Oracle® Security Server Guide

Release 2.0.3

June, 1997

Part No. A54088-01

ORACLE®

Enabling the Information Age™

Oracle Security Server Guide

Part No. A54088-01

Release 2.0.3

Primary Author: Kendall Scott

Contributing Authors: Mary Ann Davidson, Gilbert Gonzalez, John Heimann, Patricia Markee, Rick Wessman

Contributors: Quan Dinh, Jason Durbin, Gary Gilchrist, Wendy Liau, Bob Porporato, Andy Scott, Andre Srinivasan, Juliet Tran, Sandy Venning

The Programs that this manual accompanies are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free.

The Programs that this manual accompanies contain data encryption routines which are subject to export regulations, and which may be subject to usage restrictions in your country. By opening this package, you agree to comply fully with all United States government laws and regulations to assure that neither the Programs, nor any direct product thereof, are exported, directly or indirectly, in violation of United States law. You further agree to comply fully with any applicable local laws regarding the use of these Programs. These Programs may not be transferred outside the country where delivery is taken or transferred, sold, assigned, or otherwise conveyed to another party without Oracle's prior written consent.

Preface

Oracle Security Server Guide describes the features, architecture, and administration of the Oracle Security Server. The Oracle Security Server is a security product, based on public-key cryptography, that supports centralized authorization and distributed authentication in an Oracle network environment.

The Oracle Security Server, release 2.0.3, provides:

- a centralized authorization and distributed authentication framework that is based on public-key cryptography and that includes the Oracle Security Adapter and the Oracle Security Server Repository. This framework supports X.509 version 1 certificates, an industry-standard method of authentication.
- the Oracle Security Server Manager, a management tool that an administrator uses to configure the framework.
- the Oracle Cryptographic Toolkit, a programmer's toolkit. This toolkit contains a set of application programming interfaces (APIs) that enable application programs to access cryptographic functions, such as generating and verifying digital signatures. These APIs, available via the Oracle Call Interface (OCI) and PL/SQL, can be used to provide assurance to a wide variety of applications, such as electronic mail and electronic commerce. For more information on the Oracle Cryptographic Toolkit, see the *Oracle Cryptographic Toolkit Programmer's Guide*.

Intended Audience

Oracle Security Server Guide is designed as the basic document to help security system administrators understand, manage, and configure the Oracle Security Server. *Oracle Security Server Guide* is available in HTML format for viewing through a Web browser. It can also be ordered in hardcopy (paper) format.

Structure

This manual contains four chapters, a glossary, and a bibliography:

Chapter 1	Describes basic concepts associated with the Oracle Security Server.
Chapter 2	Provides a description of the architecture and operation of the Oracle Security Server.
Chapter 3	Details how a security administrator initializes the Oracle Security Server.
Chapter 4	Details how the security administrator uses the Oracle Security Server Manager to define elements to the Oracle Security Server.
Glossary	Defines security-related terms that appear within this manual.
Bibliography	Provides details for the external references cited within this manual.

Conventions

The following conventions are used in this manual:

Convention	Meaning
boldface text	Boldface type in text is used for terms being defined, names of pulldown menus, pushbuttons and field names on windows, and path (directory) information.
italic text	Italic type in text is used for the values of fields,the names of subareas on windows and options on pulldown menus, and the titles of other manuals.
angle brackets <>	Variable names appear inside angle brackets.
square brackets []	Optional items appear inside square brackets.

Related Documents

For more information, see the following manuals:

- *Oracle® Advanced Networking Option™ Administrator's Guide*
- *Oracle8 Server Distributed Database Systems*
- *Oracle8 Server SQL Reference*
- *Oracle Cryptographic Toolkit Programmer's Guide*
- *Programmer's Guide to the Oracle Call Interface*

Your Comments Are Welcome

We value and appreciate your comments as an Oracle user and reader of the manual. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. At the back of each of our printed manuals is a Reader's Comment Form, which we encourage you to use to tell us what you like and dislike about this manual or other Oracle manuals. If the form is not available, please use one of the following addresses or the FAX number.

Oracle Network Products Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood City, CA 94065
U.S.A.

E-Mail: *ossdoc@us.oracle.com*

FAX: 415-506-7200

Contents

1 Oracle Security Server Concepts

Figures

1

Oracle Security Server Concepts

This chapter describes basic concepts associated with the Oracle Security Server. The chapter includes the following sections:

- Introduction
- Basic Concepts
- Oracle–Specific Features
- Global Intranet Authentication and Authorization

Introduction

The Oracle Security Server is a security product that supports centralized authorization and distributed authentication in an Oracle environment. **Authentication** provides assurance that the alleged identity of a party who wishes to access one or more Oracle database servers is valid. **Authorization** assures that a given party can only operate according to privileges that have been defined for that party by an administrator.

The Oracle Security Server is bundled with Oracle8 Server for use on any platform that supports that product. However, the Oracle Security Server can be used with an Oracle7 Server as well.

Basic Concepts

Cryptography

Introduction

Cryptography is the science of providing security for information through the reversible transformation of data. It is a science of great antiquity. (Julius Caesar used a simple letter substitution cipher that still bears his name.) The development of digital computing revolutionized cryptography, and made today's highly complex and secure cryptographic systems possible.

A modern cryptographic system contains an algorithm and one or more keys. A **cryptographic algorithm** (also known as a **cipher**) is a general procedure for transforming data from **plaintext** (a usable, readable form) to **ciphertext** (a protected form) and back again. The former process is called **encryption**; the latter, **decryption**. The **keys** are variable parameters of the algorithm. In order to transform a given piece of plaintext into ciphertext, or ciphertext into plaintext, one needs both the algorithm and a key.

Modern algorithms are designed so that a user who knows the algorithm and the ciphertext, but not the key, cannot easily derive the plaintext from the corresponding ciphertext. Normally, algorithms are widely distributed or even public, while knowledge of keys is limited to the fewest users possible, since knowledge of the key provides access to the data encrypted with that key.

If an algorithm is well–designed, the size of the key is an indication of the algorithm's **strength**, which is the difficulty an attacker would have deriving the plaintext from the ciphertext without prior knowledge of the key.

Private–Key Cryptography

Until relatively recently, cryptographic algorithms were designed so that the same key was used to both encrypt and decrypt data. Algorithms designed this way are referred to as "private–key," "secret–key," or "symmetric–key" algorithms.

As an example, if Alice and Bob wish to communicate, they must each know the secret key, and the key must be exchanged in such a way that its secrecy is preserved. If Bob and Steve also wish to communicate, they must obtain another secret key so that Alice cannot read their messages.

Prominent examples of secret–key algorithms include the *Data Encryption Standard* (*DES*), which the National Bureau of Standards (now the National Institute of Standards and Technology [NIST]) brought out in 1975, and the *International Data Encryption Algorithm* (*IDEA*), developed in 1990 by two men in Sweden.

There are certain problems associated with using secret–key cryptography in the enterprise. As the number of users (N) increases linearly, the number of possible "pairwise-secret" keys increases by a factor of N^2. This causes the management and distribution of keys to become overwhelming. To deal with this problem, most large systems provide centralized key servers from which users must retrieve a new key for each communications session if they wish to establish a secure session. These centralized private–key servers are often the "Achilles heel" of a communications system, since a single failure can compromise the entire system.

Public-Key Cryptography

In 1976, Whitfield Diffie and Martin Hellman proposed a new type of cryptographic algorithm, referred to as "public key," which greatly facilitates key distribution in a large user community.

In **public-key cryptography** (also known as "asymmetric" cryptography), the key used to encrypt plaintext into ciphertext is different from the key that decrypts ciphertext into plaintext. Each person gets a pair of keys: a **public key** and a **private key**. The public key is published, while the private key is kept secret.

The keys are related in that a message encrypted with the public key can only be decrypted with the corresponding private key, and a message encrypted with a private key can only be decrypted with the corresponding public key. Furthermore, the keys are designed so that the private key cannot, for all practical purposes, be deduced from the public key. For instance, cryptanalysis of the most famous public–key algorithm, *RSA*, requires the cryptanalyst to factor numbers that contain in excess of 100 digits each; the difficulty in factoring numbers of that magnitude is well–known in the computer science community.

Confidentiality

Public–key cryptography provides **confidentiality** or data secrecy. For example: If Alice wishes to send a message to Bob that only Bob can read, she encrypts the message with Bob's public key, and Bob subsequently decrypts the message with his private key. Since only Bob has the private key that can decrypt the message, only Bob can read it. Anyone else wishing to send an encrypted message to Bob must also use his public key for encryption.

Authentication

Public–key cryptography can also be used in **authentication** of senders of information. If Alice encrypts data with her private key, any other user can read it using Alice's public key, but no other user can duplicate Alice's encryption without access to Alice's private key.

Diffie and Hellman's paper [Diffie and Hellman] appeared in 1976; it is the original paper about public–key cryptography. Other good sources for information on this subject are RSA's Frequently Asked Questions document [RSA FAQ] (see **http://www.rsa.com/PUBS**) and Bruce Schneier's *Applied Cryptography* [Schneier] (see **http://www.counterpoint.com**).

Mixed Private/Public Key Systems

In a practical security system, private– and public–key algorithms are used together. Public keys are typically much larger than private keys (a DES private key is 56 bits, while an RSA public key is usually 512 or 1024 bits), and public–key algorithms are generally much slower than private–key algorithms.

In a **hybrid cryptosystem**, two parties who wish to communicate with each other use a public–key encryption algorithm to authenticate each other and a more streamlined private–key algorithm to transmit bulk data. The steps involved in this process include the following:

- The two parties agree on a common private–key encryption algorithm.
- Each party uses a computer tool to generate a public key and a private key.
- The sender and the receiver transmit their public keys to each other.
- The sender and the receiver each generate half of a random **session key**. This is a key that is used to encrypt and/or decrypt the data transmitted during one and only one communication session. (A communication session can consist of more than one transmission, but it usually has just one functional purpose and is relatively short in duration.)
- Each party uses the other party's public key to encrypt the session key half.

