Authority Policies
- Policy Sub-tab of Oracle Application Server Certificate Authority
- Predicates in Policy Rules
- Developing a Custom Policy Plug-in

Definitions

Table 5–1 Policy Concepts, Terms, and Definitions in Oracle Application Server 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, etc. 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.
Predicate	A predicate, in Oracle Application Server 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 OCA 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 algorithm, key size, and validity period.

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

- How Oracle Application Server Certificate Authority (OCA) 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 Oracle Application Server 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. Oracle Application Server 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 Oracle Application Server Certificate Authority package, as described in Developing a Custom Policy Plug-in on page 5-31.

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

Policy rules are enforced by the policy processor module in the Oracle Application Server 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, i.e., 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 Oracle Application Server 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 Oracle Certificate Authority server applies the rule to the certificate request being processed

Table 5–2 lists the default constraint-specific policy rules; the first column contains links to the discussion of each policy rule.

Table 5–2 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 5–3 describes the parameters of the RSA key constraints module.

Table 5–3 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.
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=512	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.

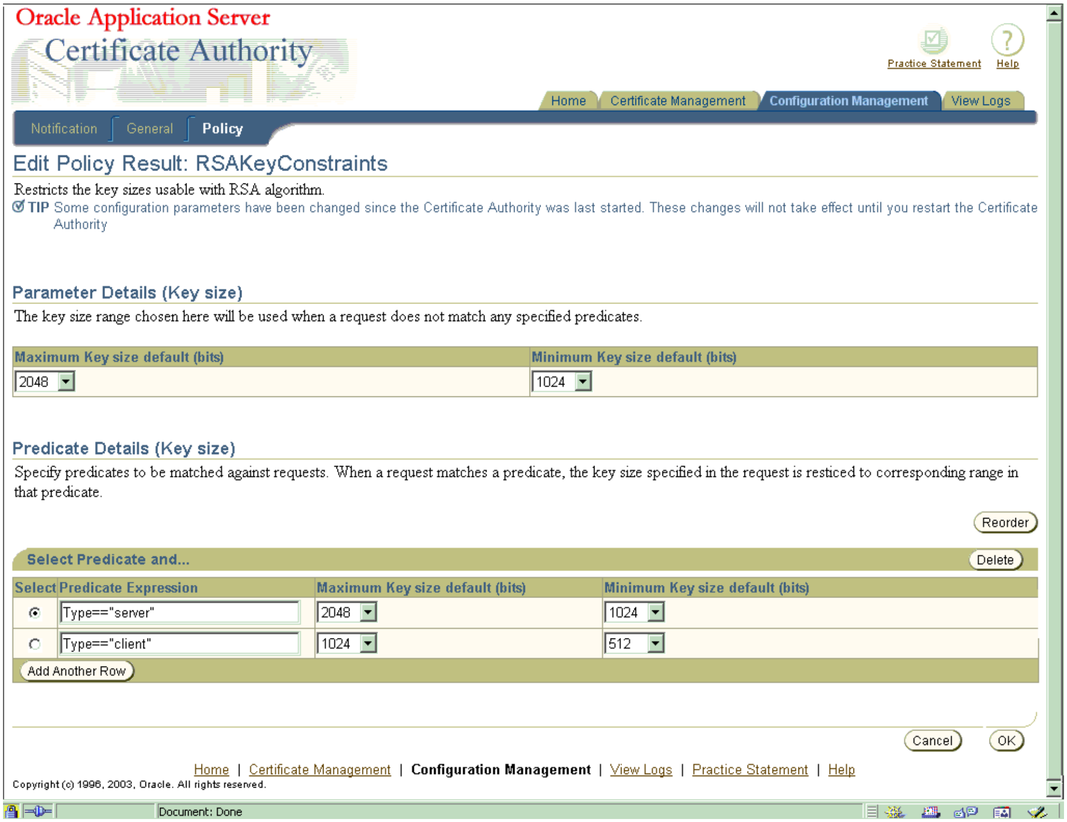
For example, an organization might need (minsize, maxsize) for Sales and Finance departments to be (512,1024) and (1024,2048), respectively. Multiple predicate expressions and value sets can be used to specify this requirement:

Predicate 1: dn=="ou=Sales"

minSize, maxSize are specified as 512,1024

Predicate 2: dn=="ou=Marketing"

minSize, maxSize are specified as 1024,2048



Edit Policy Result: RSAKeyConstraints

Restricts the key sizes usable with RSA algorithm.

TIP Some configuration parameters have been changed since the Certificate Authority was last started. These changes will not take effect until you restart the Certificate Authority

Parameter Details (Key size)

The key size range chosen here will be used when a request does not match any specified predicates.

Maximum Key size default (bits)	Minimum Key size default (bits)
2048	1024

Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="server"	2048	1024
○	Type=="client"	1024	512

Add Another Row
Cancel
OK
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

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 (SSO 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.

Table 5–4 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

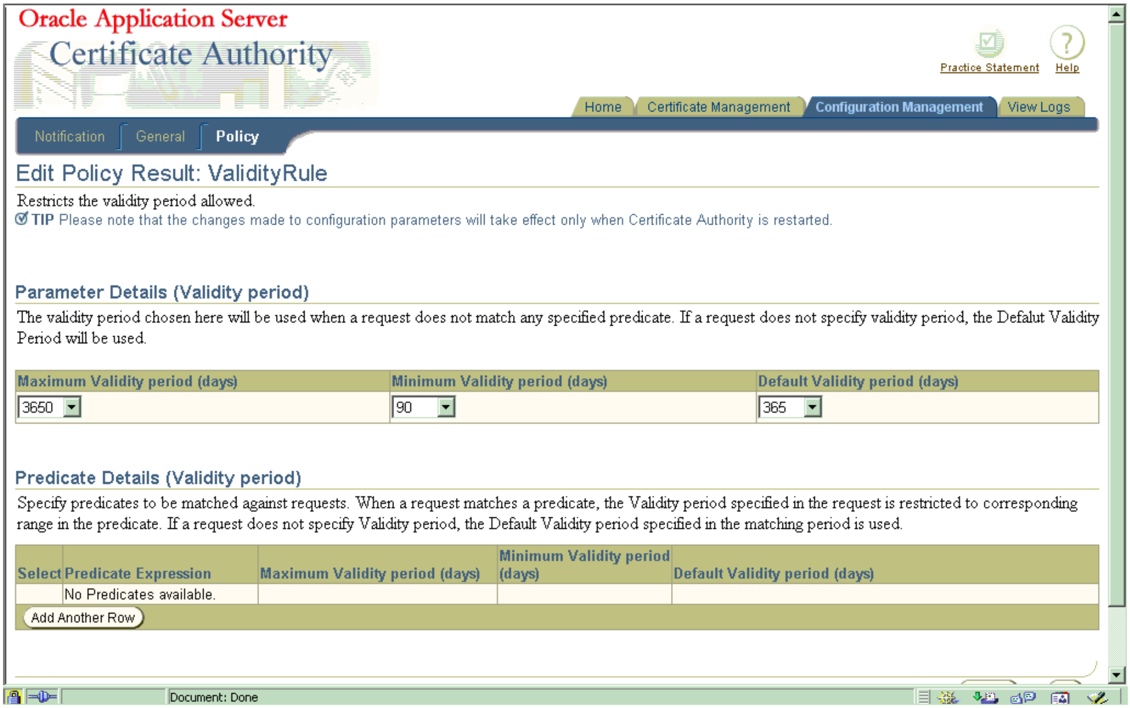
Table 5–4 Parameters in the ValidityRule Policy

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 SSO / 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, Oracle Application Server 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 (i.e., SSO- 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 Oracle Application Server Certificate Authority uses a default of 10 years for the root CA. The following illustration shows the validity-rule parameters.



Edit Policy Result: ValidityRule

Restricts the validity period allowed.

TIP Please note that the changes made to configuration parameters will take effect only when Certificate Authority is restarted.

Parameter Details (Validity period)

The validity period chosen here will be used when a request does not match any specified predicate. If a request does not specify validity period, the Defalut Validity Period will be used.

Maximum Validity period (days)	Minimum Validity period (days)	Default Validity period (days)
3650	90	365

Predicate Details (Validity period)

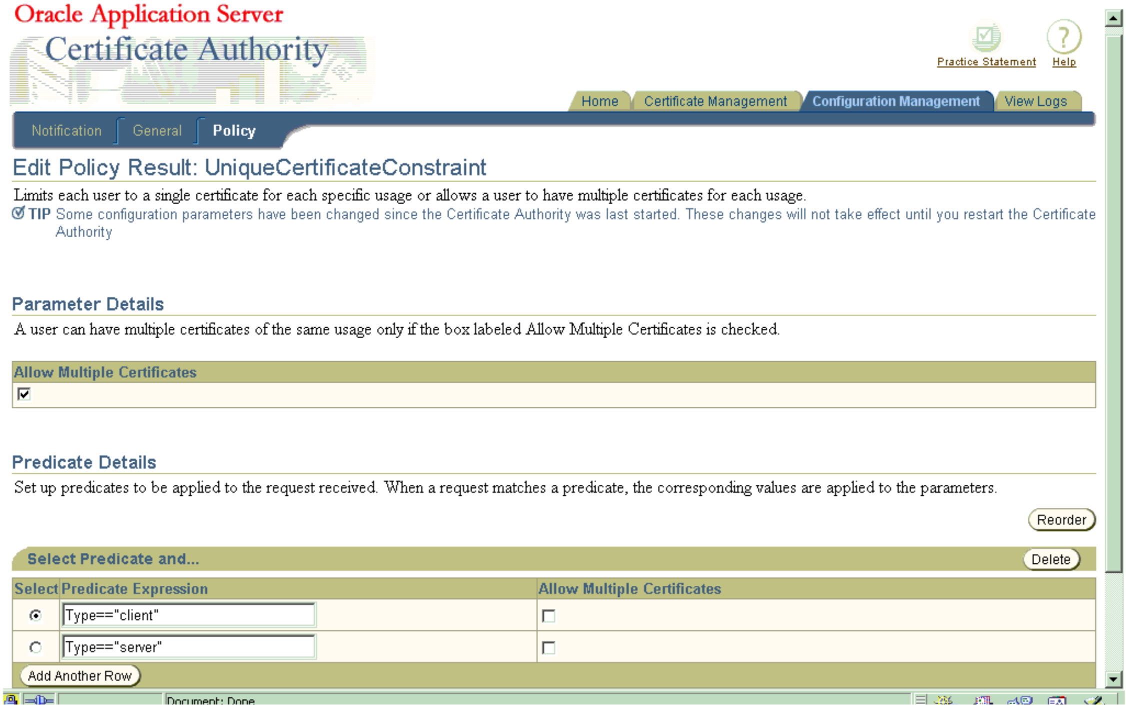
Specify predicates to be matched against requests. When a request matches a predicate, the Validity period specified in the request is restricted to corresponding range in the predicate. If a request does not specify Validity period, the Default Validity period specified in the matching period is used.

Select	Predicate Expression	Maximum Validity period (days)	Minimum Validity period (days)	Default Validity period (days)
	No Predicates available.			

UniqueCertificateConstraint

The UniqueCertificateConstraint policy rule prevents OCA 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 (ssl, signing, etc.) 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.



Edit Policy Result: UniqueCertificateConstraint

Limits each user to a single certificate for each specific usage or allows a user to have multiple certificates for each usage.

TIP Some configuration parameters have been changed since the Certificate Authority was last started. These changes will not take effect until you restart the Certificate Authority

Parameter Details

A user can have multiple certificates of the same usage only if the box labeled Allow Multiple Certificates is checked.

Allow Multiple Certificates

☑

Predicate Details

Set up predicates to be applied to the request received. When a request matches a predicate, the corresponding values are applied to the parameters.

Select Predicate and...

Select	Predicate Expression	Allow Multiple Certificates
●	Type=="client"	□
○	Type=="server"	□

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

Table 5–5 Parameters in the UniqueCertificateConstraint Policy Rule

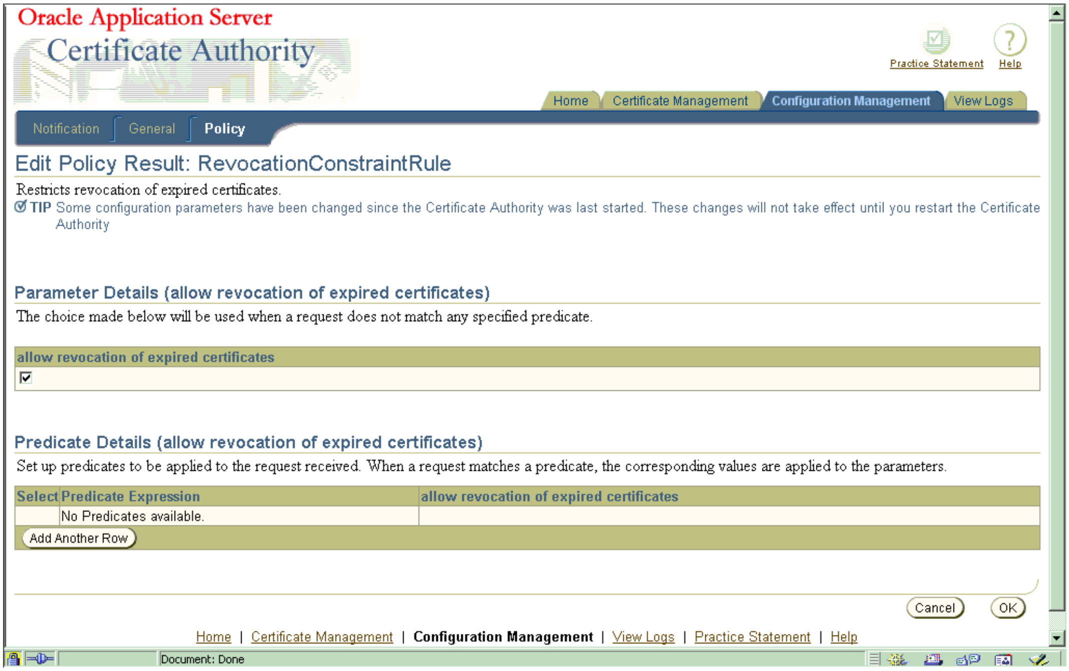
Parameter	Description
Status (Enabled / Disabled)	Specifies (on the Policy Rules page) whether the rule is enabled or disabled.
Default=Enabled	When enabled, the rule uses the checkbox allowing multiple certificates with the same usage. If it prohibits multiple certificates for the same subject name and the same usage, the request is rejected.
	If you disable the rule, OCA will approve multiple certificate requests for the same subject name and the same usage.

Table 5–5 Parameters in the UniqueCertificateConstraint Policy Rule(Cont.)

Parameter	Description
Checkbox allowing multiple certificates to have the same DN and the same usage Default: checked	When checked, this box allows OCA to issue a new certificate for a DN that already has a certificate even if the usages are same. When unchecked, this box prevents OCA 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 OCA 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.



Edit Policy Result: RevocationConstraintRule

Restricts revocation of expired certificates.

TIP Some configuration parameters have been changed since the Certificate Authority was last started. These changes will not take effect until you restart the Certificate Authority

Parameter Details (allow revocation of expired certificates)

The choice made below will be used when a request does not match any specified predicate.

allow revocation of expired certificates
☑

Predicate Details (allow revocation of expired certificates)

Set up predicates to be applied to the request received. When a request matches a predicate, the corresponding values are applied to the parameters.

Select	Predicate Expression	allow revocation of expired certificates
	No Predicates available.	

Add Another Row
Cancel
OK
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help
Document: Done

Table 5–6 describes the parameters of the revocation constraints module.

Table 5–6 Parameters in the Revocation 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 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, OCA does not verify the validity period of the certificate being revoked, nor whether it is expired. The certificate is simply revoked.
allowExpiredCerts Default: True	Specifies whether to allow (True) or prevent (False) revocation of expired certificates.

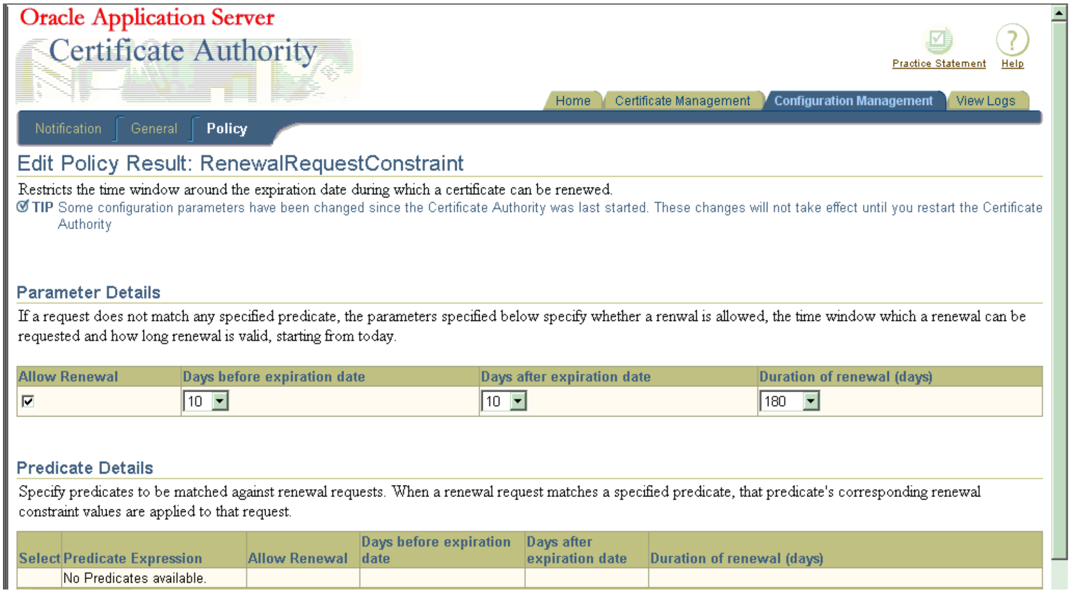
RenewalRequestConstraint

The OCA 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 5–7 describes the parameters of the renewal constraints policy rule.

Table 5–7 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 OCA 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, e.g., 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.) 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.



Edit Policy Result: RenewalRequestConstraint

Restricts the time window around the expiration date during which a certificate can be renewed.

TIP Some configuration parameters have been changed since the Certificate Authority was last started. These changes will not take effect until you restart the Certificate Authority

Parameter Details

If a request does not match any specified predicate, the parameters specified below specify whether a renwal is allowed, the time window which a renewal can be requested and how long renewal is valid, starting from today.

Allow Renewal	Days before expiration date	Days after expiration date	Duration of renewal (days)
☑	10	10	180

Predicate Details

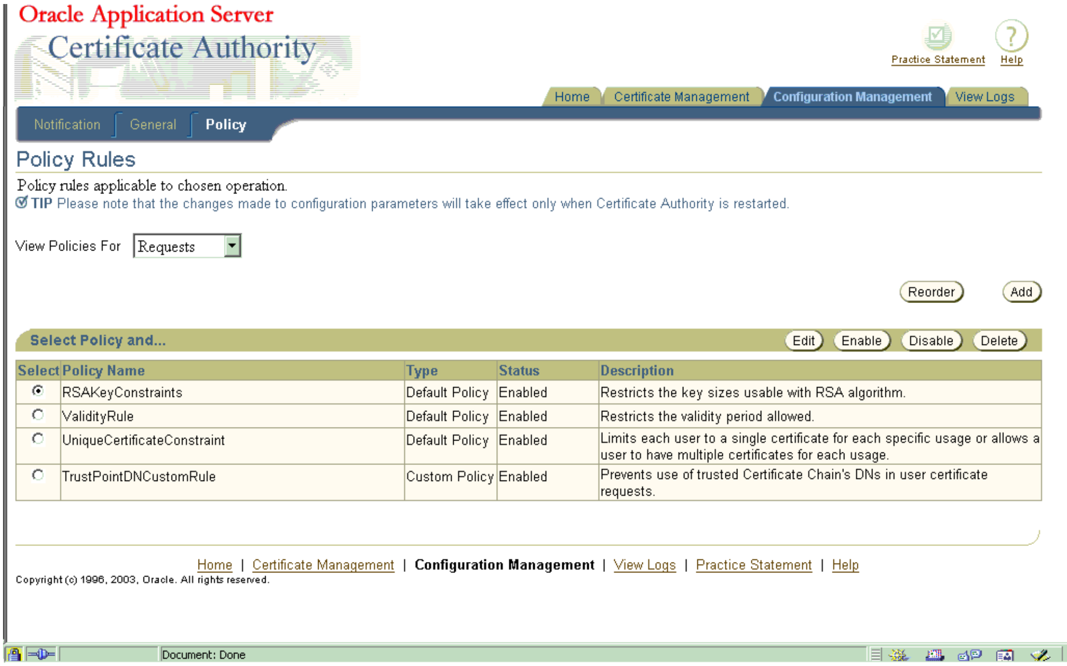
Specify predicates to be matched against renewal requests. When a renewal request matches a specified predicate, that predicate's corresponding renewal constraint values are applied to that request.

Select	Predicate Expression	Allow Renewal	Days before expiration date	Days after expiration date	Duration of renewal (days)
	No Predicates available.				

All Oracle Application Server Certificate Authority policies are managed by the OCA 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.



Policy Rules

Policy rules applicable to chosen operation.

TIP Please note that the changes made to configuration parameters will take effect only when Certificate Authority is restarted.

View Policies For: Requests

Select Policy and...

Select	Policy Name	Type	Status	Description
●	RSAKeyConstraints	Default Policy	Enabled	Restricts the key sizes usable with RSA algorithm.
○	ValidityRule	Default Policy	Enabled	Restricts the validity period allowed.
○	UniqueCertificateConstraint	Default Policy	Enabled	Limits each user to a single certificate for each specific usage or allows a user to have multiple certificates for each usage.
○	TrustPointDNCustomRule	Custom Policy	Enabled	Prevents use of trusted Certificate Chain's DNs in user certificate requests.

Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

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 Oracle Application Server Certificate Authority, and the actions available to the administrator, are summarized in the following sections:

- Certificate Request Policies as Shipped
- Certificate Revocation Policy as Shipped
- 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 OCA 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 OCA 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 OCA repository. Enabling a disabled policy (or one that was specified in the OCA 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 Oracle Application Server Certificate Authority is restarted, as described in the section titled Starting and Stopping Oracle Application Server Certificate Authority in .

Oracle Application Server 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.

Certificate Request Policies as Shipped

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 issues in the policies as shipped:

- 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 per type of usage, that is, for signing, key encipherment, or data encipherment, respectively designated SSL, CA, or SMIME
- Allow or disallow the use of trusted-certificate-DNs as certificate applicants or owners.

Certificate Revocation Policy as Shipped

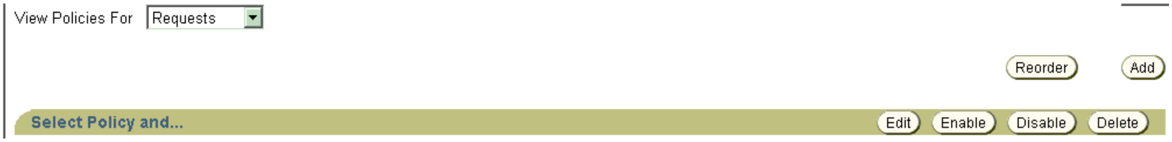
The RevocationConstraintRule is an OCA default policy shipped with Oracle Application Server Certificate Authority. You can set parameters and predicates on this policy as required, such as to allow or disallow revocation of expired certificates.

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.

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.



View Policies For: Requests

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.

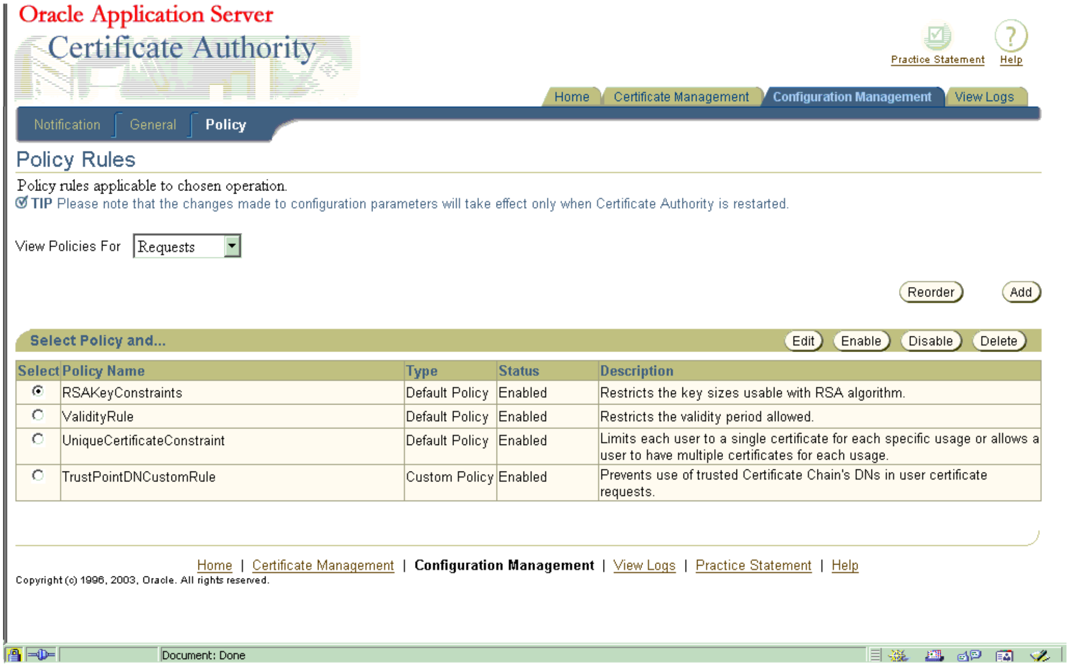
Delete

On the Policy Rules page, the default Oracle Application Server 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**. Oracle Application Server 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:



Policy Rules

Policy rules applicable to chosen operation.

TIP Please note that the changes made to configuration parameters will take effect only when Certificate Authority is restarted.

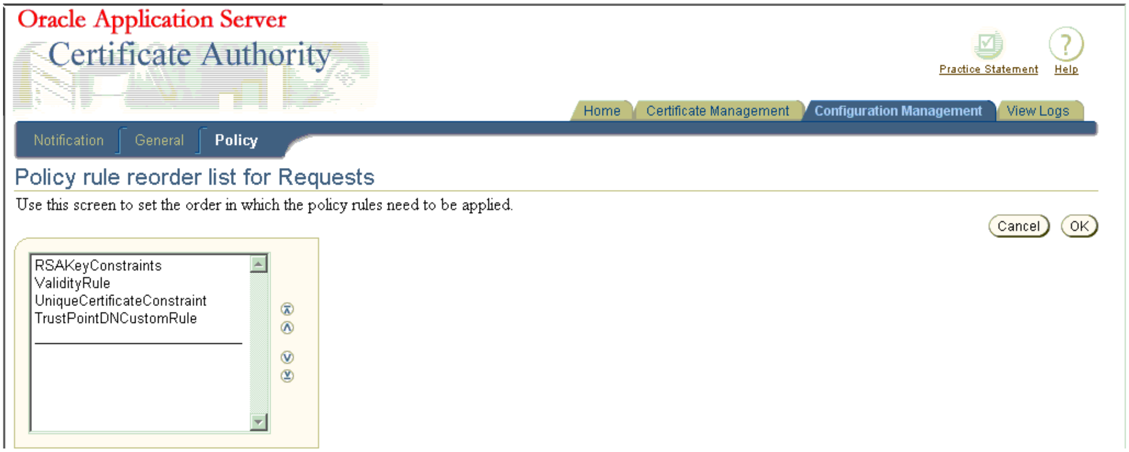
View Policies For: Requests

Select	Policy Name	Type	Status	Description
●	RSAKeyConstraints	Default Policy	Enabled	Restricts the key sizes usable with RSA algorithm.
○	ValidityRule	Default Policy	Enabled	Restricts the validity period allowed.
○	UniqueCertificateConstraint	Default Policy	Enabled	Limits each user to a single certificate for each specific usage or allows a user to have multiple certificates for each usage.
○	TrustPointDNCustomRule	Custom Policy	Enabled	Prevents use of trusted Certificate Chain's DNs in user certificate requests.

Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

When you click **Reorder**, Oracle Application Server 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:

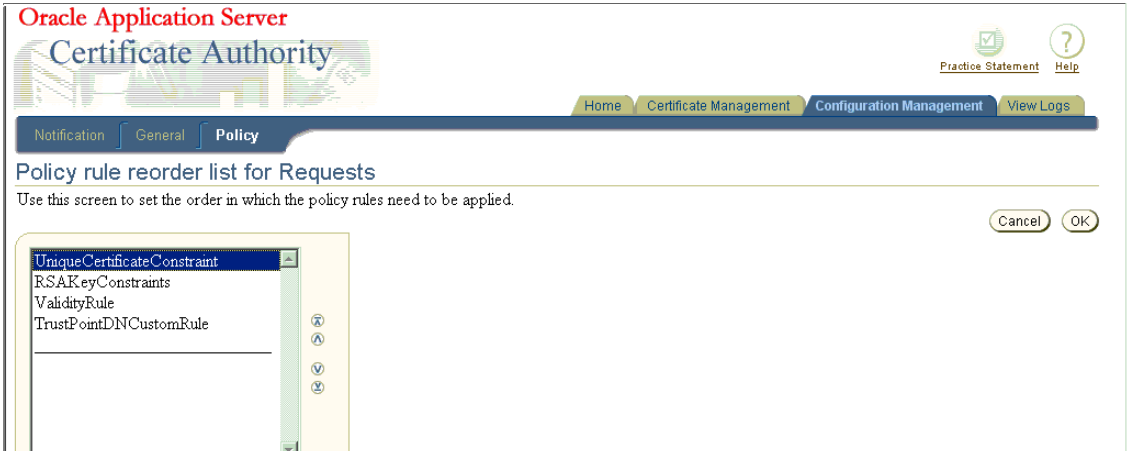


Policy rule reorder list for Requests

Use this screen to set the order in which the policy rules need to be applied.

RSAKeyConstraints
ValidityRule
UniqueCertificateConstraint
TrustPointDNCustomRule

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:

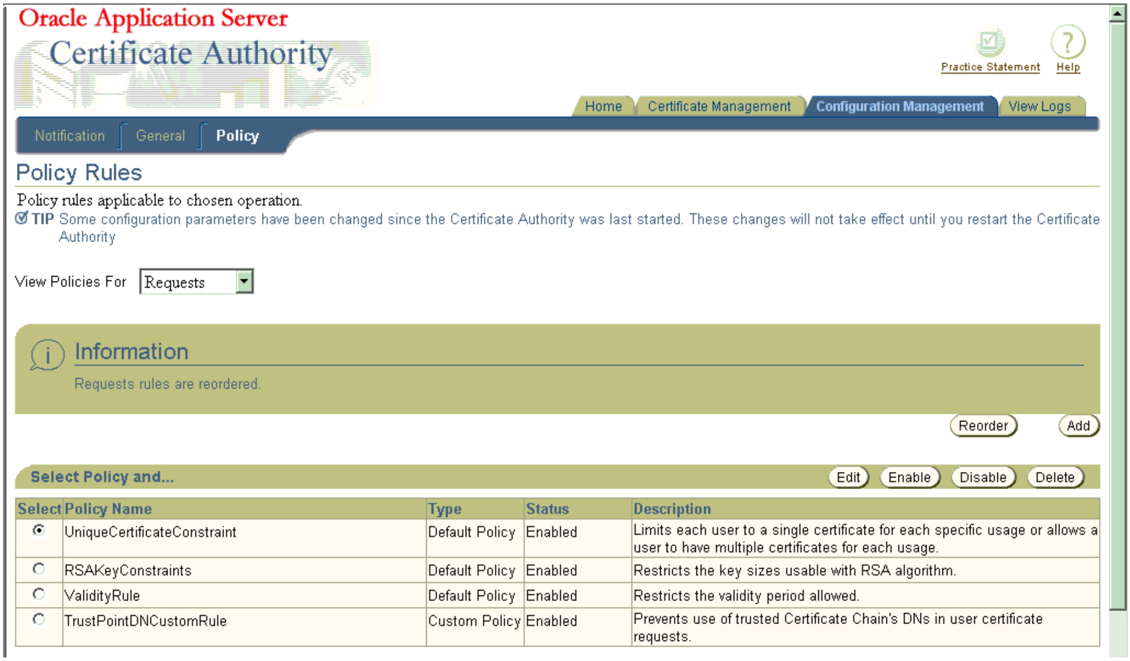


Policy rule reorder list for Requests

Use this screen to set the order in which the policy rules need to be applied.

UniqueCertificateConstraint
RSAKeyConstraints
ValidityRule
TrustPointDNCustomRule

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



Policy Rules

Policy rules applicable to chosen operation.

TIP Some configuration parameters have been changed since the Certificate Authority was last started. These changes will not take effect until you restart the Certificate Authority

View Policies For Requests

Information

Requests rules are reordered.

Select Policy and...