- Each party transmits its encrypted session key half to the other party.
- Each party uses its private key to recover the half of the session key that it did not generate.
- The two parties use the full session key with the private–key algorithm in exchanging data.

In addition to the speed advantages that this provides over public–key cryptography, it is also better than private–key cryptography on its own, because key management is simplified and the keys are more secure.

Benefits of Public-Key Cryptography

Public–key cryptography simplifies key distribution by eliminating the need to share private keys. Holders of public keys can safely conduct business with parties whom they never see and with whom they had no previous relationship. In essence, the public–key encryption system becomes an effective substitute for face–to–face commerce.

Since private keys are only known to the owning party, public–key authentication eliminates the need for a server that manages the private keys for all the parties in a system. This eliminates all single points of failure, and considerably reduces and simplifies the management of keys. Keys can be used for longer periods of time than those used in secret–key encryption systems because private keys are never shared. Since the security for private keys is one of the most critical issues in any cryptographic system, simplifying private–key management not only simplifies the system, but it also makes it an order of magnitude more secure than previous security technologies.

Please note that although the Oracle Security Server uses cryptographic mechanisms to support authentication and authorization, it does not provide bulk encryption keys for data stream encryption. Data stream encryption is provided by the Oracle Advanced Networking Option encryption adapters (for example, RSA Data Security, Inc.’s *RC4)*. Refer to the *Oracle Advanced Networking Option Administrator’s Guide* for more information about encrypted data streams.

Digital Signatures

A **digital signature** is a quantity associated with a message that only someone with knowledge of an entity's private key could have generated, but which can be verified through knowledge of that entity's public key.

Digital signatures perform three very important functions:

- *integrity* — A digital signature allows the recipient of a given file or message to detect whether that file or message has been modified.
- *authentication* — A digital signature makes it possible to verify cryptographically the identity of the person who signed a given message.
- *nonrepudiation* — A digital signature prevents the sender of a message from later claiming that it did not send the message.

The process of generating a digital signature for a particular document type involves two steps.

First, the sender uses a **one–way hash function** to generate a **message digest**. This hash function can take a message of any length and return a fixed–length (say, 128 bits) number (the message digest). The characteristics that make this kind of function valuable are as follows:

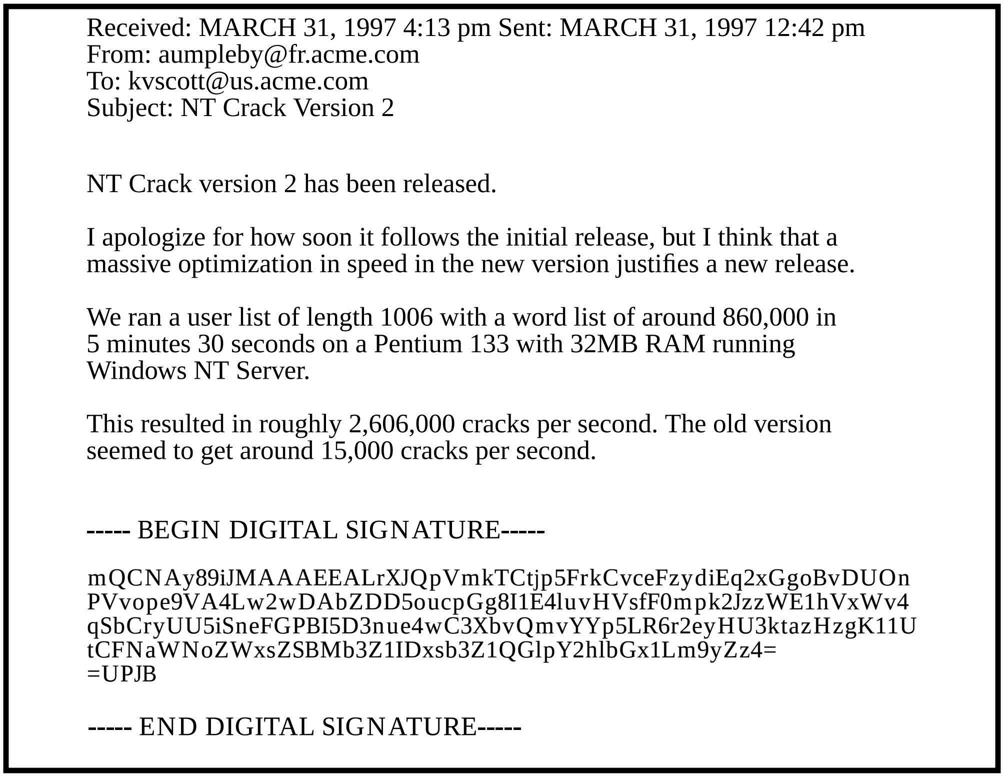
- Given a message, it is easy to compute the associated message digest.
- Given a message digest, it is hard to determine the message.
- Given a message, it is hard to find another message for which the function would produce the same message digest.

Second, the sender uses its private key to encrypt the message digest.

Thus, to **sign** something, in this context, means to create a message digest and encrypt it with a private key.

Figure 1–1 shows a typical E–mail message and what the associated digital signature might look like.

Figure 1–1 Message With Attached Digital Signature



Received: MARCH 31, 1997 4:13 pm Sent: MARCH 31, 1997 12:42 pm
From: aumpleby@fr.acme.com
To: kvscott@us.acme.com
Subject: NT Crack Version 2

NT Crack version 2 has been released.

I apologize for how soon it follows the initial release, but I think that a massive optimization in speed in the new version justifies a new release.

We ran a user list of length 1006 with a word list of around 860,000 in 5 minutes 30 seconds on a Pentium 133 with 32MB RAM running Windows NT Server.

This resulted in roughly 2,606,000 cracks per second. The old version seemed to get around 15,000 cracks per second.

----- BEGIN DIGITAL SIGNATURE-----

mQCNAy89iJMAAAEEALrXJQpVmkTCtjp5FrkCvceFzydiEq2xGgoBvDUOn
PVvope9VA4Lw2wDAbZDD5oucpGg8I1E4luvHVsfF0mpk2JzzWE1hVxWv4
qSbCryUU5iSneFGPBI5D3nue4wC3XbvQmvYYp5LR6r2eyHU3ktazHzgK11U
tCFNaWNoZWxsZSBMb3Z1IDxsb3Z1QGlpY2hlbGx1Lm9yZz4=
=UPJB

----- END DIGITAL SIGNATURE-----

The receiver of a message can **verify** that message via a comparable two–step process:

- Apply the same one–way hash function that the sender used to the body of the received message. This will result in a message digest.
- Use the sender’s public key to decrypt the received message digest.

If the newly computed message digest matches the one that was transmitted, the message was not altered in transit, and the receiver can be certain that it came from the expected sender.

Certification Authority (CA)

A **certification authority** (**CA**) is a trusted entity that certifies that other entities are who they say they are.

The CA is something of an electronic notary service: it generates and validates electronic IDs, in the form of certificates (see the following section) that are the equivalent of driver’s licenses and passports. The CA uses its private key to sign each certificate; an entity that receives a certificate can trust that signature just as a person in real life can trust the written signature of a notary.

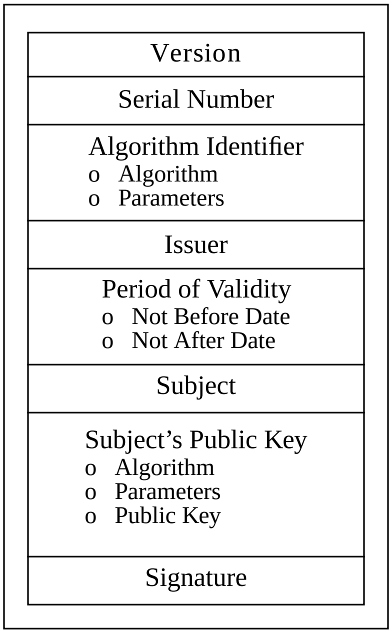
Certificates

A **certificate** is a message, signed by a CA, stating that a specified public key belongs to someone or something with a specified name.

Certificates prevent someone from using a phony key to impersonate a party, and also enable parties to exchange keys without contacting a CA for each authentication. Distributing keys in certificates is as reliable as if the keys were obtained directly from the CA. Certificate–based authentication works even when the security database server is temporarily unavailable.

Figure 1–2 shows the format of a typical certificate.

Figure 1–2 Certificate



The elements of this certificate are as follows:

- **Version** is *0* or *1*. (This is *0* within Oracle Security Server certificates. See the subsection “Oracle Security Server Certificates,” which appears later in this chapter, for more information.)
- **Serial Number** is the unique identifier for a given certificate.
- **Algorithm Identifier** identifies which cryptographic algorithm the CA used to sign the certificate and also provides any necessary parameters.
- **Issuer** is the name of the CA.
- **Period of Validity** indicates the date range over which the certificate is valid. This is the range between the date of creation and the expiration date specified by the person who requested the certificate.
- **Subject** is the name of the entity to which the certificate belongs.

- **Subject's Public Key** includes the public key for the given Subject, and also identifies which cryptographic algorithm the CA used to generate the key and provides any necessary parameters.
- **Signature** is the CA's digital signature.

A subject that receives a certificate belonging to another subject will try to verify that the CA issued the certificate, by applying that CA's public key to the **Signature**. If the receiving subject can understand the resulting text, the certificate was indeed signed by the CA, and the receiver can trust that the public key contained within the certificate belongs to the other subject.

Certificate Revocation Lists (CRLs)

A **certificate revocation list** (**CRL**) is a data structure, signed and timestamped by a CA, that lists all of the certificates created by that CA that have not yet expired but are no longer valid.