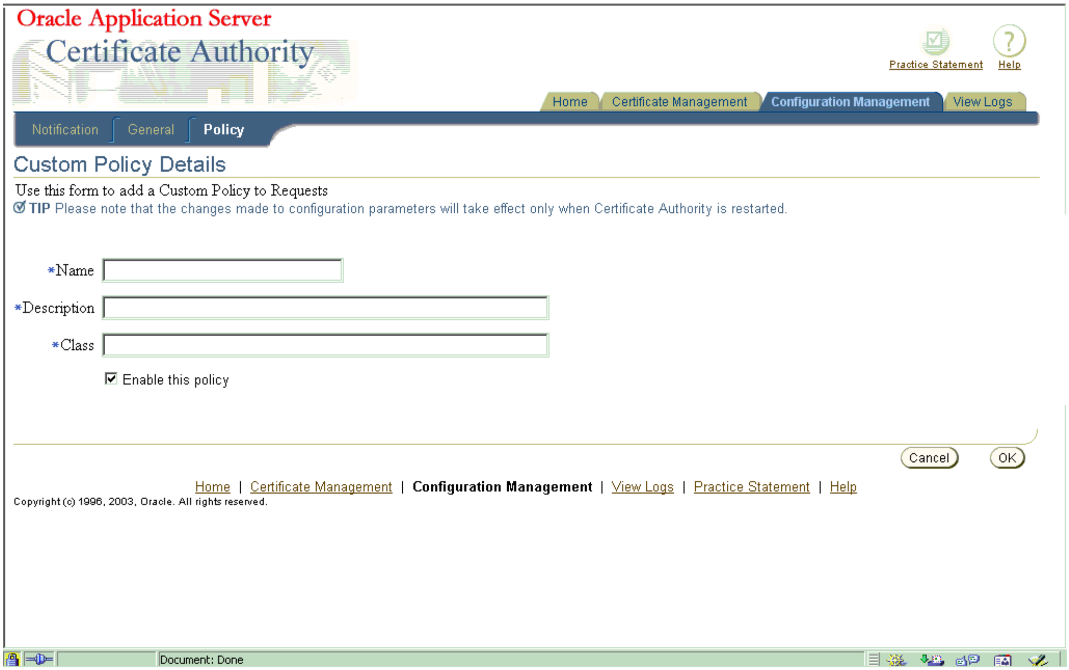
Select	Policy Name	Type	Status	Description
●	UniqueCertificateConstraint	Default Policy	Enabled	Limits each user to a single certificate for each specific usage or allows a user to have multiple certificates for each usage.
○	RSAKeyConstraints	Default Policy	Enabled	Restricts the key sizes usable with RSA algorithm.
○	ValidityRule	Default Policy	Enabled	Restricts the validity period allowed.
○	TrustPointDNCustomRule	Custom Policy	Enabled	Prevents use of trusted Certificate Chain's DNs in user certificate requests.

Note that Oracle Application Server 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, i.e., 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. Oracle Application Server 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.



Custom Policy Details

Use this form to add a Custom Policy to Requests

TIP Please note that the changes made to configuration parameters will take effect only when Certificate Authority is restarted.

*Name

*Description

*Class

☑ Enable this policy

Cancel
OK
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

See Developing a Custom Policy Plug-in on page 5-31 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 subsections:

- Multiple Predicate Evaluation
  - 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, i.e., 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, i.e., 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 that 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 5–8 lists the logical operators used in predicate expressions.

Table 5–8 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
```

Oracle Application Server 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, e.g., type=="*" matches all the certificate types. However, using "*" with any other string to form partial-pattern string matching is not supported.

Table 5–9 describes the attributes and the values they can have.

Table 5–9 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 the type of certificate usage. Allowable values including the following: ■ usage=="ssl" ■ usage=="smime_enc" ■ usage=="smime_sign" ■ usage=="code_sign" ■ usage=="ca_sign"
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.)

Oracle Application Server 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.

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 above 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 where as, "ou=ST,o=Oracle" is an invalid partial DN as it stops at "o=Oracle" and doesn't contain the root (i.e. "c=US").

To support the big-endian order, where the most significant component is first, OCA 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"

The above 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

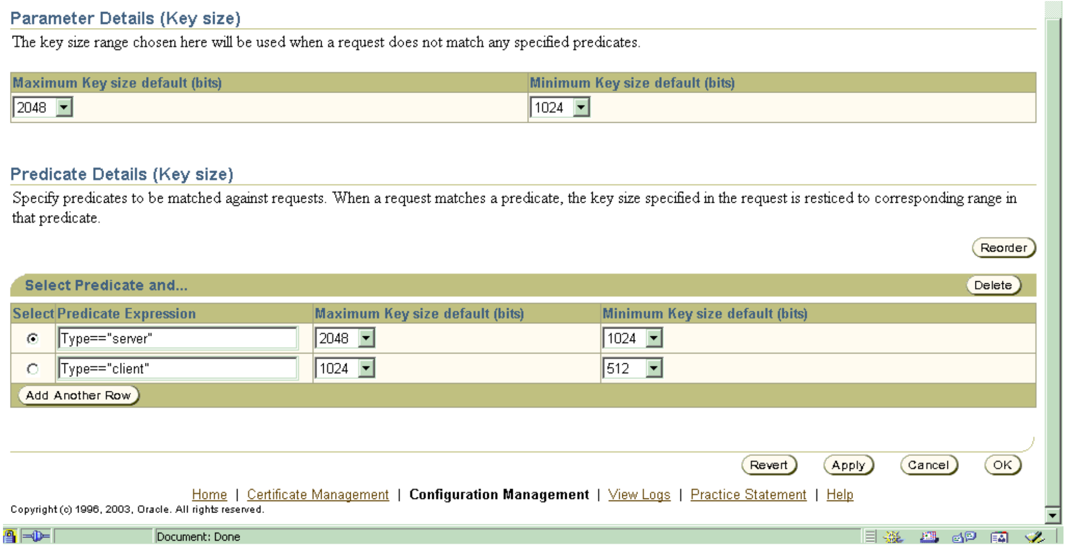
A policy rule can have more than one predicate. When the policy rule has multiple predicates, evaluation begins by comparing the first predicate expression against the incoming certificate request object. If it matches, the rule is applied. If not, then evaluation compares the next predicate expression 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.



Parameter Details (Key size)

The key size range chosen here will be used when a request does not match any specified predicates.

Maximum Key size default (bits)	Minimum Key size default (bits)
2048	1024

Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

Select Predicate and...

Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="server"	2048	1024
○	Type=="client"	1024	512

Home | Certificate Management | **Configuration Management** | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

Document: Done

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.

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.

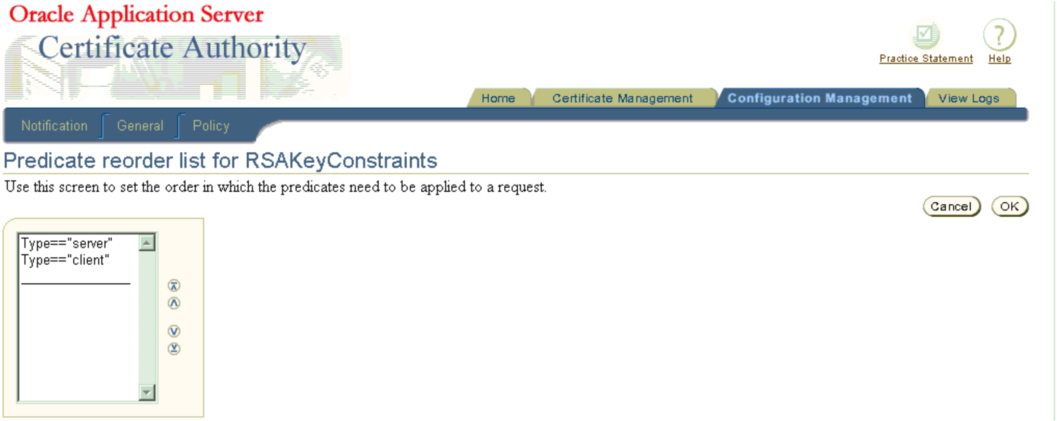
Then, suppose a later predicate in that same policy specifies `Type=="client"` and `OU=="Oracle"`, setting the keylength to 512.

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 5-17. 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:

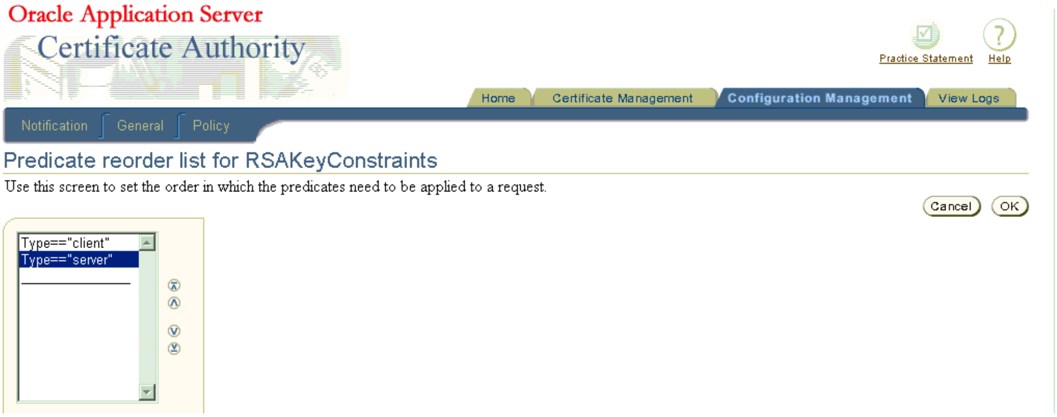


Predicate reorder list for RSAKeyConstraints

Use this screen to set the order in which the predicates need to be applied to a request.

Type=="server"
Type=="client"

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:

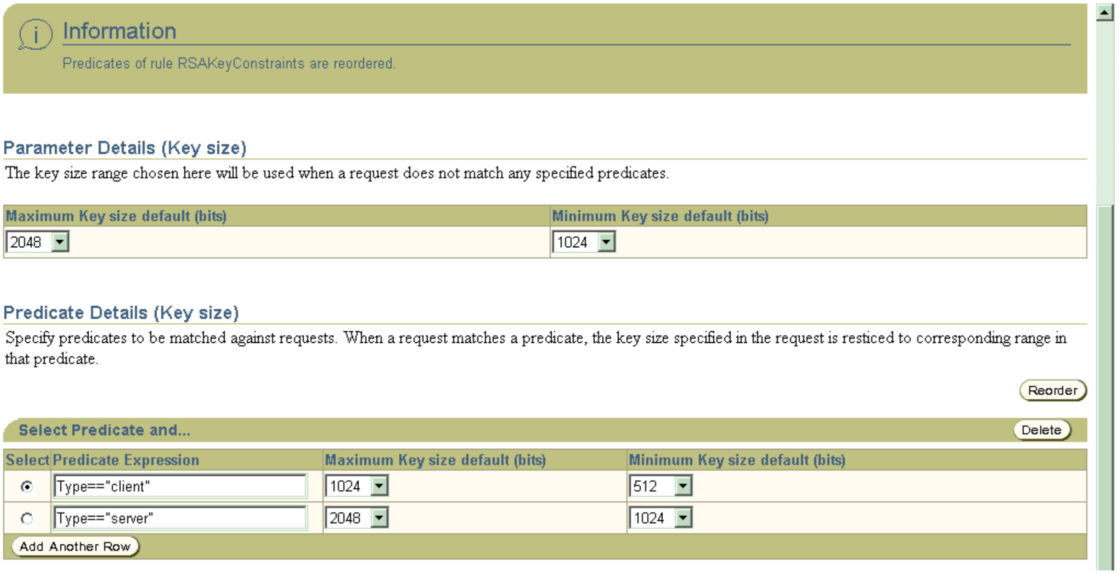


Predicate reorder list for RSAKeyConstraints

Use this screen to set the order in which the predicates need to be applied to a request.

Type=="client"
Type=="server"

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



Information

Predicates of rule RSAKeyConstraints are reordered.

Parameter Details (Key size)

The key size range chosen here will be used when a request does not match any specified predicates.

Maximum Key size default (bits)	Minimum Key size default (bits)
2048	1024

Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

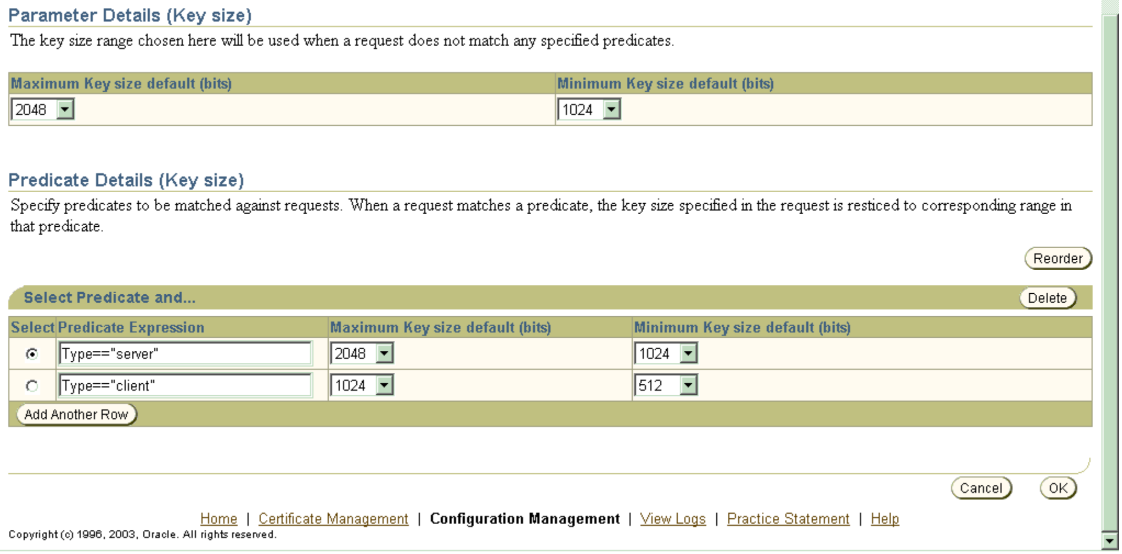
Select Predicate and...

Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="client"	1024	512
○	Type=="server"	2048	1024

Notice that Oracle Application Server 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. Oracle Application Server Certificate Authority displays a blank row for you to fill:



Parameter Details (Key size)

The key size range chosen here will be used when a request does not match any specified predicates.

Maximum Key size default (bits)	Minimum Key size default (bits)
2048 ▾	1024 ▾

Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

Reorder

Select Predicate and... Delete

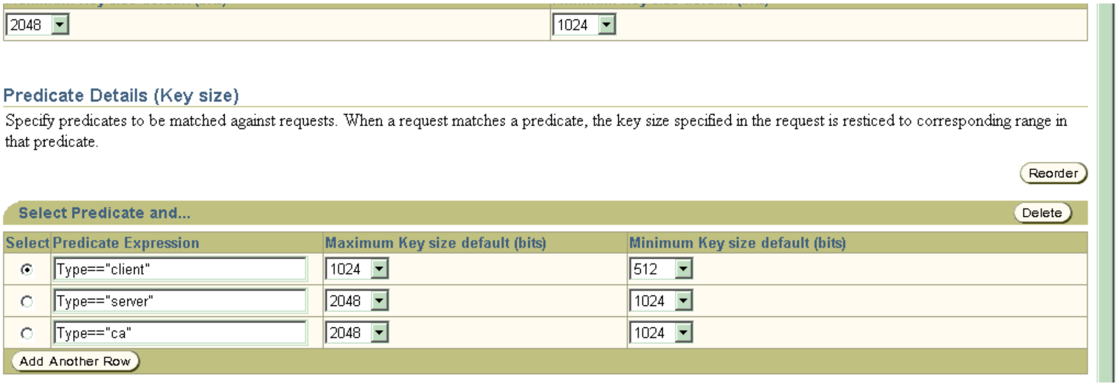
Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="server"	2048 ▾	1024 ▾
○	Type=="client"	1024 ▾	512 ▾

Add Another Row

Cancel OK

Home | Certificate Management | **Configuration Management** | View Logs | Practice Statement | Help

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



2048 ▾	1024 ▾

Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

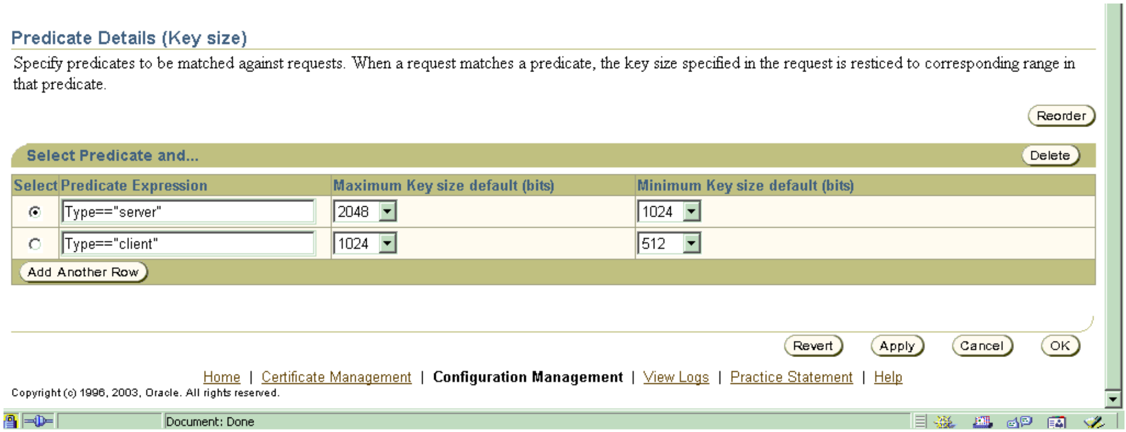
Reorder

Select Predicate and... Delete

Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="client"	1024 ▾	512 ▾
○	Type=="server"	2048 ▾	1024 ▾
○	Type=="ca"	2048 ▾	1024 ▾

Add Another Row

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:



Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

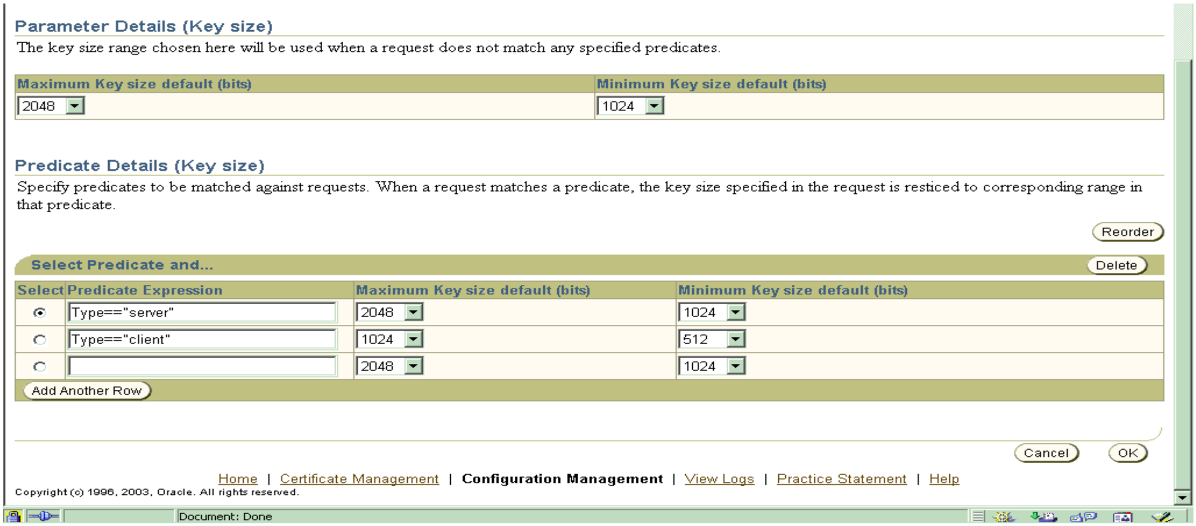
Reorder
Select Predicate and...
Delete

Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="server"	2048	1024
○	Type=="client"	1024	512

Add Another Row
Revert
Apply
Cancel
OK
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

When you click **Add Another Row**, Oracle Application Server 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:



Parameter Details (Key size)

The key size range chosen here will be used when a request does not match any specified predicates.

Maximum Key size default (bits)	Minimum Key size default (bits)
2048	1024

Predicate Details (Key size)

Specify predicates to be matched against requests. When a request matches a predicate, the key size specified in the request is resticed to corresponding range in that predicate.

Select	Predicate Expression	Maximum Key size default (bits)	Minimum Key size default (bits)
●	Type=="server"	2048	1024
○	Type=="client"	1024	512
○		2048	1024

Add Another Row
Cancel
OK
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

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**. Oracle Application Server Certificate Authority 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 OCA are generic. To enhance the policy structure for specific organizational requirements, an administrator can write a custom plug-in using the framework OCA 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 OCA, 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 OCA 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 5–10 Steps in Custom Policy Plug-in Processing

Step	Results
Enforce method of OCA 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, etc.
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

The following four steps describe creating a new policy plug-in:

1. Write a Java class that implements 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 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'ed and kept in example.jar.

      To jar the class, use the jar utility available under $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 oca 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 OCA, OCA's OC4J, and OHS. Use the following commands in ORACLE_HOME:

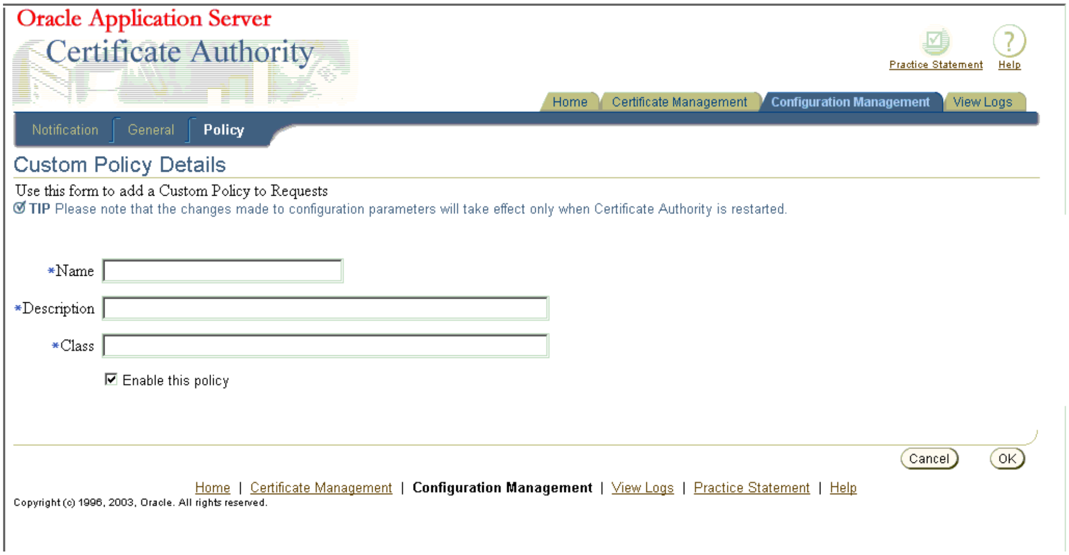
   ```
   $ORACLE_HOME/oca/bin/ocactl stop
   $ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=oca
   $ORACLE_HOME/opmn/bin/opmnctl stopproc type=ohs
   ```

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

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

7. Use the OCA 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.



Custom Policy Details

Use this form to add a Custom Policy to Requests

TIP Please note that the changes made to configuration parameters will take effect only when Certificate Authority is restarted.

*Name

*Description

*Class

☑ Enable this policy

Cancel
OK
Home | Certificate Management | Configuration Management | View Logs | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

8. Restart OCA, as described in the section titled Starting and Stopping Oracle Application Server Certificate Authority in Chapter 3, "Introduction to OCA

Administration and Certificate Management". Only after OCA 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.

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.

```
package oca.custom.policy.plugin;
import oracle.security.oca.policy.custom.OCACustomPolicyplugin;
import oracle.security.oca.policy.OCAPolicyRequest;
public class examplePlugin implements OCACustomPolicyPlugin
{
    public boolean enforce (OCAPolicyRequest policyRequest)
   {
   // Add the functionality here.
   // Assume the plug-in should reject all requests with country code as US.
         if (!policyRequest.getCountry().equals("US"))
         {
              //Plug-in check succeeded. Country ID in request is not US.
              //Hence return true.
              return true;
          }
          else
          {
```

```
                    //Plug-in check failed: Country ID in request is US. Set error and
return false.

13:
policyRequest.setError("examplePlugin",OCMPolicyConstants.POLICY_ERROR,

                                    "Country ID cannot be US.");

                    //The first parameter is the plug-in name.

                    //The second parameter is the status, which is an ERROR.

                    //The third parameter is the Message to be displayed.

14:                   return false;

15:             }

16:       }

17:   }
```

In the example above, 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 and 3 import the class files required. The Javadoc API documentation contains the details of these files.

Line 4 implements the OCACustomPolicyPlugin interface. The custom policy interface must be implemented by all custom plug-ins. The interface that OCA gives belongs to the package oracle.security.oca.policy.custom and is available in $ORACLE_HOME/oca/lib/oca-1_3.jar.

Line 6 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 7 checks if the country code is not US. The API documentation accompanying OCA contains the details of the methods that can be used on policyRequest.

Line 8 returns success to the policy processor.

Line 11 introduces the handling for the error condition, exercised when the country code in the request is US.

Line 13 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 SSO User certificate and immediately try to renew this certificate. The Renewal plug-in will throw an error.

Line 14 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 OCA, 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 above mentioned custom policy example, if line 13 is changed to

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

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

See also: The Javadoc provided with the other documentation for descriptions of the classes and methods provided in OCA Custom plug-in, and the constants available for your use.

Note: The generic error messages supported by OCA are translated into the languages supported by OCA, 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 OCA.

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.

6

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
- Configuration Operations for OracleAS Certificate Authority
- Customization Support
- Log or Trace OCA Actions for Oracle Application Server Certificate Authority
- Changing the Infrastructure Services That OCA Uses
- OracleAS Certificate Authority and High-Availability Features
- OracleAS Certificate Authority Backup and Recovery Considerations
- Restricting the Realm of Certificate Publication

Wallet Operations for OracleAS Certificate Authority

Wallets are containers for certificates and trusted authorities' certificates. Oracle Application Server 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
- Regenerating the CA SSL and CA SMIME Wallets

- Renewing Critical Wallets
- Changing Passwords

Regenerating the CA Signing Wallet

Note: You need to be extremely cautious before this operation is attempted. Because this operation regenerates the CA signing certificate, it replaces the existing CA certificate and invalidates all the certificates issued by the existing CA.

Installation of Oracle Application Server Certificate Authority as a root certificate authority (CA) also creates the CA signing certificate, CA SSL wallet, and CA SMIME wallet. If the CA key is somehow compromised, these wallets can be regenerated using the `ocactl` command line tool, as described in the next section.

The new CA certificate and private key will be stored in the OCA 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. After regeneration of the CA wallet, a CRL issued by the old CA will not be useful.

Example of the command to generate the CA wallet:

```
ocactl generatewallet -type CA
```

OCA 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 3, "Introduction to OCA Administration and Certificate Management".

Note: Domain component entries in the DN can be used in addition to (or to replace) entries for organization or country. Examples include dc=be (for belgium) or dc=us (for United States) or dc=oracle or dc=com. See also the section entitled Domain Component Attributes in Appendix E, "Glossary".

Regenerating the CA SSL and CA SMIME Wallets

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 SMIME wallets in order to establish secure communications with the OCA server. These circumstances include a wallet becoming compromised or corrupted, or the CA wallet being regenerated, or a new Sub CA certificate being imported.

Example of the command to generate the CA SSL wallet:

```
ocactl generatewallet -type CASSL
```

OCA, OCA's OC4J, and OHS all need to be stopped to execute this command. After this command executes, restart OHS, OCA's OC4J, and OCA, 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 SSO 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 SSO-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 OCA instance in OPMN will use this wallet for SSL server authentication.

The CA SMIME Wallet

The CA SMIME wallet is used to enable the Oracle Application Server Certificate Authority to sign alerts and notification messages. This wallet will be used only when "Send SMIME E-Mails" is enabled in Notification page of Configuration Management in OCA Admin page.

> **See Also:** Mail Details in Chapter 4, "Configuring Oracle Application Server Certificate Authority"

If this SMIME wallet is compromised or corrupted, or when the CA wallet is regenerated, you must regenerate the CA SMIME wallet. This wallet is encrypted by the password that was provided during its generation.

Example of the command to generate CA SMIME wallet:

```
ocactl generatewallet -type CASMIME
```

The following steps generate and use the CASMIME wallet:

1. Stop OCA. Use the command

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

2. Generate the CA SMIME wallet using above command.
3. If you have not already done so, go to the Notification page of Configuration Management in the OCA web interface and enable the "Send SMIME E-Mails" option. It uses the generated CA SMIME wallet to sign alerts and notifications.
4. Start OCA. Use the command

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

After regeneration of the CA SMIME wallet, the old CA SMIME will not be of any use. The new CA SMIME wallet is used to sign alert and notification messages.

Renewing Critical Wallets

When a certificate is going to expire, renewal will be required. CA, CA SSL, and CASMIME wallets can be renewed using the `ocactl` 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 OCA’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 SMIME wallet is renewed, the new wallet overwrites the old CA SMIME wallet at $ORACLE_HOME/oca/wallet/email.

Example to renew CA wallet:

```
ocactl renewcert -type CA
```

Renewed wallets take effect only after OHS, OCA's OC4J, and OCA are restarted, in that order, as described in the section titled Starting and Stopping Oracle Application Server Certificate Authority in Chapter 3, "Introduction to OCA Administration and Certificate Management".

Changing Passwords

After installation, you can change any of the following passwords: CA, CA SSL, CA SMIME, or DB. With the exception of the database password (DB), all other passwords can be changed even when Oracle Application Server Certificate Authority is in operation. The possibility of active connections to the CA, using the existing DB password, precludes allowing the DB password to be changed until Oracle Application Server Certificate Authority has been stopped.

The changes resulting from executing these commands take effect after the next start of Oracle Application Server Certificate Authority. Each use of `ocactl` requires the OCA 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 OCA administrator password.

Example to change OCA repository password:

```
ocactl setpasswd -type DB
```

Note: If the CA SSL wallet password is changed, you must restart OHS, OCA's OC4J, and OCA, in that order.

Configuration Operations for OracleAS Certificate Authority

The administrator for OracleAS Certificate Authority 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 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 OCA Web Administrator's Certificate
- Configuring National Language Support (NLS) for OCA Screens

Configuring Oracle HTTP Server to Use a Third Party SSL Wallet

When OCA is installed, it is automatically configured in SSL mode. Browsers will warn that this site is not trusted until you import the CA certificate, either through explicit CA import or an additional act of editing the CA entry. To avoid this warning, the OCA administrator can get an SSL certificate for the OCA 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 SSO 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 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 SSO (single sign-on) to bring up the web server in SSL mode automatically, without asking a human for the wallet password.

To import a wallet from a well-known CA, the administrator can do the following:

1. Shut down OCA, 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. Import the certificate of the third-party CA that issued the server certificate.
   c. Import your requested server certificate.
4. Using OWM, import the current OCA CA's certificate as a trust point into this wallet.

   See Also: The *Oracle Application Server 10g Security Guide*, particularly the Appendix on Managing PKI Credentials with Oracle Wallet Manager.

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. Restart OCA, OCA's OC4J, and OHS, in that order.

Revoking a Certificate Authority Certificate

Revoking a CA signing certificate is a very drastic operation, which will make OCA 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 OCA Administrators certificate. It can only be used when OCA services are not running. Revoking a root certificate authority certificate is required before installing a new CA signing for ongoing OCA 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 OCA repository.

Then use `revokecert` to revoke the old CA 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 OCA 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 OCA:

1. Stop OCA. Use the command

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

2. Revoke the CA wallet using the above command.
3. Regenerate the CA SSL wallet.
4. Start OCA. Use the command

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

Revoking the OCA 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 OCA 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 command-line tool `ocactl`, which requires the OCA Administrator password.

Once the OCA 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 Oracle Application Server Certificate Authority web page and fills in the OCA Admin forms.

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 OCA. Use the command

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

2. Stop OCA’s OC$4J. Use the following command in `ORACLE_HOME`:

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

3. Revoke the Web Administrator's certificate using the above command.
4. Start OCA. Use the command

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

Configuring National Language Support (NLS) 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, as follows:

4. The Database Character Set must be UTF8.
5. The UI and online help of OracleAS Certificate Authority are rendered in the locale of the client. 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 OCA, then English is the language used.
6. The practice statement is rendered in the locale in which the Administrator edits the practice statement irrespective of the client locale.
7. 'ocactl' supports NLS depending on the server locale. If the server locale is anything other than the OCA supported languages, display is in English.

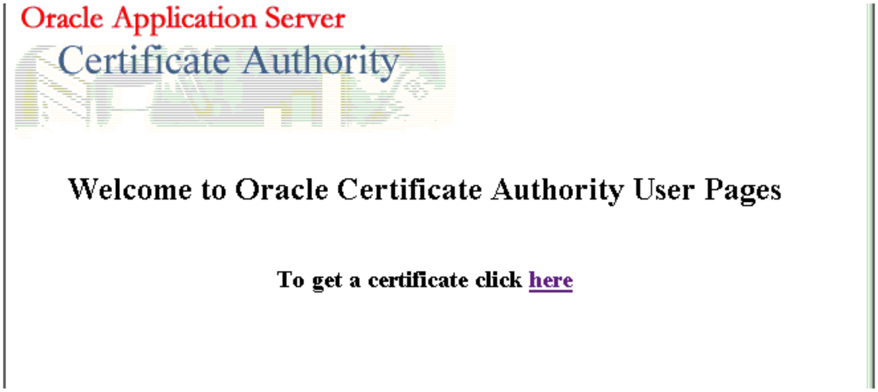
Notes:

- In every locale, ocactl commands are in English.
- Informational, error messages, etc. are displayed in the language of the server locale.
- The CA and CA SSL certificate DN's must not contain multibyte characters. If the CA's DN contains multibyte characters, the install will fail (Bug: 2991110).

Customization Support

OCA enables you to customize the SSO-OCA interface by specifying your own headers and footers for the following three provisioning pages:

1. The Welcome screen



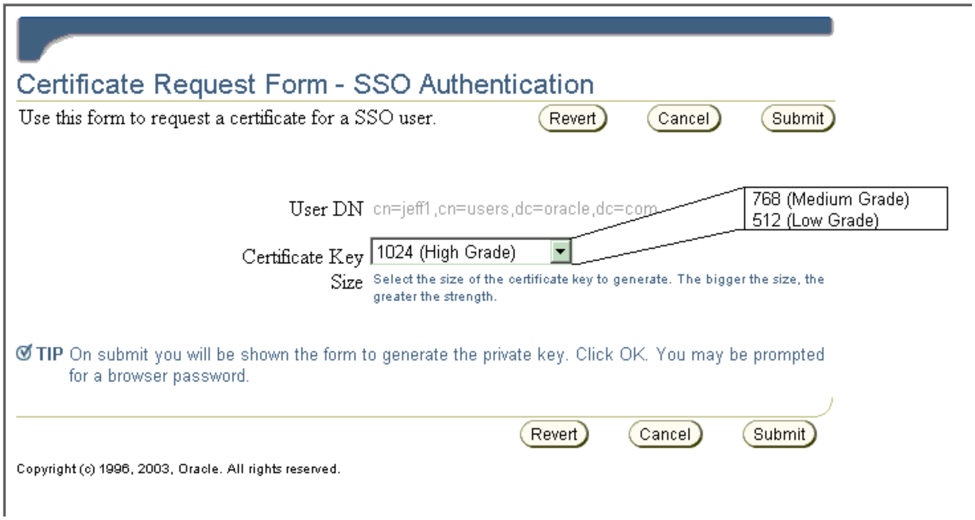
Oracle Application Server

Certificate Authority

Welcome to Oracle Certificate Authority User Pages

To get a certificate click here

2. The Enrollment screen



Certificate Request Form - SSO Authentication

Use this form to request a certificate for a SSO user.

User DN cn=jeff1,cn=users,dc=oracle,dc=com

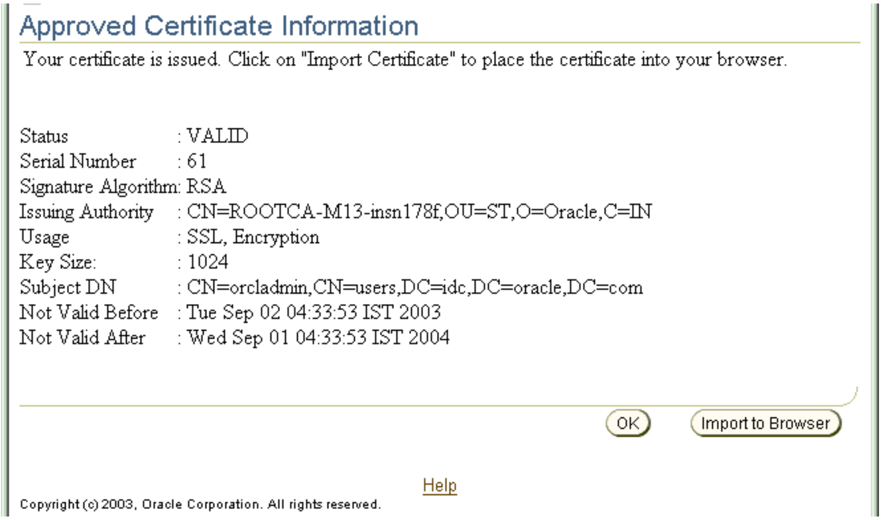
Certificate Key Size 1024 (High Grade)

Select the size of the certificate key to generate. The bigger the size, the greater the strength.

TIP On submit you will be shown the form to generate the private key. Click OK. You may be prompted for a browser password.

Copyright (c) 1996, 2003, Oracle. All rights reserved.

3. The Import Certificate screen



Approved Certificate Information

Your certificate is issued. Click on "Import Certificate" to place the certificate into your browser.