A certificate may be revoked in response to any of several events:

- The private key of the subject to which the certificate belongs has been compromised.
- The CA's private key has been compromised.
- The CA no longer wants to certify the given subject (because, for instance, the subject is a user who is no longer employed by the company).

A party retrieving a certificate from the CA can check one or more CRLs to see whether that certificate has been revoked. Note, though, that since checking a CRL incurs significant overhead, users may want to make these checks only for documents that are especially important, or they may want to limit themselves to periodic checks of CRLs.

Oracle–Specific Features

The Oracle Security Server conforms to a number of security industry and Oracle standards to facilitate interfaces with other products and systems.

Authentication

The Oracle Security Server supports a version of SKEME as its authentication protocol. (A paper about SKEME, written by Hugo Krawczyk of IBM [Krawczyk], is available at **http://www.research.ibm.com/security/publ.html**.)

Oracle Security Server Certificates

The Oracle Security Server supports X.509 version 1 certificates. (The *0* in the **Version** area of the certificate, as described in the section "Certificates" that appears earlier in this chapter, refers to version 1. Future releases of the Oracle Security Server will support version 3 certificates, which correspond with the value *1* for **Version**.)

Three documents define the standards for X.509 certificates.

- The original X.509 document [X.509] provides the formal definition of these certificates and the type of certificate revocation list (CRL) that the Oracle Security Server will be implementing in the future.
- The X.509 "amendments" document [X.509A] defines amendments to X.509 that future versions of the Oracle Security Server will address.
- The X.500 document [X.500] defines the directory service that serves as the "parent" of X.509.

You can order all of these documents from the International Telecommunications Union (ITU) directly; see **www.itu.ch/itudoc/itu-t/rec/x/x500up/**.

Oracle Security Server Digital Signatures

The Oracle Security Server uses the RSA cryptographic algorithm and RSA's Message Digest 5 (MD5) one–way hash function in generating and verifying digital signatures. These algorithms are implemented in software, using functions in the RSA TIPEM and BSAFE security toolkits. (See **http://www.rsa.com/rsa/PRODUCTS/TIPEM/** and **http://www.rsa.com/rsa/prodspec/bsafe/bsafe_3_0_f.htm**, respectively.)

The default version of the RSA algorithm is the 512–bit US–exportable version. Versions that use larger key sizes are available to eligible customers in accordance with applicable export and import regulations. MD5 produces a 128–bit hash value.

Two of the Public Key Cryptography Standards (PKCS) that RSA has defined are relevant to this discussion. PKCS #1 [PKCS1] describes a method for RSA encryption and decryption that is meant for use in conjunction with digital signatures, and also describes the syntax associated with the combination of RSA and MD5. PKCS #7 [PKCS7] describes the general syntax for data that may be signed with a digital signature. Both of these specifications are available at **www.rsa.com/PUBS/**. Ron Rivest's original paper about MD5 [MD5] contains technical details about that function.

Distinguished Names (DNs)

The Oracle Security Server supports the X.509 standard for fully–qualified **distinguished names** (**DNs**) in certificates.

A fully–qualified DN is a string that uniquely identifies a subject. This string may also define a path. The subject names contained within X.509 certificates are defined by the X.500 standard.

The Oracle Security Server limits the syntax of DNs so that certificates conform to a more restricted format, as defined by the following template:

DN = ([*Country,*] [*Organization,*] [*OrganizationUnit,*] [*State,*] [*Locality,*] *CommonName*)

Within this template, each DN must have a Common Name, and all of the other values are optional.

Note: The order in which these values appear within a DN is important with regard to defining global users (see "Authorization of Entities" later in this chapter) to an Oracle8 Server.

Table 1–1 provides an example of the information that one would enter in defining a DN for an entity that will be doing business with the Oracle Security Server.

Table 1–1 User-Entered Information for Certificates

FIELD NAME	USER-ENTERED INFORMATION
Country (C)	US
Organization (O)	Oracle Corporation
Organizational Unit (OU)	Network Management Products
State (ST)	California
Locality (L)	Belmont
Common Name (CN)	Lisa

Public/Private Key Pairs

The Oracle Security Server generates public/private key pairs using an RSA Data Security Inc. TIPEM library function. (See **http://www.rsa.com/rsa/PRODUCTS/TIPEM/**.)

Global Intranet Authentication and Authorization

The Oracle Security Server enables the use of public–key cryptographic technologies for Oracle and non–Oracle products. This technology provides:

- centrally defined identities, certificates, and roles—all of which enhance the support of single sign–on—and centralized administrative control over the generation and revocation of private keys and certificates for subjects
- distributed authentication of entities to each other involving X.509 certificates
- centralized authorization of users acting as "global" users to perform "globally identified" roles

The combined effect of these features is to enhance the security of any system. In particular, it enhances the security of those distributed systems that cannot control the number of users who can sign on to the system.

Identities, Certificates, and Roles

The Oracle Security Server enables an administrator to define identities for many types of subjects, including users, database servers, and Oracle WebServers. These identities, along with public keys, are captured in certificates that, used in conjunction with private keys, allow entities to authenticate themselves to each other using public–key cryptography (see "Authentication of Entities" below). Certificates can be revoked for entities that no longer belong to the enterprise.

The administrator can also define roles (collections of privileges) that can be used across databases (see "Authorization of Entities" below).

The Oracle Security Server also supports the implementation of single sign-on by replacing password authentication with certificate authentication.

The uniform management of enrollment and authorization of entities in large enterprises significantly improves the scalability of large distributed systems.

Authentication of Entities

Authentication provides assurance that the alleged identity of a party who wishes to communicate with another party over a network is valid.

Once a certificate has been assigned to an entity, that entity can use its certificate to authenticate itself to other subjects with which it wishes to communicate. For instance, an Oracle8 Server can find out with a high degree of certainty that a given user is who she says she is, while the user can be sure that she is communicating with the correct server.

Authorization of Entities

Authorization assures that a given entity can only operate according to privileges that have been defined for that entity, in the context of the Oracle Security Server, by an administrator.

Global users are users who need access to more than one Oracle8 Server using one set of credentials. **Global roles** (also known as **globally identified roles**) are roles that global users perform across Oracle8 Servers. The Oracle Security Server maintains the mapping of global users in a distributed Oracle8 enterprise to the globally identified roles that these users may perform for each database within that enterprise. (Note that the meaning of a globally identified role with regard to a specific Oracle8 Server remains the responsibility of that database's DBA.) See the *Oracle8 Server Distributed Database Systems* manual for more information about global users and global roles.

2

Oracle Security Server Architecture and Operation

This chapter provides a description of the architecture and operation of the Oracle Security Server. The sections and subsections within this chapter include:

- Oracle Security Server Architecture
- Oracle Security Server Operation

Oracle Security Server Architecture

The Oracle Security Server consists of the following major components:

- Oracle Security Server Manager
- Oracle Security Server Repository
- Oracle Security Server Authentication Adapter

The combination of the Oracle Security Server Manager, the security administrator (SA) who uses that tool, and the Oracle Security Server Repository forms the Oracle Security Server's implementation of a certification authority (CA).

Oracle Security Server Manager

A person uses the Oracle Security Server Manager, an application that runs in the Oracle Enterprise Manager framework, to administer the Oracle Security Server Repository. This application provides a graphical user interface (GUI) that an administrator can use to define and maintain information about identities and the authorizations granted to those identities on the databases within the enterprise.

The Oracle Security Server Manager runs under Windows NT 4.0 or Windows 95 on "low–end" machines, such as 486s, as well as on large–scale distributed PC networks.

Oracle Security Server Repository

The Oracle Security Server Repository is an Oracle7 Server (release 7.3.2 or higher) or Oracle8 Server that contains the data that an administrator enters using the Oracle Security Server Manager, as well as other data such as encrypted private keys. This repository also acts as the primary force behind the certification authority (CA) for the Oracle Security Server: it generates and stores certificates in response to administrator requests. responds to requests for information about certificate expirations and revocations, and stores requests for certificates posted from Oracle WebServers.

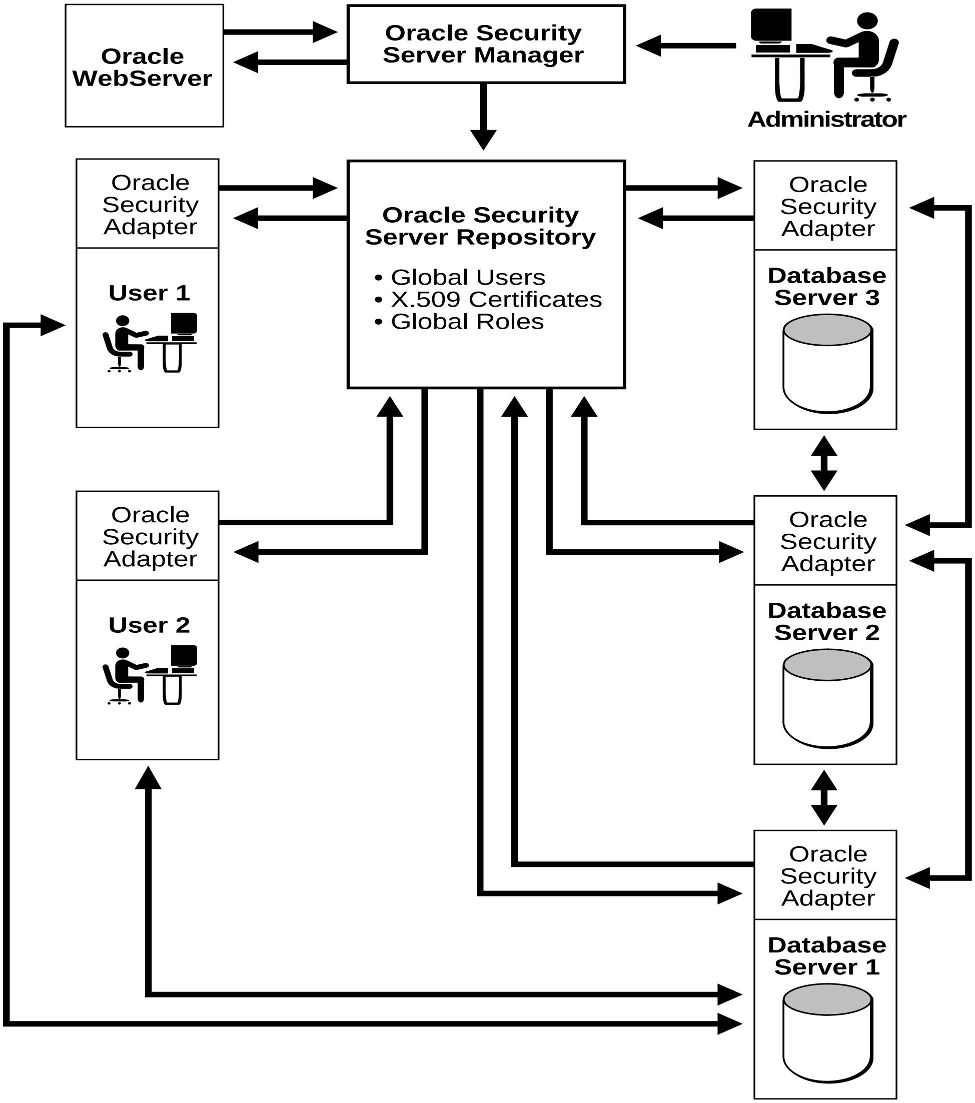
Oracle Security Server Authentication Adapter

The Oracle Security Server Authentication Adapter provides an interface from a Net8 client or an Oracle7 or Oracle8 database server to the Oracle Security Server Repository. This adapter allows Oracle products to request, obtain, and use certificates created by the Oracle Security Server CA. The adapter also queries the Oracle Security Server Repository for certificate status and authorization data.

Oracle Security Server Operation

Figure 2–1 illustrates the relationships among the components of the Oracle Security Server and the relationships among outside entities and these components.

Figure 2–1 Oracle Security Server Operations



If an Oracle WebServer is present in an enterprise, it may request the creation of identities and certificates within the Oracle Security Server. The administrator fulfills these requests using the Oracle Security Server Manager.

The Oracle Security Server Manager accesses the Oracle Security Server Repository using the version of SQL*Net or Net8 distributed with the Oracle Enterprise Manager. The Oracle Security Server Authentication Adapters and the Oracle Security Server Repository also communicate using SQL*Net / Net8.

Figure 2–1 indicates that authentication occurs between subjects by way of their Oracle Security Server Authentication Adapters. The steps involved in this **mutual authentication** process, in which one subject is acting as the *client* and the other is acting as the *server*, include the following:

1. The client sends a copy of its certificate to the server. The server responds by sending its certificate to the client.
2. Each subject uses the CA's public key to verify that the CA indeed signed the given certificate, and then extracts the identity and public key of the other subject.
3. Each subject checks with the CA to make sure that the certificate of the other subject has not expired or been revoked.
4. Each subject generates a random **nonce**, a binary value that is used only once, then uses the other subject's public key to encrypt that nonce and sends the encrypted nonce to the other subject.
5. Each subject uses its private key to decrypt the nonce that it received from the other party.
6. Each subject combines the nonce it received with the one it generated to create a hash key.
7. Each subject uses that key with the MD5 algorithm (see the section "Digital Signatures“ within Chapter 1) to generate a hash of the combination of the two nonces and the client's and server's identities, and then sends that hash to the other subject.
8. If each subject discovers that the hash it received matches the hash it sent, then both client and server are assured that the other subject is authentic. The server then retrieves, from the Oracle Security Server Repository, the roles that the client is authorized to perform.

3

Installing and Configuring the Oracle Security Server

This chapter details how database administrators (DBAs) and a security administrator perform the tasks involved in initializing the Oracle Security Server, including:

- Oracle Security Server Repository Dependencies
- Defining Global Users and Global Roles to Oracle8 Servers
- Installing the Oracle Security Server Repository
- Constructing the Oracle Security Server Repository
- Configuring Oracle Security Adapters on Clients and Servers
- Installing Wallets at Clients and Servers
- Removing the Oracle Security Server Repository

Oracle Security Server Repository Dependencies

In order for you to use a given database as an Oracle Security Server Repository, that database must be running Oracle7 Server release 7.3.2 or higher, or Oracle8 Server, on any platform that Oracle supports.

Before proceeding with this installation, you must also make sure that SQL*Net release 7.3.2 or higher, or Net8 release 8.0.2 or higher, is running on the given database.

Defining Global Users and Global Roles to Oracle8 Servers

It is recommended that global users and global roles be defined to Oracle8 Servers before those users and roles are identified to the Oracle Security Server. The DBA associated with each relevant server should follow these steps, using the Security Manager feature of Oracle Enterprise Manager:

1. Define each global user using the following syntax:

   CREATE USER *user* IDENTIFIED GLOBALLY AS ‘C=*country*, O=*organization*, OU=*organization_unit*, ST=*state*, L=*locality*, CN=*user*’

   Of the items that appear between the single quotes, only **CN** is mandatory.

   See the *Oracle8 Server SQL Reference* for more information about the CREATE USER command.

2. Define each global role using the following syntax:

   CREATE ROLE *role* IDENTIFIED GLOBALLY

   See the *Oracle8 Server SQL Reference* for more information about the CREATE ROLE command.

Installing the Oracle Security Server Repository

A DBA should perform the following steps to configure an Oracle database to contain the Oracle Security Server Repository:

1. Launch Oracle Enterprise Manager.
2. Install **Oracle Security Server Manager 2.0.3**.

A new program group named **Oracle Security Server** appears on your desktop in response.

3. Launch the **Create Security Server** program from that program group.

The Database Login Information Window appears in response.

4. Use the Database Login Information window to define the database that will contain the Oracle Security Server Repository.

   a. Type *system* in the **Username** field.

   b. Type the password that you wish to define for use by the Oracle Security Server administrator, in the **Password** field. This password should contain at least eight characters; at least one of these characters should not be alphanumeric.

   c. Type the name of the database on which the Oracle Security Server Repository will reside, in the **Service** field.

   d. Click the **OK** button.

A confirmation window appears in response. This window will ask you to confirm that you want the Oracle Security Server Repository to reside on the specified database.

5. Click the **OK** button on the confirmation window.

Installing the Oracle Security Server Repository creates a new username called "oracle_security_service_admin." The oracle_security_service_admin user has read / write access to data in the Oracle Security Server Repository. You defined the password for this username within Step 4 of the procedure described above.

> **Note**: Only one oracle_security_service_admin user can connect to the Oracle Security Server Repository at a time.

6. Launch the **Oracle Security Server Manager** program from the **Oracle Security Server** program group.

The Login Information window appears in response.

7. Log in to the Oracle Security Server Manager, using the Login Information window.

   a. Type *oracle_security_service_admin* in the **Username** field.

   b. Type the password you defined in Step 4, in the **Password** field.

   c. Type the service name you defined in Step 4, in the **Service** field.

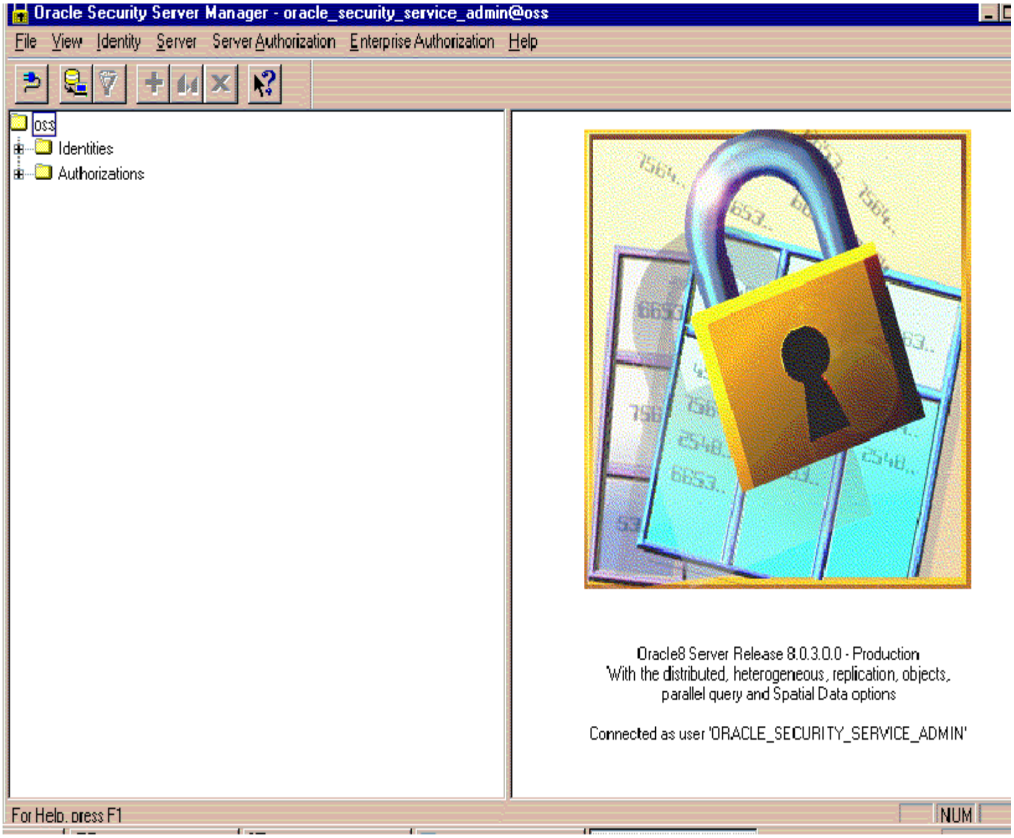
   d. Click the **OK** button.