```
Status              : VALID
Serial Number       : 61
Signature Algorithm: RSA
Issuing Authority   : CN=ROOTCA-M13-insn178f,OU=ST,O=Oracle,C=IN
Usage               : SSL, Encryption
Key Size:           : 1024
Subject DN          : CN=orcladmin,CN=users,DC=idc,DC=oracle,DC=com
Not Valid Before    : Tue Sep 02 04:33:53 IST 2003
Not Valid After     : Wed Sep 01 04:33:53 IST 2004
```

Copyright (c) 2003, Oracle Corporation. All rights reserved.

See: Single Sign-on (SSO) and OracleAS Certificate Authority (OCA)in Chapter 3, "Introduction to OCA Administration and Certificate Management"

Although OCA will by default render the existing screens without customization, the OCA 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 OCA 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 directory. The administrator controls the look and feel of this content.

If any customization screen HTML file exists with nonzero size, its content is used instead of the default screen.

Note: After making changes to any of these HTML files, the administrator must restart OCA to cause those changes to be used.

Table 6–1 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 OCA lines reading "Welcome to OracleAS Certificate Authority" and "To get a certificate Click Here"	$ORACLE_HOME/oca/templates/screens/homeheader.html
	For a footer: below 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: below the lines at the bottom, reading "Key Size" and "SKI".	$ORACLE_HOME/oca/templates/screens/enrollfooter.html
Import Certificate Screen	For a header: between the OCA lines reading "View Certificate" and "Certificate details"	$ORACLE_HOME/oca/templates/screens/importheader.html
	For a footer: between the OCA 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 OCA Actions for Oracle Application Server Certificate Authority

You can use the `ocactl` set command to enable log and trace, so that OCA/Admin operations can be viewed in the log/trace storage.

Table 6–2 Storage Locations for OCA Log and Trace Data

Type of Data	Storage Form	Location
Log	OCA repository	OCA repository
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 OCA -state ON`

   Enables storing log messages in the OCA repository.

2. `ocactl set -type TRACE -mode OCA -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` command line tool enables removal of existing log or trace storage at the administrator's choice. The OCA log will be stored in the OCA repository, and the OCA 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 log or storage. The files oca.trc, admin.trc and admin.log can be removed from the file system using OS remove commands.

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 OCA repository

4. `ocactl clear -type TRACE -mode OCA`

   Removes the OCA trace file oca.trc from $ORACLE_HOME/oca/logs

Changing the Infrastructure Services That OCA Uses

Changes to OracleAS Single Sign-On (SSO) and Oracle Internet Directory (OID), 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 Guide*

OCA is installed as part of the OracleAS Identity Management (IM) infrastructure and uses the services of OID, SSO, and a metadata repository. If any of these components is replaced or restored, OCA can be configured to use these new services. can either use existing versions of these three components or work with a new OID, SSO and metadata repository.

Two types of infrastructure change are supported by OracleAS:

- Changing Identity Management (IM) Services (SSO/OID) Used by OCA
- Changing Metadata Repository (MR) Services Used by OCA

The following section describes the display of data regarding these services:

- Where OCA Connection Information Is Stored and Displayed

Changing Identity Management (IM) Services (SSO/OID) Used by OCA

After installation of a new SSO or OID, changing OCA's IM Services requires two steps:

- Installing a new IM and migrating the existing data.
- Configuring OCA to use the newly installed IM Services.

OracleAS provides scripts to migrate data from one IM instance to another, assuming that a new IM (SSO / OID) has been installed. However, you cannot use the Change Identity Management Wizard on the Infrastructure page of the Application Server Control to OCA because OCA itself is an infrastructure component. So OCA supports changing OCA's IM Services by providing the "changesecurity" command from the OCA administration command line tool `ocactl`.

> **See Also:**
>
> - Appendix A, "Command-Line Administration" for more details on the OCA 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 OCA:

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 OCA (Machine 1), bring down OCA, OCA'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, i.e., Machine 2.
6. On Machine 1, execute the following command:

   ```
   $ORACLE_HOME/oca/bin/ocactl changesecurity -server_auth_port portno
   ```

   This command updates the file oca.conf at `$ORACLE_HOME/oca/conf` to point to the new IM Services machine (Machine 2) and port number ("portno"), and registers OCA with the new SSO (Machine 2).

Changing Metadata Repository (MR) Services Used by OCA

Changing OCA's Metadata Services also requires two steps:

- Installing a new MR and migrating the existing data
- Configuring OCA to use the newly installed MR Services

OCA provides scripts to migrate OCA data from one MR to another. Please refer to the Export Utility and Import Utility sections below for more details on the migration scripts.

The Change Metadata Repository Wizard on the Infrastructure page of the Application Server Control is not available for OCA because OCA is an infrastructure component. To change Metadata Services, OCA provides the following command from the OCA administration command line tool:

```
changeschema -host hostname -service service
```

This command (used in step 9 below) changes the password for the new OCA schema used in the new MR to be the old password stored in the old OCA schema's password store. It also updates `$ORACLE_HOME/oca/conf/oca.conf` with the new MR host name and service.

The following steps establish the new metadata services for OCA:

1. Install OCA and metadata repository on Machine 1.
2. Install metadata repository on another machine (Machine 2).
3. Stop OCA, OCA's OC4J, and OHS on Machine 1. Use these commands:

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

4. Export data from MR of Machine 1 using the 'migoca' script under `$ORACLE_HOME/oca/bin`. (See Export Utility for more details.)
5. Transfer the export dump file to Machine 2.
6. Import data to MR of Machine 2 using the 'migoca' script under `$ORACLE_HOME/oca/bin`. (See Import Utility for more details.)
7. Manually edit the entry in Oracle Internet Directory regarding the Metadata Services used by OCA.
8. Restart OHS, OCA's OC4J, and OCA, in that order, on Machine 1, as follows:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl startproc type=ohs
   ```

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

9. Then execute the following command, where "new" refers to Machine 2:

```
ocactl changeschema -host <new-hostname> -service <new-MR-service>
```

Export Utility

This utility, available under `$ORACLE_HOME/oca/bin`, is used to export OCA data from the source MR, that is, from the specified OCA Server. After the export command is executed, an export dump (migoca.dmp) file is created, which must be transferred to the destination machine.

Syntax for the export command is as follows:

```
migoca export    -p <system_password>   [-c <connect_string>]  [-d <log_dir>]
[-f <log_file>] [-df <dump_file_name>]
```

where

Table 6–3 Parameters for Export Utility

Flag and Parameter	Meaning or Usage	Default Parameter
`-p  system_password`	"system" user password	None
`-c  connect_string`	Net8 Connect string	None
`-d  log_dir`	Log directory name	Current directory
`-f  log_file`	Log file name	migoca.log
`-df dump_file_name`	Dump file name	migoca.dmp

Note: After export, the dump file (migoca.dmp) needs to be transferred to the destination machine.

Import Utility

This utility, available under `$ORACLE_HOME/oca/bin`, is used to import OCA data obtained from the source database to destination MR.

Syntax for the import command is as follows:

```
migoca import  -po <oca_password> -pp <oraoca_password> -p <system_password>
[-c <connect_string>] [-d <log_dir>] [-f <log_file>]
[-df <dump_file_name>]
```

where

Table 6–4 Parameters for Import Utility

Flag and Parameter	Meaning or Usage	Default Parameter
`-po oca_password`	"oca" user password	None
`-pp oraoca_password`	"oraoca_public" user password	None
`-p system_password`	"system" user password	None
`-c connect_string`	Net8 Connect string	None
`-d log_dir`	Log directory name	Current directory
`-f log_file`	Log file name	migoca.log
`-df dump_file_name`	Dump file name	migoca.dmp

Note: The existing data in the target database will be replaced with the data from the source MR. After this source and destination MR will be in sync.

Where OCA Connection Information Is Stored and Displayed

Information defining connections to the OCA repository and directory (used for publishing certificates) is stored in Oracle Internet Directory (OID). This connection information is originally written to OID when OracleAS is installed, at which time it is also fetched from OID and written into the Oracle Application Server Certificate Authority configuration file `$ORACLE_HOME/oca/conf/oca.conf`.

This connection information is displayed under Settings in the General subtab of the Oracle Application Server Certificate Authority web interface for the administrator.

OracleAS Certificate Authority and High-Availability Features

The primary reference for Oracle Application Server high-availability features is the *Oracle Application Server 10g High Availability Guide*. The following discussion is merely an overview to orient you to those features.

Oracle Application Server 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:

- OracleAS Certificate Authority Deployment Using Cold Failover
- OracleAS Certificate Authority Deployment Using Real Application Clusters

See Also: *Oracle Application Server 10g High Availability Guide*.

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 OracleAS 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 OCA / OracleAS software and database reside. If the hardware for OracleAS 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 OCA / OracleAS processes to effect transparent failover. Clients continue to talk to the same logical host as before, with minimal service disruption. Oracle Application Server Certificate Authority, too, must be restarted with the `ocactl start` command, after HTTP server and OC4J are brought up.

See Also: *Oracle Application Server 10g High Availability Guide*.

OracleAS Certificate Authority Deployment Using Real Application Clusters

Oracle Real Application Clusters provides a robust cluster architecture for the OracleAS Infrastructure, offering a more transparent high availability solution than cold failover clusters. Because all nodes in the Real Application Clusters solution are active, failover from one node to another is quick, requiring no manual intervention. This active-active set up also provides scalability to the Infrastructure deployed on it. A standard configuration is illustrated below.

The database files are installed in shared storage accessible by all nodes. The database instances open the database concurrently for read / write operations.

Infrastructure configuration files are not in the database but in the file systems local to each node, containing identical but node-specific configuration information. More than two nodes can exist in the cluster and all of them actively accept requests from the middle tier.

The cluster is front-ended by a load balancer appliance, which Oracle recommends be deployed in a fault-tolerant mode to maintain availability in case of load balancer failure. This load balancer directs non-Oracle Net traffic, such as HTTP, HTTPS, and LDAP (directory) requests, from the middle tier to the Infrastructure. The configuration of the load balancer is set to direct requests from the middle tier to any of the active Infrastructure nodes.

OCA provides limited support for RAC. It can use the RAC configuration’s other infrastructure components, such as Oracle Internet Directory, Oracle Database, and OracleAS Single Sign-On, but OCA itself cannot be in the RAC mode.

> **See Also:** *Oracle Application Server 10g High Availability Guide* for guidance regarding how to install these components in the RAC mode.

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 OracleAS backup recovery tool aids in backing up and recovering the OracleAS 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 OracleAS backup / recovery tool, and component-wise backup and recovery, please refer to the backup / recovery documentation in the *Oracle Application Server 10g 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 above.

Scenarios in which backup/recovery techniques could be used to recover data include the following situations:

Table 6–5 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 crash 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 OracleAS 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 Oracle Application Server Certificate Authority (OCA) 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 6–6 Backup/Recovery Tools

Tool Name	Functionality
Cold Backup / Recovery	Refers to restoring the entire OracleAS infrastructure instance including the Oracle Home, configuration files, and database files that were backed up after completing a clean and normal shutdown of all OracleAS infrastructure processes and metadata repository
Partial Online (hot) Backup / Recovery	Refers to restoring the OracleAS 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 OracleAS infrastructure configuration files taken from an online backup

Since OCA uses the Oracle Database as its primary repository, the OCA information stored in that database will be automatically backed up when that database is backed up. Similarly, OCA relies on Oracle Internet Directory (OID) for publishing certificates and for certain SSO operations. The three books named above provide the detailed information supporting all related backup and recovery operations.

In addition to information stored in the database and directory, Oracle Application Server 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 OCA is installed):

- `$ORACLE_HOME/oca/conf/oca.conf`
- `$ORACLE_HOME/oca/pwdstore/ocmpassword.p12`
- `$ORACLE_HOME/oca/wallet/ssl/ewallet.p12`
- `$ORACLE_HOME/oca/wallet/ssl/cwallet.sso`
- `$ORACLE_HOME/Apache/Apache/conf/httpd.conf`
- `$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 OCA may be Sub CAs or independent CAs that trust each other.

By default, when an Oracle Application Server Certificate Authority (OCA) instance is installed in a particular machine, an entry is placed into Oracle Internet Directory representing that installed OCA 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 OCA, 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 OCA.

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 OCA, 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 OCA instance so that it can only publish to a particular subscriber realm. For example, the UK OCA could be restricted to publishing only to the UK subscriber realm. If this is done, then a certificate issued by the UK OCA to a US user could not be published, because the user's standard realm would not be accessible to the UK OCA.

To restrict an OCA 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 OCA 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 5, "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 7 of the example in that chapter's section entitled "An Example of a Custom Policy Plug-in":

```
7:          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.

7

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.

Oracle Certificate Authority has separate HTML interfaces for end-user and administrator interaction with the Oracle Application Server Certificate Authority server. Using these HTML forms, end-users can perform personal certificate-related operations and the administrator can perform certificate administration and management.

For the OracleAS Certificate Authority web administrator interface, see

This 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
- Downloading a CA Certificate
- Importing the Certificate Revocation List (CRL) into Your Browser
- Downloading Certificate Revocation Lists into Your File System
- Importing a Certificate to Your Browser
- Exporting (Backing up) Your Wallet from Your Browser

- Importing a Certificate from Your File System

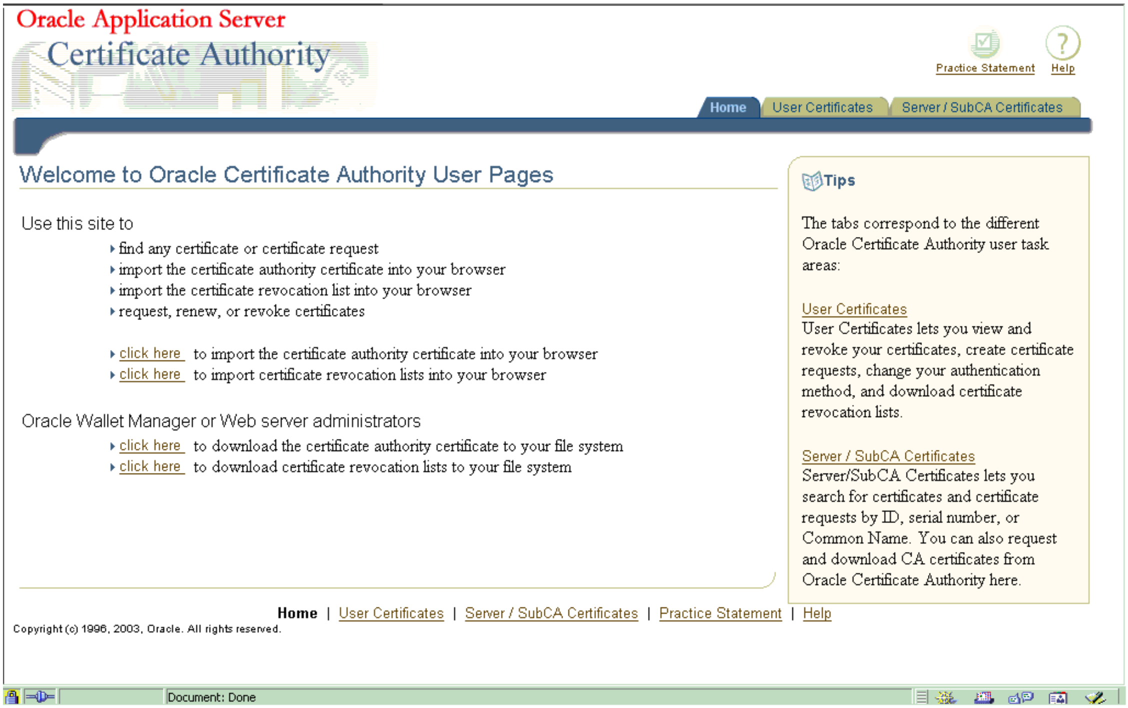
Both Netscape and Internet Explorer are supported.

Accessing the User Interface

To access the home page for the end-user interface to Oracle Application Server 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://Oracle_HTTP_host:ssl_port/oca/user
```

The Oracle Application Server Certificate Authority user home page appears:



Oracle Application Server Certificate Authority

Practice Statement | Help

Home | User Certificates | Server / SubCA Certificates

Welcome to Oracle Certificate Authority User Pages

Use this site to
- find any certificate or certificate request
- import the certificate authority certificate into your browser
- import the certificate revocation list into your browser
- request, renew, or revoke certificates

- click here to import the certificate authority certificate into your browser
- click here to import certificate revocation lists into your browser

Oracle Wallet Manager or Web server administrators
- click here to download the certificate authority certificate to your file system
- click here to download certificate revocation lists to your file system

Tips

The tabs correspond to the different Oracle Certificate Authority user task areas:

User Certificates
User Certificates lets you view and revoke your certificates, create certificate requests, change your authentication method, and download certificate revocation lists.