A confirmation window appears in response. This window will ask you to confirm that you want to establish a certificate authority (CA) in connection with the new Oracle Security Server Repository.

8. Click the **OK** button on the confirmation window.

The Oracle Security Server Manager window (Figure 3–1) appears in response.

Figure 3–1 Oracle Security Server Manager Window



Oracle8 Server Release 8.0.3.0.0 - Production
With the distributed, heterogeneous, replication, objects, parallel query and Spatial Data options

Connected as user 'ORACLE_SECURITY_SERVICE_ADMIN'

Constructing the Oracle Security Server Repository

In order to construct your Oracle Security Server Repository, you need to become familiar with the Oracle Security Server Manager. Chapter 4, Using the Oracle Security Server Manager, describes all the tasks that appear within the procedure that follows, and also other tasks that you can perform.

Please note the following in connection with this procedure:

- The user of the Oracle Security Server Manager, a Security Administrator (SA), controls the CA. The Oracle Security Server implements the concept of a CA within the Oracle Security Server Repository.
- In this context, a **Server** is simply a representation of an Oracle8 Server.
- A **Server Authorization** is a representation of a role that has been “identified globally” at an Oracle8 Server.
- An **Enterprise Authorization** is a role that a global user can perform across multiple Oracle8 databases. An Enterprise Authorization can contain one or more Server Authorizations and/or one or more other Enterprise Authorizations.
- A user becomes a global user once he or she has an Identity defined in the Oracle Security Server Repository.

Follow these steps to construct your Oracle Security Server Repository:

1. To establish your certification authority:

   a. Select *Create* from the **Identity** pulldown on the Oracle Security Server Manager window.

   The Create Identity window appears in response. The **Certificate Authority** radio button at the top of the window is filled in.

   b. Fill out the fields within the *Distinguished Name* area of the Create Identity window as appropriate. (Click the **Help** button at the bottom of the window if you need more information about any of these fields.)

   c. Click the **OK** button at the bottom of the window.

   The Create New Credentials window appears in response.

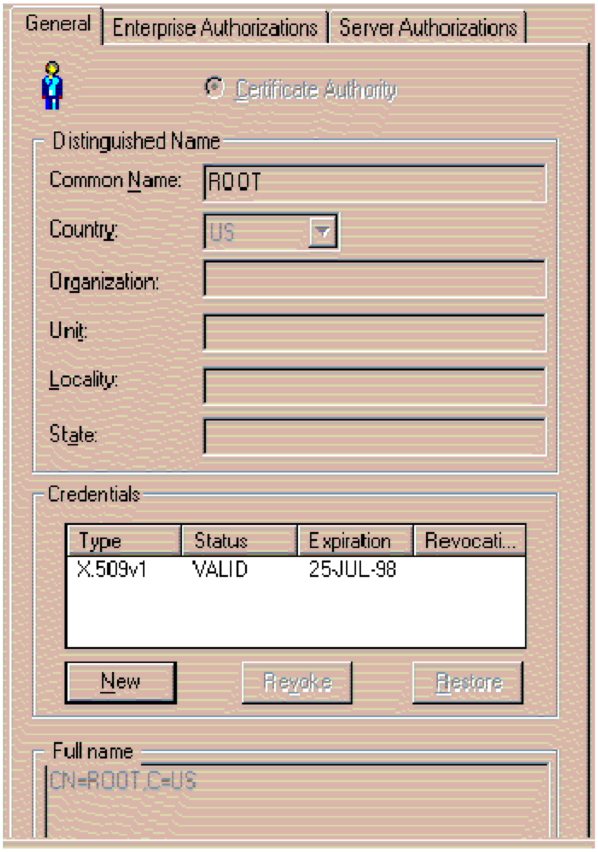
   d. Enter and/or change the values of the fields on the Create New Credentials window as appropriate. (Click the **Help** button at the bottom of the window if you need more information about any of these fields.)

e. Click the **Create** button at the bottom of the window.

The CA will appear in the tree structure on the Oracle Security Server Manager window within the **oss/Identities/Approved** folder.

Figure 3–2 shows the Identity and credentials information for a typical CA.

Figure 3–2 Identity Window for Root User



Common Name: ROOT

Country: US

Organization:

Unit:

Locality:

State:

Type	Status	Expiration	Revocati...
X.509v1	VALID	25-JUL-98	

CN=ROOT,C=US

2. To define a Server:

   a. Select *Create* from the **Server** pulldown on the Oracle Security Server Manager window.

   The Create Server window appears in response.

   b. Type the name of the new Server, in the **Server Name** field.

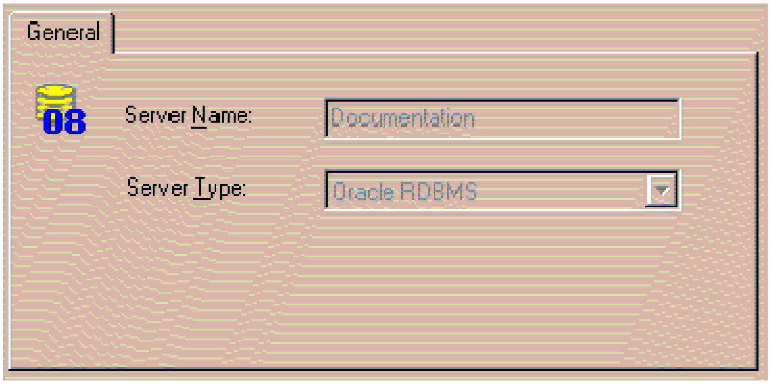
   > **Note:** This name must match the global name of the associated database.

   c. Click the **OK** button at the bottom of the window.

   The new Server will appear in the tree structure on the Oracle Security Server Manager window within the **oss/Authorizations/Server Authorizations** folder.

Figure 3–3 shows the information for a typical Server.

Figure 3–3 Create Server Window for Sample Server



Server Name: Documentation

Server Type: Oracle RDBMS

You can define as many Servers as you wish during this step.

3. To define a Server Authorization:

   a. Select *Create* from the **Server Authorization** pulldown on the Oracle Security Server Manager window.

   The Create Server Authorization window appears in response.

   b. Type the name of the new Server Authorization, in the **Role Name** field.

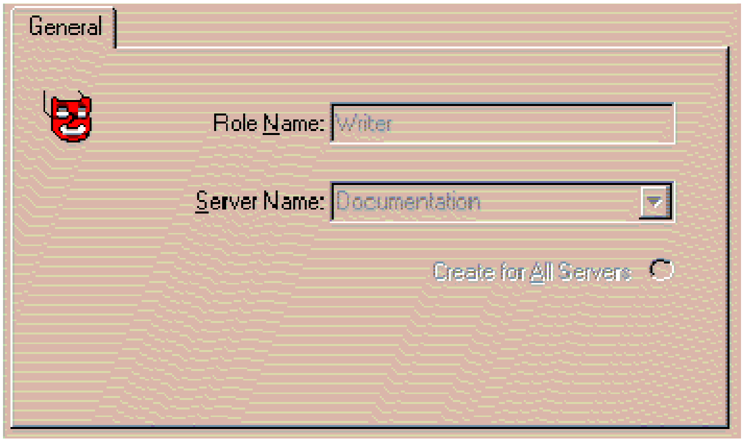
   c. If you wish to define the new Server Authorization for only one Server, select the name of that Server from the Server Name pulldown menu. If you wish to define the new Server Authorization for all of the Servers you have defined to the Oracle Security Server, click on the radio button next to **Create for All Servers**.

   d. Click the **OK** button at the bottom of the window.

   The new Server Authorization will appear in the tree structure on the Oracle Security Server Manager window within the **Roles** folder under the entry for each Server with which the new Server Authorization is associated. Each of these Server entities resides under the **oss/Authorizations/Server Authorizations** folder.

Figure 3–4 shows the information for a typical Server Authorization.

Figure 3–4 Server Authorization Window for Sample Server Authorization



Role Name: Writer

Server Name: Documentation

Create for All Servers ○

You can define as many Server Authorizations as you wish during this step.

4. To define an Enterprise Authorization:

   a. Select *Create* from the **Enterprise Authorization** pulldown on the Oracle Security Server Manager window.

   The Create Enterprise Authorization window appears in response.

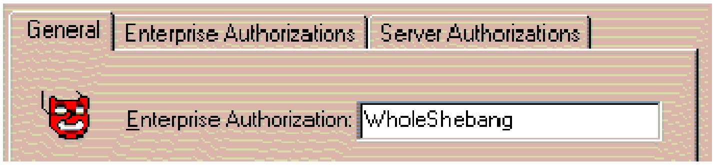
   b. Type the name of the new Enterprise Authorization, in the **Enterprise Authorization** field.

   c. Click the **OK** button at the bottom of the window.

   The new Enterprise Authorization will appear in the tree structure on the Oracle Security Server Manager window within the **oss/Authorizations/Enterprise Authorizations** folder.

Figure 3–5 shows the basic information for a typical Enterprise Authorization.

Figure 3–5 Enterprise Authorization Window for Sample Enterprise Authorization



Enterprise Authorization: WholeShebang

   d. In the tree structure, click the symbol for the new Enterprise Authorization.

   The Enterprise Authorization window appears in response.

   e. Click the *Server Authorizations* tab on the Enterprise Authorization window.

   The window associated with that tab appears in response.

   f. The Server Authorizations that you defined at Step 3 of this procedure are available for you to roll up into Enterprise Authorizations. To assign a Server Authorization to the Enterprise Authorization you are defining:

      * Select a Server from the **Name** pulldown menu.

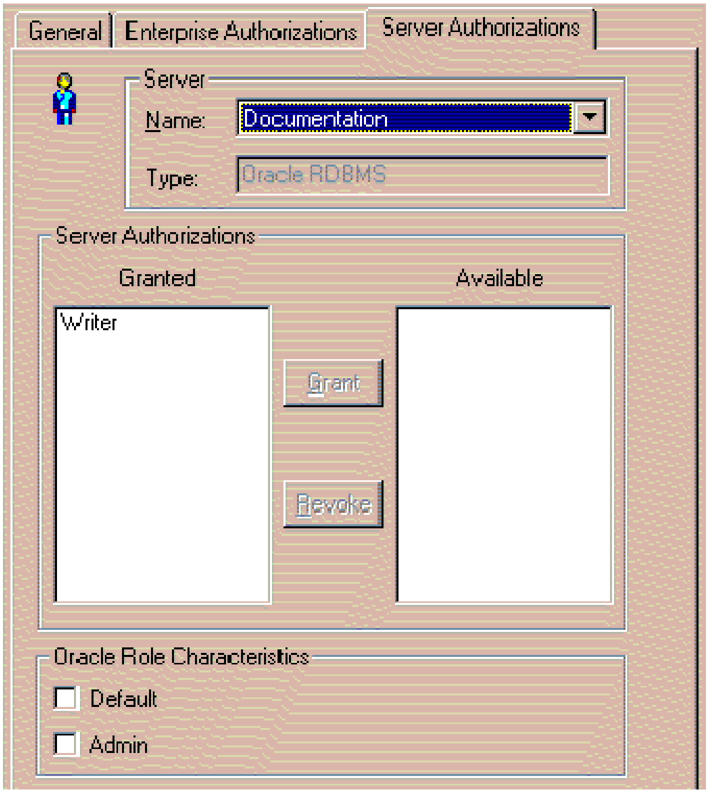
      * Click the name of a Server Authorization that appears in the *Available* column.

- Click the **Grant** button.

The Server Authorization has been moved from the *Available* column to the *Granted* column.

Figure 3–6 shows the Server Authorizations, associated with a particular Server, that have been assigned to a typical Enterprise Authorization.

Figure 3–6 Server Authorizations for Typical Enterprise Authorization



Name: Documentation

Type: Oracle RDBMS

Writer

□ Default

□ Admin

You can define as many Enterprise Authorizations as you wish during this step.

Taken together, Server Authorizations and Enterprise Authorizations are the Oracle Security Server's implementation of the global role concept introduced in Chapter 1, "Oracle Security Server Concepts." Specifically, if an Identity defined within the Oracle Security Server is authorized to perform a particular role on a particular server, then a user who has been "identified globally" on that server can acquire a role, of the same name, that has been "identified globally" on that server.

5. Define Identities for each of the users that have been defined as "identified globally" to one or more Oracle8 Servers. (See the section "Defining Global Users and Global Roles to Oracle8 Servers," which appears earlier in this chapter.)

   Oracle recommends that you, in your role as security administrator, validate each user's identity, using some form of strong identification (such as a driver's license or passport), before defining that user to the Oracle Security Server.

   To define an Identity for a user:

   a. Select *Create* from the **Identity** pulldown on the Oracle Security Server Manager window.

   The Create Identity window appears in response.

   b. Fill out the fields within the *Distinguished Name* area of the Create Identity window as appropriate. (Click the **Help** button at the bottom of the window if you need more information about any of these fields.)

   c. Click the **OK** button at the bottom of the window.

   The Create New Credentials window appears in response.

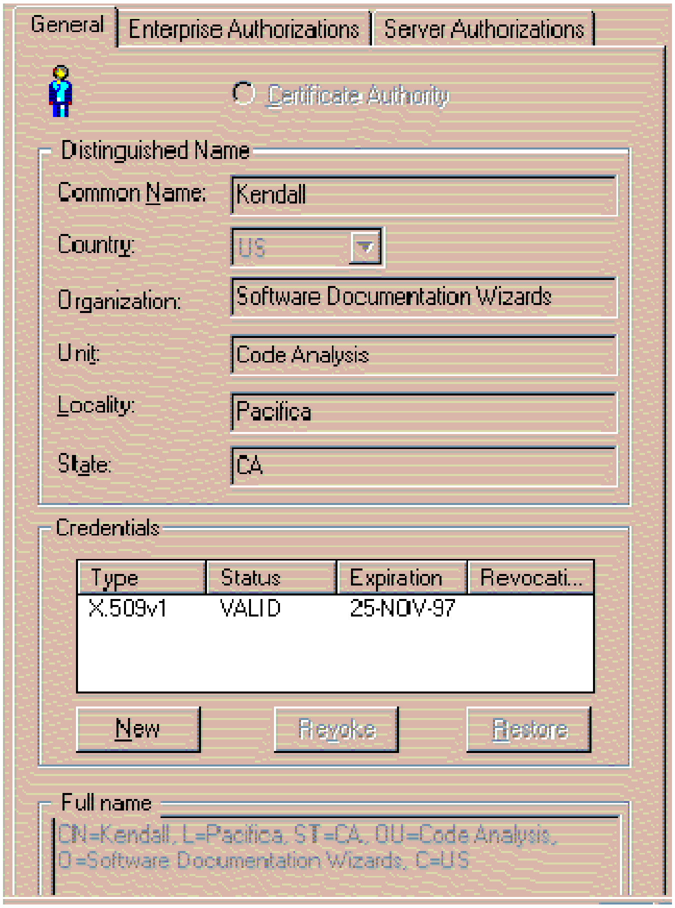
   d. Enter and / or change the values of the fields on the Create New Credentials window as appropriate. (Click the **Help** button at the bottom of the window if you need more information about any of these fields.)

   e. Click the **Create** button at the bottom of the window.

   The new Identity will appear in the tree structure on the Oracle Security Server Manager window within the **oss/Identities/Approved** folder.

Figure 3–7 shows the Identity and credentials information for a typical user Identity.

Figure 3–7 Identity Window for Sample User



Distinguished Name

Common Name:	Kendall
Country:	US
Organization:	Software Documentation Wizards
Unit:	Code Analysis
Locality:	Pacifica
State:	CA

Credentials

Type	Status	Expiration	Revocati...
X.509v1	VALID	25-NOV-97	

Full name

CN=Kendall, L=Pacifica, ST=CA, OU=Code Analysis, O=Software Documentation Wizards, C=US

f. Click the *Server Authorizations* tab.

The window associated with that tab appears in response.

g. The Server Authorizations that you defined at Step 3 of this procedure are available for you to assign to the new Identity. To assign a Server Authorization to the Identity you are defining:

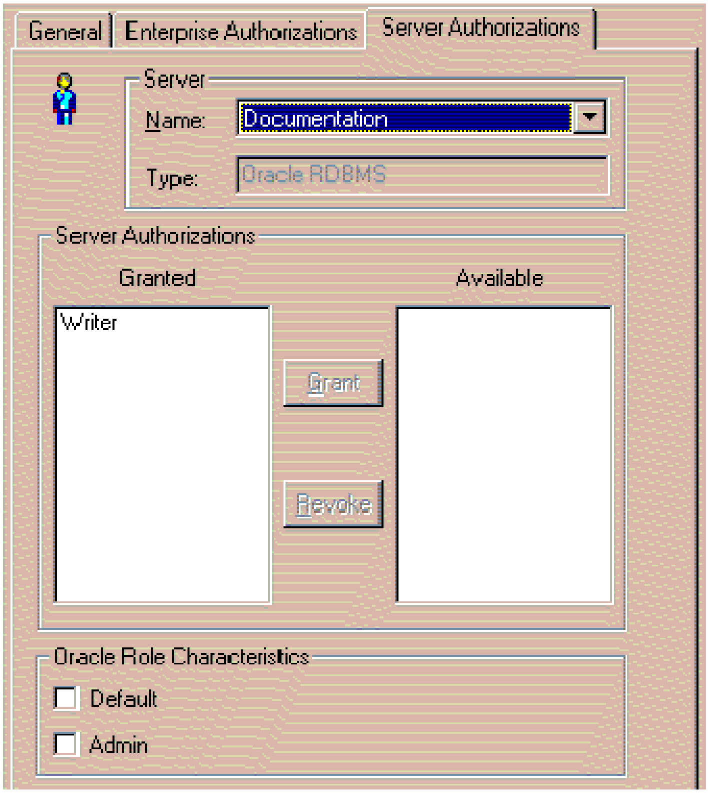
- Select a Server from the **Name** pulldown menu.

- * Click the name of a Server Authorization that appears in the *Available* column.
- * Click the **Grant** button.

The Server Authorization has been moved from the *Available* column to the *Granted* column.

Figure 3–8 shows the Server Authorizations, associated with a particular Server, that have been assigned to a typical Identity.

Figure 3–8 Server Authorizations for Typical Identity



General | Enterprise Authorizations | Server Authorizations

Server

Name: Documentation

Type: Oracle RDBMS

Server Authorizations

Granted: Writer

Available:

Oracle Role Characteristics

□ Default

□ Admin

h. Click the *Enterprise Authorizations* tab.

The window associated with that tab appears in response.