Server / SubCA Certificates
Server/SubCA Certificates lets you search for certificates and certificate requests by ID, serial number, or Common Name. You can also request and download CA certificates from Oracle Certificate Authority here.

Home | User Certificates | Server / SubCA Certificates | Practice Statement | Help

Copyright (c) 1996, 2003, Oracle. All rights reserved.

Document: Done

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 import into your browser the Certificate Authority’s certificate or the latest certificate revocation list (CRL).

Similarly, administrators can use their **click here** links to download the CA certificate or CRL into their file system for additional uses.

End-User Tabs and Processes

The Oracle Application Server Certificate Authority web interface enables two types of end-user interaction with Certificate Authority, as represented by the two tabs:

From the **User Certificates** tab you can

- authenticate yourself to the Oracle Application Server Certificate Authority, either by using existing Single Sign-On (SSO) or SSL certificates, or by requesting manual authentication by an administrator,
- create a new certificate request for manual approval by the OCA administrator (for end-users or servers),
- request and receive a certificate automatically (for SSL and SSO users)
- import, view, revoke, or renew your certificates,
- change your authentication method,
- download the CA certificate, or
- download the latest certificate revocation list (CRL).

Table 7–1 lists the types of certificates that Oracle Application Server Certificate Authority supports and provides a brief explanation for each.

Table 7–1 Certificate Types and Uses

Certificate Type	Meaning/Usage
Encryption	User intends to enable others to send messages encrypted with the public key so that only user can decipher them using the private key.
Signing	User intends to sign message digests with the private key, enabling others to use the public key to verify that user originated the message and it is unchanged.

Table 7–1 Certificate Types and Uses

Certificate Type	Meaning/Usage
Code signing	User intends to sign software with private key, enabling clients to use the public key to verify that user is the source of the software.
Certificate signing	User intends to use private key to sign certificates it issues, enabling recipients to use its public key to verify that the certificate was in fact signed by user.
SSL	User intends the certificate for use in SSL authentication.

From the **Server/SubCA Certificates** tab, you can

- search for certificates and certificate requests by ID, serial number, or Common Name, etc.,
- request server and sub-CA certificates, or
- import CA certificates or certificate revocation lists (CRLs).

User Certificates Tab

Upon first entering this tab, you see the **Authentication** page, which allows you to select how you intend to authenticate yourself to Oracle Application Server Certificate Authority.

Table 7–2 lists the available types and methods:

Table 7–2 Types of Authentication

Authentication Type	Description	Method in brief (details in following sections)
Single Sign-On (SSO)	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: 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:
Netscape shows choices for key size (512, 1024).

Internet Explorer uses a hard-coded "basic" choice of 512 bits, usually the first choice in its list, with an enhanced choice, of 1024 bits, further down. However, in some versions of IE, Gemplus is listed as the first choice. If the computer does not have a smart card card-or-reader, then choosing Gemplus give users an error because no key-size resolution is found. If there *is* a card-reader, then the Gemplus smart card determines the key size.

Single Sign-on Authentication (SSO)

The following steps enable SSO users to get a certificate automatically, or to manage their certificates, by supplying the required SSO authentication information, such as username and password:

1. In the **Authentication** form, select the **Use Your Oracle Single Sign-On Name and Password** option and click **Submit**.
   You will be redirected to the SSO login page.
2. Enter your SSO 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 **Key Size** appears, referring to the size, in bits, of the key-pair to be generated: 512, 1024, ...
   - **In Internet Explorer**, the phrase **Key Store** appears, referring to a choice of providers for cryptography service. Standard choices include Microsoft Basic Crypto Provider, Microsoft Enhanced Crypto Provider, and Microsoft Strong Cryptographic Provider, for which the key sizes are fixed at 512, 1024, and 2048 bits, respectively. Other choices may also be present, such as Gemplus for smartcard usage. Select the size according to your requirements. Oracle recommends using 1024 bits (the "Enhanced" choice).
   - **Validity Period:** Duration of the certificate’s validity, in days. However, SSO 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 **Import to Browser** button to import the certificate into your browser. Netscape and Internet Explorer report successful import differently, as described below:

Note: If you click **OK** instead of **Import to Browser**, your certificate ***is*** created, stored in the OCA repository, and published to the Oracle Internet Directory. However, your browser cannot supply it to a server until you import it. See Importing a Certificate to Your Browser

- **In Netscape**, you will know the certificate has been imported 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 imported.

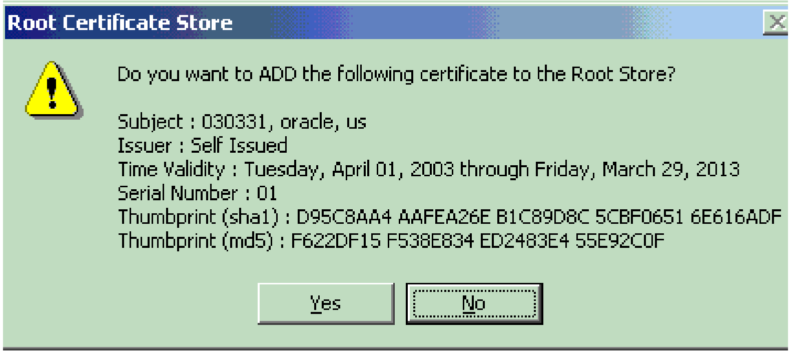
  Note: For this certificate to be trusted, you need to edit the CA certificate's uses, specifying that you trust certificates issued by this Certificate Authority for network sites, email users, software developers, or all three. Checkboxes for these choices are reached through the Security choice on Netscape's menu bar: see Trusting a Certificate Issuer in Netscape.

- **In Internet Explorer**, you know the certificate has been imported when you see the message "Certificate has been imported successfully". You are also asked whether you want the Signer's certificate imported, on a window showing the details of that CA. Click **OK** to ensure that certificate is also imported. Internet Explorer automatically treats such a certificate as trusted.

Configuring Your Browser to Trust OracleAS Certificate Authority

This process is slightly different in Netscape and Internet Explorer.

Trusting a Certificate Issuer in Internet Explorer When you import a certificate using Internet Explorer, it asks whether you wish to add that certificate to the Root Store:



Do you want to ADD the following certificate to the Root Store?

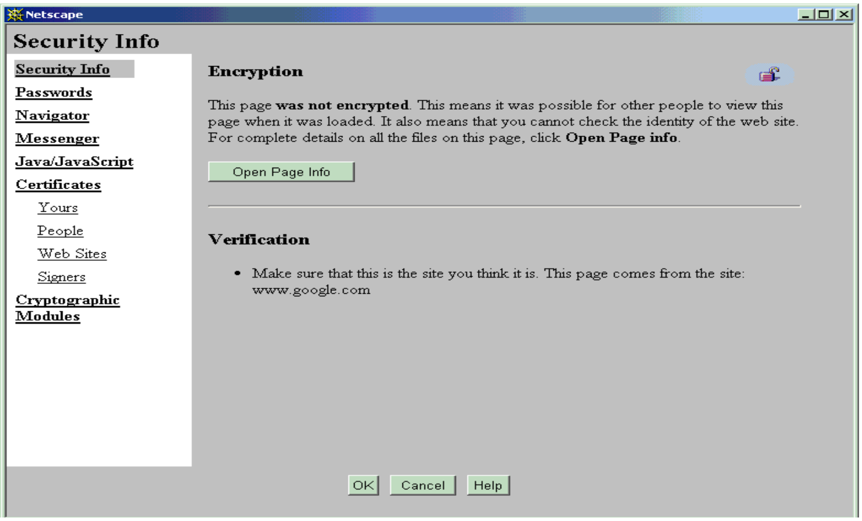
Subject : 030331, oracle, us
Issuer : Self Issued
Time Validity : Tuesday, April 01, 2003 through Friday, March 29, 2013
Serial Number : 01
Thumbprint (sha1) : D95C8AA4 AAFEA26E B1C89D8C 5CBF0651 6E616ADF
Thumbprint (md5) : F622DF15 F538E834 ED2483E4 55E92C0F

Clicking **Yes** imports 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 import a certificate using Netscape, it imports 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 have explicitly identified 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:



Security Info

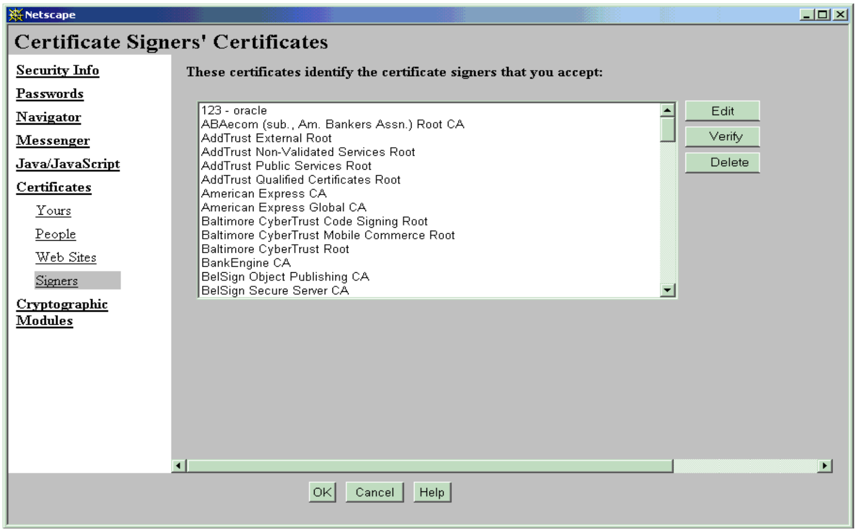
Encryption

This page **was not encrypted**. This means it was possible for other people to view this page when it was loaded. It also means that you cannot check the identity of the web site. For complete details on all the files on this page, click **Open Page info**.

Verification

- Make sure that this is the site you think it is. This page comes from the site: www.google.com

1. Click the "Signers" link. A page like the following appears:

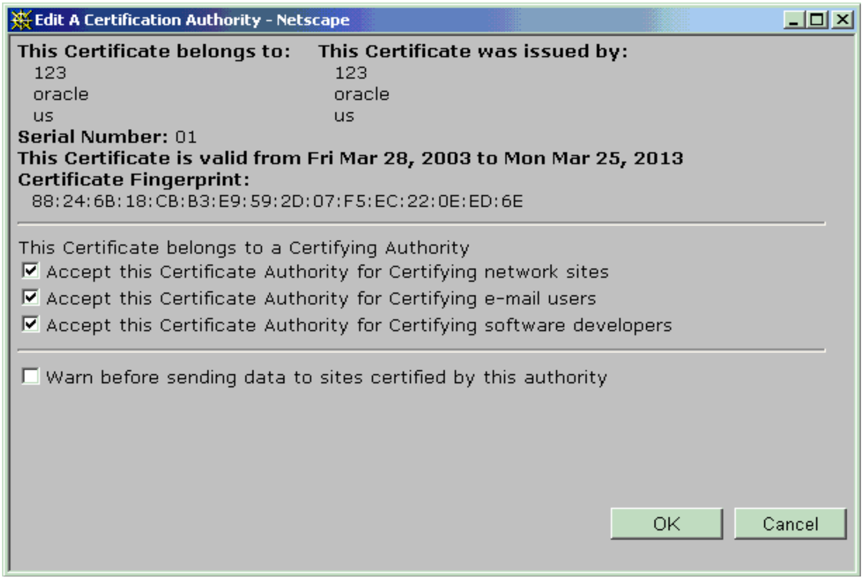


Certificate Signers' Certificates

These certificates identify the certificate signers that you accept:

123 - oracle
ABAecom (sub., Am. Bankers Assn.) Root CA
AddTrust External Root
AddTrust Non-Validated Services Root
AddTrust Public Services Root
AddTrust Qualified Certificates Root
American Express CA
American Express Global CA
Baltimore CyberTrust Code Signing Root
Baltimore CyberTrust Mobile Commerce Root
Baltimore CyberTrust Root
BankEngine CA
BelSign Object Publishing CA
BelSign Secure Server CA

2. Click the name of the signer that you noted when viewing the certificate's details, and click **Edit**. A page like the following appears:



This Certificate belongs to:
123
oracle
us

This Certificate was issued by:
123
oracle
us

Serial Number: 01
This Certificate is valid from Fri Mar 28, 2003 to Mon Mar 25, 2013
Certificate Fingerprint:
88:24:6B:18:CB:B3:E9:59:2D:07:F5:EC:22:0E:ED:6E

This Certificate belongs to a Certifying Authority
☑ Accept this Certificate Authority for Certifying network sites
☑ Accept this Certificate Authority for Certifying e-mail users
☑ Accept this Certificate Authority for Certifying software developers

☐ Warn before sending data to sites certified by this authority

3. Click the three checkboxes shown as checked in the above 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.

Secure Sockets Layer (SSL) Authentication

If you already have an SSL certificate from the Certificate Authority, you can obtain an Oracle Application Server 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 above 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 **Import to Browser** button to import 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, use the Enrollment form's drop-down list to select key size and either SSL/Encryption or Signing certificate, 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 import the certificate.

Certificate Retrieval, Renewal, and Revocation

After a certificate request is approved, the issued certificate can be retrieved for review and importation. Use the same machine and browser as when you requested the certificate.

After an SSO- 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 import 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 import it as described in Importing a 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

SSO 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 on **View Details**, and then renew the certificate.

Certificate Revocation

SSO and SSL certificate users can revoke certificates.

If errors or problems are found with a certificate, or if a private key is stolen, etc., 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 OCA repositories and it will 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 10g 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 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.

Oracle Application Server Certificate Authority can be a root CA or it can obtain a Subordinate CA certificate from a third-party CA. Oracle Application Server 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.

Downloading a CA Certificate

In Netscape, after you click **Request a Certificate**, Oracle Application Server Certificate Authority presents a sequence of dialog boxes. These dialogs describe the operations that need to happen in order to accept the OCA certificate. Click **Next** on each dialog box as it is presented, and **Finish** on the last one. Your CA certificate will be automatically downloaded into your browser.

For Internet Explorer, you are asked simply to accept or reject the CA Certificate import. 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 import 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 import to place the CA certificate into the browser's repository.

Importing the Certificate Revocation List (CRL) into Your Browser

Importing a certificate revocation list (CRL) into your browser enables it 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.

This operation requires different actions in different browsers:

- In Netscape
- In Internet Explorer (IE)

In Netscape

From the **User Certificates** tab of Oracle Application Server Certificate Authority, do the following tasks:

1. Click the **Download CRL** button.
   The **Download CRL** form appears.
2. Click **Import CRL into Browser**.
   The CRL is imported.

The CRL can be seen under, Security-> Signer's-> View / Edit CRL.

If you already have the CRL and it has the same or later validity of the CRL being downloaded, a small dialog box informs you that the CRL you are attempting to download is not later than one already in your browser.

You can also

- download a binary copy of the CRL by clicking the button labeled **Download CRL in Binary** and choosing the directory in which you want it stored, or
- download a copy in Base64 format by clicking the button labeled **Download CRL inBASE64 format** and choosing the target directory.

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 **Save to Disk** is chosen, you then do the following actions:

1. Select from the Tools menu **Tools->Internet Options->Content->Certificate**

2. Select **Import Certificate**.
3. Select the CRL.

This procedure will then show the message "CRL Imported."

Downloading Certificate Revocation Lists into Your File System

Downloading a certificate revocation list (CRL) into 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 download a CRL, follow these steps:

1. Go to the Oracle Application Server Certificate Authority User Certificates Page.
2. Click **Download CRL**.
3. Click either **Download CRL in Binary** or **Download CRL in BASE64 format.**
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.

Importing a Certificate to Your Browser

After your request for a certificate is approved,Oracle Application Server 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 on the request form all the correct information.

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 then 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, the CA certificate imported along with the user certificate is not automatically trusted in Netscape. You need to establish the trust, as follows:

- In Netscape:

  1. Click on the security icon.
  2. Select **Signers** repository.
  3. Select the appropriate CA Certificate by name. (You might be prompted for the repository password.)
  4. Edit the CA Certificate.
  5. Check the appropriate check boxes.
  6. Select **OK**.

  This process establishes the desirable trust relationships, so that when you try to establish an SSL session, the Netscape browser will trust the certificates issued by this importing certificate.

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 them 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.

In Netscape, use the following steps:

1. Select the Security icon in the menu bar.

   A window opens showing your choices for reviewing security information.

2. Under the **Certificates** heading, click **Yours**.

   A subordinate window opens showing the names of your certificates.

3. Click the particular certificate you want to export.
4. Click the **Export** button to the right of that subordinate window.
5. When asked, enter a password to preserve the security and integrity of this certificate. 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 reinstall this certificate. Without the password, it will not be usable.

6. When asked, enter the file system destination, pathname and filename, where this encrypted certificate is to be stored.

   A message appears saying "Your certificates have been successfully exported."

In Internet Explorer, use the following steps:

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 **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 below 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, pathname 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. The steps are as follows:

In Netscape, use the following steps:

1. Select the Security icon in the menu bar (or the status bar at the bottom).
2. Under Certificates, click the "Yours" link. A list and some buttons appear.
3. Click **Import a Certificate...**.
4. Navigate to the directory containing the wallet with your desired certificate, and double-click the .p12 file.

   A dialog box will ask you for the wallet’s password.
5. Enter the password. (If the password you supply is incorrect, Netscape says the file is corrupt or not valid, since decryption failed. If the certificate is already imported, Netscape will tell you and take no further action on this request.)
6. After successfully importing the certificate, click **OK**.

In Internet Explorer (IE), use the following steps:

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 through 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 Oracle Application Server Certificate Authority administration tasks using the command line tool `ocactl`, operating through the computer hosting the Oracle Application Server Certificate Authority.

This chapter contains the following topics:

Table A–1 Links to Commands and Configuration Operations

Link to 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 OCA Repository Connection Information
Root Certificate Operations	■ Regenerating the Root Certificate Authority's Certificate ■ Revoking a Root CA Certificate

Table A–1 Links to Commands and Configuration Operations (Cont.)

Link to General Topic	Links to Specific Subtopics
SSL/SSO 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 Wallet from Oracle Application Server Certificate Authority ■ Installing/Importing a Sub CA 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 OCA administrator, you use the command line tool named `ocactl` to specify the parameters needed to perform the various Oracle Application Server Certificate Authority operations. (You may need to add `oca/bin` to your path.) Each time this tool is invoked it requests your OCA 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 Oracle Application Server 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 Oracle Application Server 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–1.

Table A–1 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:

changeschema, changesecurity, clear, generatewallet, help, importwallet, linksso, renewcert, revokecert, set, setpasswd, start, stop, unlinksso, updateconnection

Table A–1 Operations and Parameters of the Oracle Application Server Certificate Authority (OCA) ocactl Tool

Operation	Parameters	Meaning
changeschema	-host hostname -service service	Used when the entire database is changed to a different one and the data is migrated to the new database. hostname is the name of the new machine; service is the name of the service on that machine.
changesecurity	-server_auth_port port	Changes the Identity Management services (OID/SSO) used by OCA to the new OID and SSO server. Updates oca.conf with the new IM machine and port number, and uses the specified port while registering OCA with the new SSO.
clear	LOG, TRACE OCA 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 OCA or ADMIN. (If OCA is not running, all such data is cleared.) Examples of each command appear in Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics" at Log or Trace OCA Actions for Oracle Application Server Certificate Authority.
convertwallet	See later discussion	after this table: "Convertwallet" Explained with Examples.

Table A–1 Operations and Parameters of the Oracle Application Server Certificate Authority (OCA) ocactl Tool (Cont.)

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 certificate. A sample "generatewallet" command will thus look like this: `ocactl generatewallet -type CASSL` Wallets of the type named below are store in the indicated place: CA Oracle Application Server Certificate Authority repository CASSL `$ORACLE_HOME/oca/wallet/ssl` CASMIME Oracle Application Server Certificate Authority repository
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: a subordinate CA server wallet. A sample "importwallet" command will thus look like this: `ocactl importwallet -type SUBCA`
linksso	<none>	Registers OCA with SSO to display OCA certificate enrollment form to SSO users who lack a certificate, so they can request one. (This command does not require OCA service to be shut down, but it won't take effect until the SSO server is restarted.)
renewcert	CA, CASSL, CASMIME	When OCA 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 OCA installation inoperable.)	CA WEBADMIN (Be very careful and certain before taking this action.)	Usable only when OCA 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–2 for details on revocation reasons.

Table A–1 Operations and Parameters of the Oracle Application Server Certificate Authority (OCA) ocactl Tool (Cont.)

Operation	Parameters	Meaning
set	LOG or TRACE, ON or OFF OCA or ADMIN	Sets the OCA configuration to use the additional parameters for state (ON or OFF) or mode (OCA or ADMIN) specified after LOG or TRACE, as follows: Examples of each command appear in Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics" at Log or Trace OCA Actions for Oracle Application Server Certificate Authority.
setpasswd	CA, DB, CASSL, or `CASMIME`	Requests and resets the password for the specified role: administrator, database administrator, directory, OCA user, or certificate authority SSL server. 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 Oracle Application Server Certificate Authority service. (OC4J, OHS, and the database must already be in operation for OCA to start. You control OC4J and OHS by 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 Oracle Application Server Certificate Authority services. A sample "status" command will thus look like this: `ocactl status`

Table A–1 Operations and Parameters of the Oracle Application Server Certificate Authority (OCA) ocactl Tool (Cont.)

Operation	Parameters	Meaning
stop	<no parameters>	Stops the Oracle Application Server Certificate Authority service. (Does not stop database, web server, or OracleAS. 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 OCA from SSO, so the screens for welcome and enrollment form will not be shown. (This command does not require OCA service to be shut down, but it won't take effect until the SSO server is restarted.)
updateconnection	<no parameters>	Writes the connection information stored in Oracle Internet Directory (OID) for connecting to the OCA repository and directory (used for publishing certificates) into the OCA configuration file $ORACLE_HOME/oca/conf/oca.conf. (This connection information is displayed under Settings in the General subtab of the Oracle Application Server Certificate Authority web interface for the administrator.) OCA connection information is originally written to OID when OracleAS is installed, when it is also fetched from OID and written into oca.conf. This information changes if OCA is moved to another database. Also used in RAC-enabled database when the connect string is changed because more RAC nodes are added or removed. No data needs to be migrated.

"Convertwallet" Explained with Examples

Table A–1 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.

The convertwallet command is used to convert an SSL Server wallet (ewallet.p12, in PKCS#12 format) into a wallet in the SSO 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 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 SSO (single sign-on) to bring up the web server in SSL mode automatically, without asking a human for the wallet password.

The `convertwallet` 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 OCA during OCA's installation. This command therefore uses the OCA 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).

Oracle Application Server Certificate Authority will use the new SSO wallet stored at $ORACLE_HOME/oca/wallet/ssl/ only after OHS, OCA's OC4J, and Oracle Application Server Certificate Authority are restarted (in that order). (To start the required infrastructure, see section 4.1 in *Oracle Application Server 10g 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 Oracle Application Server Certificate Authority services that support the forms for administrator and user access. To start OC4J and OHS, use the command-line tool `opmnctl` as follows:

```
$ORACLE_HOME/opmn/bin/opmnctl startproc type=oc4j instancename=oca
$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 an XML configuration file.

Stopping the Oracle Application Server Certificate Authority Server

The stop command stops the Oracle Application Server Certificate Authority services. No other services are affected: database, OracleAS, and web server remain unchanged.

To stop the Oracle Application Server Certificate Authority services, issue the following command:

```
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 Oracle Application Server Certificate Authority engine. The response shows whether the following facilities are open or closed: the database connection pool; logger, tracer, and XML files; and the password store.

To get the Oracle Application Server 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 OCA operations:

Password Type	Password Usage
OCA database user	Enables access to database tables containing OCA 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, etc. 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 OCA 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:

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
Administrator signing notification mails	`ocactl setpasswd -type CASMIME`	CASMIME password in the password store

With the exception of the database password (DB), all other passwords can be changed even when Oracle Application Server Certificate Authority is in operation. The possibility of active connections to the CA, using the existing DB password, precludes allowing the DB password to be changed until Oracle Application Server Certificate Authority has been stopped. After the DB password is changed while

Oracle Application Server Certificate Authority is stopped, the new password will be required to start it and will be used while it is in operation.

The changes resulting from executing these commands take effect after the next start of Oracle Application Server Certificate Authority. Until the Certificate Authority is restarted, the new passwords will not be used: the referenced wallets will remain usable as encrypted with the old passwords, because that information is already stored in memory. After Oracle Application Server Certificate Authority is restarted, the new passwords will be in effect.

Each use of `ocactl` requires the OCA 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 OCA administrator password.

Regenerating the Root Certificate Authority's Certificate

When installing Oracle Application Server 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` command line tool. The new CA certificate and private key will be stored in the OCA 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 CA wallet, a CRL issued by old CA will not be useful.

> **See:** Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics", specifically the sections entitled Regenerating the CA Signing Wallet and Regenerating the CA SSL and CA SMIME Wallets.

This can only be done after OCA is successfully installed and OCA service is not running.

The root CA wallet can be generated only when OCA is not running. If OCA is running, stop OCA and use this command to regenerate the Root CA wallet:

```
ocactl generatewallet -type CA
```

This certificate is stored as a binary file in the directory $ORACLE_HOME/oca/wallet/ca.

The signing key is stored in the directory $ORACLE_HOME/oca/wallet/ca, encrypted by the OCA administrator password.

The password store is kept in the directory $ORACLE_HOME/oca/pwdstore.

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 Oracle Application Server Certificate Authority engine to listen in HTTPS mode. If these are compromised or corrupted, or the CA wallet is regenerated, you must regenerate them in order to re-establish secure communications.

The CA SSL wallet can be generated only when OCA is not running. If OCA is running, stop OCA 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 SSO 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 OCA 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 OCA Administrator's certificate. It can only be used when OCA is not operating.

Revoking a root certificate authority certificate is required before installing a new root CA for ongoing OCA operations.

When you intend to install a new CA, use `revokecert` first to revoke the old CA 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 OCA repository.

Once the OCA 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 OCA. 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–2 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, 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 root CA's key has been compromised, so certificates based on it may not in fact be trustworthy.
`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.

Converting a CA SSL Server Wallet into SSO Form

You can use the command line tool's `convertwallet` command to convert a CA SSL server wallet into Oracle Single Sign-on (SSO) format. The command uses the current CA SSL wallet location, unless you specify a different location.

To convert the CA SSL Server wallet to SSO format, issue the following command as root user, with `<wlt-location>` replaced by the desired destination:

```
ocactl convertwallet -format SSO [-walletwrl <wlt-location>]
```

The optional parameter `-walletwrl` identifies the next parameter as specifying the source location where the CA SSL PKCS#12 wallet is presently located. It should be stored in the filename ewallet.p12 in that directory. 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 <wlt-location>`.

If this source location is not specified, then `ocactl` assumes the wallet is the CA SSL wallet generated by OCA during OCA's installation. This command therefore uses the OCA 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).

Generating a Sub CA Wallet from Oracle Application Server Certificate Authority

You can generate a Sub CA wallet from Oracle Application Server Certificate Authority as follows:

1. Create a new wallet and generate a certificate request using Oracle Wallet Manager.
2. Using the Server/Sub CA enrollment form, submit the PKCS10 request and select certificate usage as CA signing.
3. Using the Oracle Application Server Certificate Authority Administrative form, issue a Sub CA certificate. Please specify the path-length, i.e., 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 wallet using Oracle Wallet Manager. The wallet will be stored as ewallet.p12.

Installing/Importing a Sub CA Wallet

The steps in this section enable you to install and use a Sub CA wallet, creating a hierarchy of CAs. This wallet can be one generated from Oracle Application Server Certificate Authority, as in Generating a Sub CA Wallet from Oracle Application Server Certificate Authority, or come from any X.509v3-compliant CA, such as CMS.

> **Note:** 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 wallet, you must install Oracle Application Server Certificate Authority successfully, which will create its repository, the password store, the Root CA wallet, and the CA SSL wallet. Then do the following steps:

1. Stop OC4J and OHS 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
   ```

2. Use the ocactl importwallet command to install the Sub CA wallet, which is

   ```
   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 OCA 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 OCA 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 Oracle Application Server 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 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 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 Oracle Application Server Certificate Authority to listen in HTTPS mode, and they can be regenerated if they become compromised or corrupted, in order to re-establish secure communications.

Generating the Sub CA SSL wallet is also done when OCA 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 SSO 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 OCA 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 Oracle Application Server Certificate Authority.

Oracle Application Server Certificate Authority will now use the Sub CA certificate for signing certificate requests.

Clearing Log or Trace Storage

The 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 OCA repository; clearing trace data removes the file oca.trc from $ORACLE_HOME/oca/logs.

Updating OCA Repository Connection Information

The connection information used for publishing certificates is displayed under Settings in the General subtab of the Oracle Application Server Certificate Authority web interface for the administrator. This information is stored in Oracle Internet Directory (OID) for connecting to the OCA repository and directory.

The `ocactl` command `updateconnection` writes the connection information into the OCA configuration file $ORACLE_HOME/oca/conf/oca.conf.

OCA connection information is originally written to OID when OracleAS is installed, when it is also fetched from OID and written into oca.conf. This information changes if OCA 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.

Oracle Application Server Certificate Authority has an expedited process to enable SSO-authenticated users to request and receive such certificates.

When the OCA administrator executes the `ocactl linksso` command, it registers OCA with SSO to display OCA's certificate enrollment form to SSO users who lack a certificate. Using the short process thus presented, such users can request a certificate, which OCA then issues, and the user can import it into the browser for future authentication.

All aspects of this process are discussed in Chapter 3, "Introduction to OCA Administration and Certificate Management", in the section titled Single Sign-on (SSO) and OracleAS Certificate Authority (OCA). An overview appears in Chapter 6, "OracleAS Certificate Authority Administration: Advanced Topics", in the section OracleAS Certificate Authority and High-Availability Features.