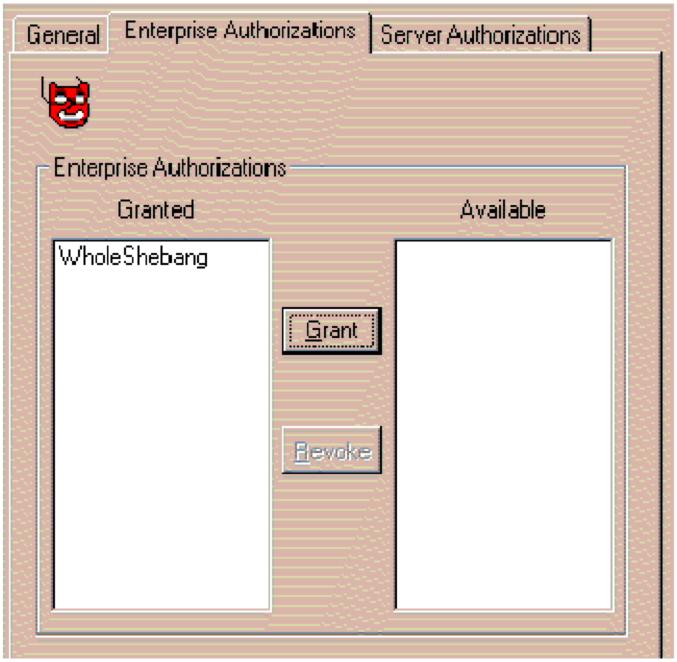
i. The Enterprise Authorizations that you defined at Step 4 of this procedure are available for you to assign to the new Identity. To assign an Enterprise Authorization to the Identity you are defining:

- * Click the name of an Enterprise Authorization that appears in the *Available* column.
- * Click the **Grant** button.

The Enterprise Authorization has been moved from the *Available* column to the *Granted* column.

Figure 3–9 shows the Enterprise Authorizations that have been assigned to a typical Identity.

Figure 3–9 Enterprise Authorizations for Typical Identity



You can define as many Identities as you wish during this step.

6. To approve, if applicable and desirable, credentials for an Identity generated a by a WebServer, which appear initially in the tree structure under the **oss/Identities/Requested** folder:

   a. Click the symbol for that Identity.

   The Approve Credentials window appears in response.

   b. Type the password for the CA, which you defined in Step 1 of this procedure, in the **Enter CA Password** field on the Approve Credentials window.

   c. Click the **Generate Identity** button near the bottom of the window.

   Once you have approved credentials for an externally defined Identity, that Identity should have moved, within the tree structure, from a folder under **oss/Identities/Requested** to one under **oss/Identifies/Approved**.

   You can approve credentials for as many of these Identities as you wish during this step.

7. Define Identities for the other principals, such as Oracle8 Servers and applications, that will be doing business with the Oracle Security Server.

Configuring Oracle Security Adapters on Clients and Servers

If you wish to enable the Oracle Security Adapter on a particular client or server, use Oracle Net8 Assistant or your command line to perform one or more of the following steps, as necessary, to modify the SQLNET.ORA file:

1. In order to be authenticated, each principal must possess a wallet. A **wallet** is a file that contains an X.509 certificate (see “Oracle Security Server Certificates” in Chapter 1) and a public/private key pair (see “Public/Private Key Pairs,“ also in Chapter 1). The private key is encrypted.

   The SQLNET.ORA file, at a given Net8 client or Oracle8 Server, contains the location of that entity’s wallet. A given client or server downloads its wallet from the Oracle Security Server Repository. The wallet is stored local to the client or server, which is responsible for protecting the wallet’s contents from unauthorized access. The client or server will use its password to decrypt the encrypted private key that is contained in the wallet.

   Define the location of the directory containing the given principal’s wallet, using the following template:

```
oss.source.my_wallet=
    (SOURCE =
       (METHOD=FILE)
       (METHOD_DATA=
          (DIRECTORY= directory_path)
       )
    )
```

where *directory_path* is the full name of the appropriate directory (for example, **/oracle_home/network/admin** on UNIX).

If you do not specify a value for oss.source.my_wallet, the Oracle Security Server uses the default value, which is a “well–known” directory under the principal’s home directory. On UNIX, for example, this directory is **$HOME/oracle/oss**.

2. Define a name for the Oracle Security Server Repository, the database that holds the wallets for all principals. Define a name that points to the entry, in the TNSNAMES.ORA file, that contains the TNS address for this server, using the following template:

```
oss.source.location =
    (SOURCE=
       (METHOD=ORACLE)
       (METHOD=DATA=
          (SQLNET_ADDRESS=<service_name>)
       )
    )
```

where *service_name* is the name defined within the TNSNAMES.ORA file (**oracle_repository**, for instance) for the Oracle Security Server Repository.

If you do not specify a value for oss.source. location, the Oracle Security Server uses the default value, **OSS**, which refers to the Oracle Security Server Repository.

Note: There **must** be a pointer to the Oracle Security Server Repository from either the TNSNAMES.ORA file or a Names Server.

Installing Wallets at Clients and Servers

You use the *osslogin* tool to download a wallet, or to generate a clear private key by decrypting an encrypted private key contained within a wallet.

The syntax of the *osslogin* command is as follows:

```
osslogin [-d] ['<X.509 Name>']
```

The X.509 name for a client or server contains all of the standard and optional values that form the name within that entity's Identity: C=*country*, O=*organization*, OU=*organization_unit*, ST=*state*, L=*locality*, CN=*user*. For a global user, these values must appear in exactly the same order as they did when that user was "identified globally" to the database. (See the section "Defining Global Users and Global Roles to Oracle8 Servers," which appears earlier in this chapter.) The single quotes are required; *osslogin* will generate an error message if you do not provide them.

What the tool does depends on what form of the command you issue and what information exists in what location(s).

Downloading a Wallet

If you issue the command

```
osslogin -d '<X.509 Name>'
```

the tool will retrieve the given client or server wallet from the Oracle Security Server Repository, using the specified X.509 name, and then download the wallet to the location specified by the value of the `oss.source.my_wallet` parameter within the SQLNET.ORA file (see the section "Configuring Oracle Security Adapters on Clients and Servers," which appears earlier in this chapter).

> **Note**: The certificate and encrypted private key contained within the wallet are protected only by the access control mechanisms provided by the client or server operating system. (The clear private key will not be generated in association with this form of the *osslogin* command.)

> **CAUTION:** You should delete your wallet when you are finished using it for a particular communications session. This will help protect its contents from unauthorized access.

Generating a Decrypted (Clear) Private Key (Name Specified)

If you issue the command

```
osslogin '<X.509 Name>'
```

the tool will look for the given client or server wallet at the location specified by the value of `oss.source.my_wallet`.

If the wallet is not at that location, the tool will retrieve it from the Oracle Security Server Repository, using the specified X.509 name, and then download the wallet to the specified location.

Note: The certificate and encrypted private key contained within the wallet are protected only by the access control mechanisms provided by the client or server operating system.

Once it has been established that the wallet is stored locally at the client or server, the tool will prompt you to enter the client or server password. The tool will generate the clear private key by using the password to decrypt the encrypted private key, and will then store the clear private key local to the client or server, at the location specified within the `oss.source.my_wallet` parameter.

WARNING: You must protect the clear private key carefully. The clear private key authenticates the client or server on the network. If unauthorized users were allowed to access the clear key file, they could masquerade as the client or server on the network and obtain the entity's privileged information.

Note that you can add **-f** to this form of the command:

```
osslogin -f '<X.509 Name>'
```

to force the tool to go directly to the Oracle Security Server Repository in search of the client or server wallet. If you choose this option, the tool will prompt you for the X.509 name.

CAUTION: You should delete your wallet when you are finished using it for a particular communications session. This will help protect its contents from unauthorized access.

Generating a Decrypted (Clear) Private Key (Name Not Specified)

If you issue the command

```
osslogin
```

the tool will look for the given client or server wallet at the location specified by the value of `oss.source.my_wallet`.

If the wallet is not at that location, the tool will prompt you to enter the elements of the X.509 name of the client or server. The tool will then use that name to retrieve the client or server wallet from the Oracle Security Server Repository, and then download the wallet to the specified location.

Note: The certificate and encrypted private key contained within the wallet are protected only by the access control mechanisms provided by the client or server operating system.

Once the wallet is stored locally at the client or server, the tool will prompt you to enter the client or server password. The tool will generate the clear private key by using the password to decrypt the encrypted private key, and will then store the clear private key local to the client or server, at the location specified within the `oss.source.my_wallet` parameter.

WARNING: You must protect the clear private key carefully. The clear private key authenticates the client or server on the network. If unauthorized users were allowed to access the clear key file, they could masquerade as the client or server on the network and obtain the entity’s privileged information.

CAUTION: You should delete your wallet when you are finished using it for a particular communications session. This will help protect its contents from unauthorized access.

Removing the Oracle Security Server Repository

WARNING: You should consider deleting the Oracle Security Server Repository *only* if you have migrated the data from the repository to another database or if you intend to discontinue use of the Oracle Security Server.

A DBA should perform the following steps to remove the Oracle Security Server Repository from its Oracle database:

1. Launch the **Remove Security Server** program from the **Oracle Security Server** program group on the desktop.

The Database Login Information Window appears in response.

2. Use the Database Login Information window to log into the database that contains the Oracle Security Server Repository.

   a. Type *system* in the **Username** field.

   b. Type the password for the Oracle Security Server administrator, in the **Password** field.

   c. Type the name of the database on which the Oracle Security Server Repository resides, in the **Service** field.

   d. Click the **OK** button.

A confirmation window appears in response. This window will ask you to confirm that you want to remove the Oracle Security Server Repository from the specified database.

3. Click the **OK** button on the confirmation window.

4

Using the Oracle Security Server Manager

This chapter details how a security administrator (SA) uses the Oracle Security Server Manager to define elements to the Oracle Security Server. The following topics are discussed:

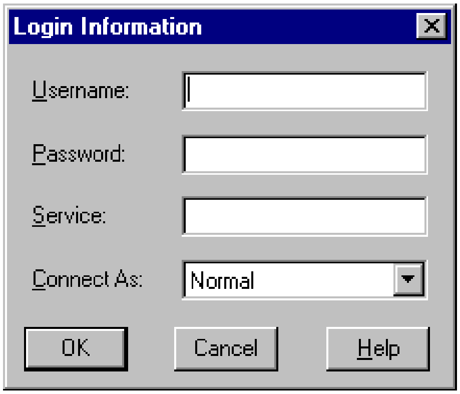
- Getting Started
- Identities
- Servers
- Server Authorizations
- Enterprise Authorizations

Getting Started

The first step is to launch the **Oracle Security Server Manager** program from the **Oracle Security Server** program group on your desktop.