The `ocactl linksso` command does not require OCA service to be shut down, but it takes effect only after the SSO 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 -state ON
ocactl set -type TRACE -state ON
ocactl set -type LOG -state OFF
ocactl set -type TRACE -state OFF
```

Data generated by the first two commands above is stored in the following locations:

- LOG data goes into the OCA repository

- TRACE data goes into the operating system file oca.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 or Oracle Application Server Certificate Authority stops operating.

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 OCA.

The following summary gives an overview of the acquisition and import process:

As the administrator of OracleAS Certificate Authority, you obtain the Sub CA 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.

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 wallet.

OCA's administration tool has an import option to enable the administrator to import that stored SubCA wallet and certificate into an OCA instance running as a Subordinate CA. The import operation includes an automatic change of encryption and location to fit OCA's standard operations. The following sections of this Appendix describe all these steps:

- Generating a Sub CA Wallet
- Installing and Using the New Sub CA Wallet
- Generating CA SSL and CA SMIME Wallets for a Sub CA

Generating a Sub CA Wallet

The following steps tell you, as OCA administrator, how to generate a Sub CA wallet from the Oracle CA:

1. Use Oracle Wallet Manager or a third-party tool to generate a PKCS#10 request.
2. Using OCA'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 7, "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 RequestsChapter 3, "Introduction to OCA 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 **Download CA Certificate**. An **Advanced** button will appear. Clicking **Advanced** will show the CA certificate along with the trust points, if any, displayed below 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 Wallet Manager in the *Oracle Application Server 10g Security 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 wallet to a file destination of your own choice.

Installing and Using the New Sub CA 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 OCA as described in Appendix D, "Extensions".

> **See Also:** Subordinate CA Certificates in Chapter 7, "End-User Interface of the Oracle Application Server Certificate Authority".

1. Install Oracle Application Server Certificate Authority, which will create an OCA repository, create the password store, and create the Root CA wallet and the CA SSL wallet.
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 wallet using the following command:

   ```
    ocactl importwallet -type SUBCA
   ```

   > **See:** Appendix A, "Command-Line Administration" for details. For example, while importing the Sub CA wallet, ocactl ensures that the correct bits are set for the right extensions. The wallet can function as a Sub CA 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 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 OCA Administrator.

b. Fetch the new Sub CA’s certificate, private key, and serial number from that wallet, and store them in the OCA repository.

This operation overwrites the corresponding earlier records in the OCA 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:

4. 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

5. Convert this wallet to SSO format using the following command

   ```
   ocactl convertwallet -format SSO
   ```

6. Start HTTP Server by using the command-line tool `opmn`.
7. Start OC4J using the same command-line tool.
8. Start Oracle Application Server Certificate Authority, which will now use the new Sub CA certificate for signing all future certificate requests.

> **See Also:** The *Oracle Application Server 10g Security Guide*, particularly the Appendix on Managing PKI Credentials with Oracle Wallet Manager.

Configuring an OCA 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 wallet from the Root CA and install that Sub CA in another OCA installation. The parent organization with the Root CA wallet can issue Sub CA 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 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 wallet from Oracle Application Server 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 OCA.

   > **See Also:** The *Oracle Application Server 10g Security Guide*, particularly the Appendix on Managing PKI Credentials with Oracle Wallet Manager.

2. Using the Server/Sub CA enrollment form of the user interface described in Chapter 7, 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 3, 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 **Download CA Certificate**, which will show the CA certificate along with its ancestors, if there are any.

   > **See Also:** Server/SubCA Certificates Tab in Chapter 7, "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:

7. 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.
8. Click **View Details** to view the Sub CA certificate in base64 format.
9. Copy that base64 format of the Sub CA certificate and import it into Oracle Wallet Manager as a user certificate.
10. Save the Sub CA 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 6's section entitled Regenerating the CA SSL and CA SMIME Wallets, the CA SSL wallet is generated during installation. It enables Oracle Application Server Certificate Authority to listen in HTTPS mode, and it can be regenerated if necessary, to re-establish secure communications. Circumstances requiring such regeneration include a wallet becoming compromised or corrupted, or the CA wallet being regenerated, or a new Sub CA certificate being imported.

Generating the Sub CA SSL wallet is also done when OCA 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 OCA 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 not valid any more. The CA SMIME wallet must be generated to sign alerts and notifications when "Send SMIME E-Mails" is enabled in Notification page of Configuration Management in OCA 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 Oracle Application Server Certificate Authority.

Oracle Application Server Certificate Authority will now use the Sub CA certificate for signing certificate requests.

C

Known Troubleshooting Tips

This chapter describes a number of issues that can arise in the installation or administration of Oracle Application Server Certificate Authority.

The following sections of this Appendix describe how to deal with or work around those issues for the current release:

1. Prerequisite Issues and Warnings

   a. Issue: Failure of Key Pair Generation during Certificate Requests on Windows.

   c. Issue: Changing Passwords Must Use OCA's Commandline Tool ocactl

   b. Issue: Cannot Log in as Administrator after Logging in as Normal User

2. Browser Issues

   a. Issue: Browser issues a warning if the CA SSL Server's CN is not identical to the machine name.

   b. Issue: Browsers use only the first (rightmost) CN component

   c. Netscape

   - i. Issue: Only one certificate appears in the popup window, though multiple certificates are available.
   - ii. Issue: Browser continues to ask if CA certificate is trusted.
   - iii. Issue: "Certificate is expired" warning appears.
   - iv. Issue: SubCA and CA SSL client certificates are listed.

   d. Internet Explorer (IE)

   - i. Issue: "Page can not be displayed" Message

1. Prerequisite Issues and Warnings

a. Issue: Failure of Key Pair Generation during Certificate Requests on Windows.

For Windows client machines, this operation requires NT to have Service pack 5 or above.

What to Do

- Visit Microsoft’s website and download the necessary upgrades for your configuration

Always use `ocactl` to change any password related to OCA. Never use any other tool.

b. Issue: Cannot Log in as Administrator after Logging in as Normal User

If you first log in to OCA as a normal user via 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 admin. The SSL session established between OCA and you as a non-admin user remains active; your enrollment does not change your SSL session.

What to Do
To log on as web admin, you must

1. Enroll as web admin,
2. Exit your browser, and
3. Login as web admin, by choosing your web admin certificate for authentication.

c. Issue: Changing Passwords Must Use OCA's Commandline Tool `ocactl`

It may occasionally be desirable or advisable to change the passwords used for the CA SSL wallet, the OCA internal repository, or the OCA administrator. If any tool other than `ocactl` is used to change any of these passwords, OCA will stop working.

2. Browser Issues

Some symptoms may arise only when you are using a certain type or level of browser. This section describes the presently known browser-related issues.

Issue: Browser issues a warning if the CA SSL Server's CN is not identical to 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.

Issue: Browsers use only the first (rightmost) CN component

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). This certificate is listed in the popup for SSL Mutual Authentication as "users's", in both MicroSoft's Internet Explorer and Netscape (4.7x and 7.x).

Netscape

i. Issue: Only one certificate appears in the popup window, though multiple certificates are available.

Netscape 4.79 shows only the latest certificate in this popup window.

What to Do

- Alter the order of certificates so that the one that you want to use is the last certificate on the list.

ii. Issue: Browser continues to ask if CA certificate is trusted.

Netscape 4.7x versions do not automatically trust the CA certificate; they require the user to state explicitly that the CA certificate is to be trusted. Until that is done, Netscape does not assume it is trusted.

What to Do

- See Importing a Certificate to Your Browser in Chapter 7, "End-User Interface of the Oracle Application Server Certificate Authority".

iii. Issue: "Certificate is expired" warning appears.

If the time zone of the client is behind that of the server, there can be a period of time in which Netscape might issue a 'certificate is expired' warning. The reason is that the CASSL certificate is not yet valid in the user's time zone.

What to Do

- The problem should resolve itself in a relatively short period of time, depending on the time zone differential.

iv. Issue: SubCA and CA SSL client certificates are listed.

Netscape 7.x browser users can face this anomaly: 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. Select the certificate appropriate to the CA in use for this SSL site.

Internet Explorer (IE)

i. Issue: "Page can not be displayed" Message

These intermittent errors can arise while interacting in SSL mode. One example arises after logging in to SSO by name and password, but then changing authentication by choosing SSL. This error is a known IE bug.

What to Do

- Try to reload the page. If that isn't helping, exit from the current browser session, and then re-access Oracle Application Server Certificate Authority to try anew.

ii. Issue: Failure to import CRL to Browser

The IE button Import CRL to Browser does show the CRL for viewing, but it does not actually import the CRL into the browser.

What to Do

- Use the IE menus to choose the following command sequence:
  Tools -> Internet Options -> Content -> Certificates -> Import

iii. Issue: Message that a page contains both secure and non-secure information

In User Pages -> Manual Authentication -> Download 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.

iv. Issue: Opening online Help can generate a security alert.

When online help is opened while using OCA, 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.

What to Do

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.

3. Network Issues

a. Issue: Error message when logging on to OCA using SSO username/password

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 SSO server.

What to Do

- 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 following change is needed on the server side: the value of the directive OssoIpCheck in the SSO configuration file must be set to "off". To do so, 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, you must restart the Oracle HTTP Server by executing the following stop and start commands:

  ```
  $ORACLE_HOME/opmn/bin/opmnctl stopproc type=ohs
  $ORACLE_HOME/opmn/bin/opmnctl startproc type=ohs
  ```

b. Issue: "Network Error" message.

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.

What to Do

- You need to re-authenticate yourself to OCA by going to the Certificate Management tab and, when asked, choosing the Web Admin Certificate.

4. Certificate Issues

a. Issue: Importing user certificate does not import CA certificate on Netscape

An attempt to import a user certificate does not in fact do so.

What to Do

- 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.
- 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.

b. Issue: Inability to Access or Use the Certificate Management Tab

Attempts to access or use the Certificate Management facility fail.

What to Do

- Access to Certificate Management requires that your browser has imported a valid Web Administrator certificate. Thus 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 ... .

c. Issue: Administrator Needs to Work from a Different Machine

An Oracle Application Server 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 OCA Web Administrator.

What to Do

- 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: Choose Security->Certificates->Yours->choose the Web Admin Cert ->Export

- Importing the certificate on Netscape: 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

5. Single Sign-on (SSO) Issues

a. Issue: Name shown on an SSO certificate appears only as "User"

These certificates do not show the common name or DN. They are distinguishable only by having different certificate serial numbers.

What to Do

- Click on "View" to check the certificate serial number, and pick the certificate identified by the serial number you wish to use.

b. Issue: VBScript Error Message While Generating Keys

In SSO, 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.

What to Do

- Try again, being sure to click "Submit" only once and to wait until the certificate is returned.

c. Issue: "Page can not be displayed" Message in Internet Explorer

After logging in to SSO by name and password, but then changing authentication by choosing SSL, a known IE bug gives the "Page cannot be displayed error."

What to Do

- Try to reload the page. If that isn’t helping, exit from the current browser session, and then re-access Oracle Application Server Certificate Authority to try anew.

d. Issue: Going to the SSO login page in Internet Explorer can get a security warning dialog

What to Do

- This warning occurs due to switching from https to http. No action is needed.

6. Search Issues

a. Issue: Pressing "Enter" in search screens produces "Internal Error".

What to Do

- This error is a known Oracle Bug, #2224035 (Marlin). To initiate a search, use the GO button rather than pressing Enter.

7. Backup Protection Issues

a. Issue: Ensuring Recoverability of the OCA Internal Repository

Errors and unpredicted events can threaten the continuity of OCA operations.

What to Do

- Take a backup of the OCA repository periodically. Oracle Application Server commandline tools such as "export" can be used to save the OCA repository to a file. It can then be restored to the "same" database using the "import" tool.

8. General Issues

a. Issue: Pages taking too long to load, or hanging

Sometimes such delays can occur, possibly after Oracle Application Server Certificate Authority has been in operation for a substantial period.

What to Do

- Restart OCA's OC4J instance, which will return you to faster operations.

b. Issue: JAZN error when enrolling a new web administrator

After a web administrator certificate has been revoked, OHS and OCA's OC4J must be restarted before starting OCA and enrolling the new web administrator.

What to Do

- Start OHS and OCA's OC4J first, then start OCA, and then enroll the new web administrator.

c. Issue: No SMIME signing certificate in Outlook Express

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.

What to Do

- You will need to use Microsoft Outlook and not Outlook Express.

d. Issue: Browser warning about CA SSL Server's CN

If the CA SSL Server's CN is not identical to the machine name, this warning will arise.

What to Do

- You will need to make the CN and machine name the same.

D

Extensions

OracleAS Certificate Authority is compliant with the X.509 V3 and IETF's PKIX standards, and supports the following extensions:

1. OCA's CA certificates contain the following extensions

   a. Basic Constraints Extension: Critical

      - CA flag set to true
      - PathLength for root (self-sign) certificate is hardcoded to 3.
      - PathLength for sub CA is between 0 and 2, depending on the pathlength of issuer's (upper CA) certificate.

   b. KeyUsage Extension: Critical

      The following bits are set on:

      - Digital Signature
      - Key Cert Sign
      - CRL Sign
      - Non-Repudiation

2. OCA's End-Entity SSL / Encryption Certificates

   a. Key Usage Extension: Non-Critical

      The following bits set on:

      - Digital Signature
      - Key Encipherment
      - Key Agreement

- Non-Repudiation

3. Code signing certificates

   a. Key Usage Extension: Non-Critical

      The following bits set on:

      - + Digital Signature

4. SMIME-Signing Certificates

   a. Key Usage Extension: Non-Critical

      The following bits set on:

      - Digital Signature
      - Data Encipherment
      - Non-Repudiation

E

Glossary

Table E–1 lists the glossary items and their definitions. Related topics are also listed: those that are in the table are links; others are accessible through the index.

Table E–1 Definitions for Terms Used in OracleAS Certificate Authority

Term	Meaning	Related Topics
Authentication	Authentication is a security measure that establishes the validity of a transmission, message, or originator. It is a means of verifying an individual's authorization to receive specific information.	
Certificate	A certificate is a digital representation that ties the user's identification to the user's public key in a trusted bond. The certificate identifies the certificate authority issuing the certificate, the names of the person, process, or equipment that is the user of the certificate, the user's public key, and is digitally signed by the certificate authority.	Certificate Authority, Code Signing Certificates
Certificate Authority	A certificate authority (CA) is an authority trusted by one or more users to issue and manage X.509 public key certificates and certificate revocation lists.	Certificate, Certificate Revocation List
Certificate Revocation List	Often abbreviated as CRL, this is a list of revoked certificates published by a certificate authority. Certificates can be revoked before their expiration date for a variety of reasons. For example, a certificate may be revoked if the private key is compromised. If automatic CRL generation is enabled and an interval is specified, then Oracle Certificate Authority automatically generates a CRL each time the specified interval elapses.	Certificate, Certificate Authority, Setting Up and Enabling Automatic CRL Generation
Client Secure Socket Layer Certificates	Used to identify clients to servers through the secure socket layer (client authentication).	Certificate

Table E–1 Definitions for Terms Used in OracleAS Certificate Authority

Term	Meaning	Related Topics
Code Signing Certificates	Used to identify signers of Java code, JavaScripts, or other signed files.	Certificate
Digital Signatures	A digital signature is an electronic analog of a written signature that is generated from a message prior to its dispatch and can be used to verify to the recipient that the message was signed by the originator. Digital signature systems require a two-step process: 1. A hash algorithm condenses data into a message digest. (Public key encryption is not used for encrypting large amounts of data.) 2. 2.The message digest is encrypted with the originator's private key. 3. The recipient re-creates the message digest from the received message, uses the public key to decrypt the digital signature, and compares the results. Digital signatures are a particular application of public key encryption.	Key Pair, Public Key Encryption
Directory	The directory provides a repository from which users can obtain public key certificates for themselves and for other users and where they can verify that certificates have not been revoked.	
Distinguished Name	Distinguished names are used to give the holder of certificates unique identifiable characteristics that distinguish one certificate from other certificates. Example: cn=Sara Will, ou=Sales, o=Acme Corporation, c=AU Where cn stands for common name, ou stands for organizational unit, o stands for organization, and c stands for country. Note Domain component entries in the DN can be used in addition to (or to replace) entries for organization or country. Examples include dc=be (for belgium) or dc=us (for United States) or dc=oracle or dc=com.	Domain Component Attributes

Table E–1 Definitions for Terms Used in OracleAS Certificate Authority

Term	Meaning	Related Topics

Domain Component Attributes	The domain component attribute can be used in constructing a DN from a domain name. For example, an organization named "Acme, Inc.", having registered the domain name "acme.com", could deploy a directory following this naming plan by proceeding as follows: it would construct the DN
	dc=acme, dc=com
	from its domain name, and then use this DN as the root of its subtree of directory information.
	The DN itself can identify a directory organization object representing information about the organization, so that subordinates of the DN are directory objects related to the organization. The domain component attribute can be used to name subdivisions of the organization, such as organizational units and localities.
	Acme, for example, might use the domain names "corporate.acme.com" and "richmond.acme.com" to construct the names
	dc=corporate, dc=acme, dc=com
	dc=richmond, dc=acme, dc=com
	under which to place its directory objects. Such subdivisions of the organization could also be assigned RDNs using the conventional X.500 naming attributes, such as
	ou=corporate, dc=acme, dc=com
	l=richmond, dc=acme, dc=com.
		Distinguished Name

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.	Certificate, Public Key
Key Pair	A key pair includes two mathematically related keys where one key can be used to encrypt a message that can only be decrypted by using the other key.	Private Key, Public Key
Policy Precedence	Policies are applied to incoming requests in the order that they are displayed on the main policy page. When the Oracle Certificate Authority 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. Note: only enabled policies are applied to incoming requests.	

Table E–1 Definitions for Terms Used in OracleAS Certificate Authority

Term	Meaning	Related Topics
Predicates	A policy predicate is a logical expression that can be applied to a policy to limit how it is applied to incoming 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" For detailed information about predicates and predicate expression syntax, see Chapter 5, "Managing Policies in Oracle Application Server Certificate Authority".	
Private Key	The private key is the key of a signature key pair used to create a digital signature, or the key of an encryption key pair used to decrypt confidential information. In both cases, the private key must be kept secret.	Key Pair, Public Key
Public Key	The public key is the key of a signature key-pair used to validate a digital signature or the key of an encryption key-pair used to encrypt confidential information. In both cases, this key is made publicly available.	Key Pair, Private Key
Public Key Encryption	Public key encryption involves two corresponding keys, commonly known as a key-pair. One of these keys is private (private key) and the other key is widely known (public key). The owner needs to know the private key, and the public key is available and known to anyone. Since only one party needs to know the private key, it does not need to be transmitted between parties. Therefore, the private key is never at the risk of interception. Knowledge of the public key by a third party does not compromise the security of data transmission. The owner of the private key can digitally sign a document by encrypting a unique digest of the message with the private key. The source of the document can be verified by decrypting the digital signature with the public key and comparing it to the digest of the message.	Key Pair, Private Key, Public Key
Public Key Infrastructure	Public key infrastructure is a system of hardware, software, policies, and people that can provide a suite of information security assurance that are important in protecting sensitive communications and transactions.	
Root CA	In a hierarchical public key infrastructure, the root certificate authority (CA) is the CA whose public key serves as the most trusted datum for a security domain.	Certificate Authority, Public Key

Table E–1 Definitions for Terms Used in OracleAS Certificate Authority

Term	Meaning	Related Topics
S/MIME	S/MIME (Secure Multipart Internet Mail Extensions) is a protocol that adds digital signatures and encryption to Internet MIME messages.	
Subordinate CA	In a hierarchical public key infrastructure, 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.	Certificate Authority
X.509	The International Telecommunications Union Telecommunication Standardization Section recommendation that defines a framework for the provision of authentication services under a central control paradigm represented by a directory. It is the most widely used standard for defining the format for digital certificates.	

Index