The Login Information window (Figure 4–1) appears in response.

Figure 4–1 Login Information Window



Login Information

Username:

Password:

Service:

Connect As: Normal

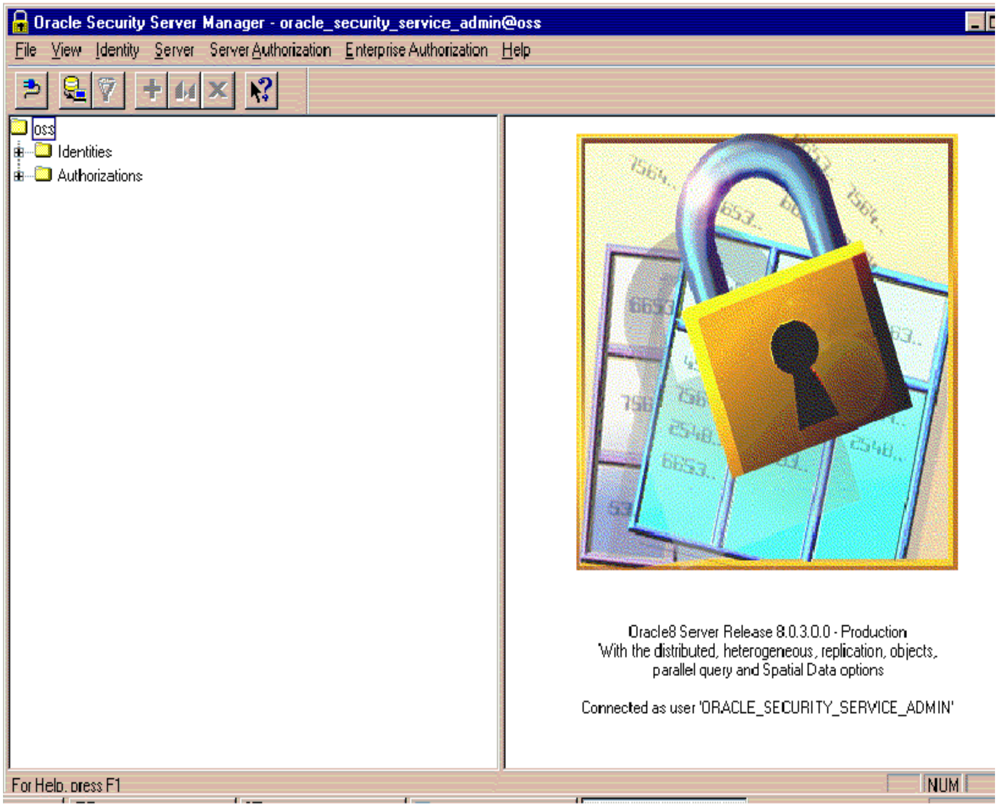
Login Information Window

Perform the following steps to log in to the Oracle Security Server Manager:

1. Type *oracle_security_service_admin* in the **Username** field. (See the section “Installing the Oracle Security Server Repository” within Chapter 3 for more information about this username.)
2. Type the security administrator password, in the **Password** field.
3. Type the name of the database on which the Oracle Security Server Repository resides, in the **Service** field.
4. Click the **OK** button.

The Oracle Security Server Manager window (Figure 4–2) appears in response.

Figure 4–2 Oracle Security Server Manager Window



Oracle Security Server Manager - oracle_security_service_admin@oss
File View Identity Server Server Authorization Enterprise Authorization Help
oss
Identities
Authorizations
For Help, press F1
NUM

Oracle8 Server Release 8.0.3.0.0 - Production
With the distributed, heterogeneous, replication, objects,
parallel query and Spatial Data options

Connected as user 'ORACLE_SECURITY_SERVICE_ADMIN'

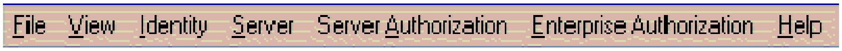
Oracle Security Server Manager Window

Three components of this window are discussed in the following subsections.

Menu Bar

A menu bar (Figure 4–3) appears at the top of the Oracle Security Server Manager window.

Figure 4–3 Menu Bar



Several of the pulldowns represented on this bar are discussed in separate sections later in this chapter. (The **File** and **View** pulldowns are not relevant to this manual.)

Of particular interest here are **Identity**, **Server, Server Authorization**, and **Enterprise Authorization**.

All four of these pulldowns contain the following selections:

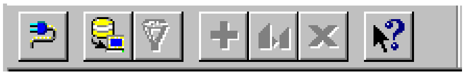
- *Create*, which enables you to define an item to the Oracle Security Server.
- *Create Like*, which enables you to define a new item using an existing item as a base.
- *Delete*, which enables you to delete an item.

Note that none of these pulldowns contains any kind of “modify” option. Within the Oracle Security Server Manager, you must delete an item and define a new one if you need to change the information for that item.

Tool Bar

A tool bar (Figure 4–4) appears in the upper lefthand area of the Oracle Security Server Manager window.

Figure 4–4 Tool Bar



As you can see, not all of the buttons are enabled. The Oracle Security Server Manager only enables a button when it is appropriate for you to be performing the corresponding function.

The two leftmost buttons are not of concern to you. The functions associated with the other five buttons, reading from left to right, starting with the funnel), are:

- *Filter Folder*, which enables you to change the order and / or quantity of items that you see within the tree structure (see “Work Area“ below).
- *Create*, which matches up with the Create option described in the previous subsection, “Menu Bar.”

- *Create Like*, which matches up with the Create Like option described previously.
- *Delete*, which matches up with the Delete option described in “Menu Bar.“
- *Help*, which offers context–sensitive help.

Work Area

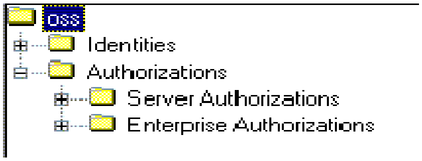
The remaining area of the Oracle Security Server Manager window is divided into two parts. The left part contains a tree structure; the right part is where you will be doing the data entry involved in defining entities to the system.

Note that the figures within this section make use of an Oracle Security Server Repository that has been configured with sample data.

You can perform three types of operations on the tree structure:

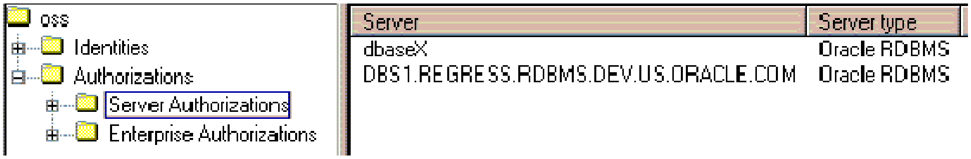
- **Look at what has been defined**.
  - As in other Windows systems, you can click the plus sign (+) that appears next to an item in the tree to uncover the entities underneath it in the structure. Figure 4–5 shows what appears within the *Authorizations* folder.

Figure 4–5 Authorizations



  - You can see all of the information that exists in the repository for the entities within a given folder, by clicking that folder. Figure 4–6 shows what might appear within the *Server Authorizations* folder.

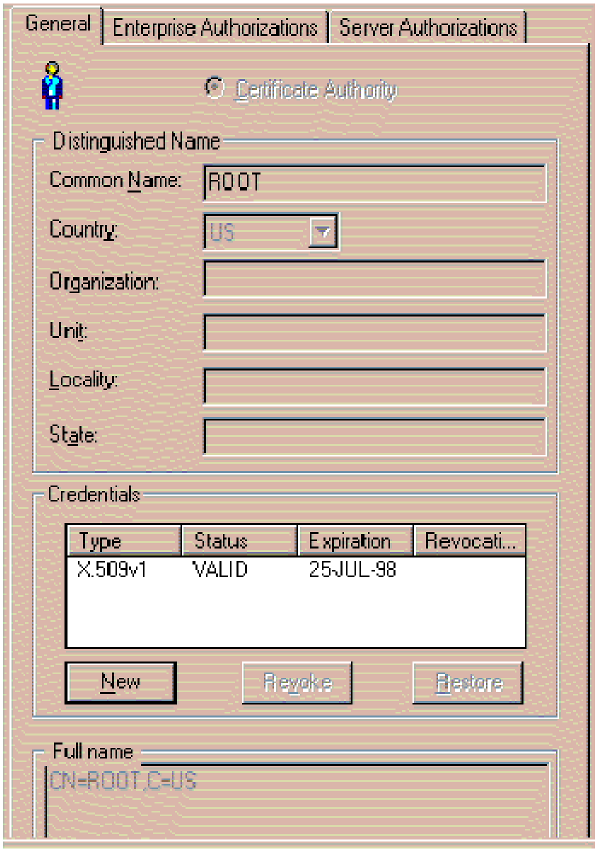
Figure 4–6 Server Authorizations



Server	Server type
dbaseX	Oracle RDBMS
DBS1.REGRESS.RDBMS.DEV.US.ORACLE.COM	Oracle RDBMS

- **Open a data entry window for viewing.** If you click on the symbol for an Identity, Server, Server Authorization, or Enterprise Authorization, the system brings up the data entry window associated with that item. This window contains the information you entered. The window is in display–only mode—remember that there is no “modify” operation. Figure 4–7 shows what might appear for the Identity that has been defined for “root” (the *oracle_security_service_admin* user).

Figure 4–7 Identity Window for Root User



○ Certificate Authority

Common Name:	ROOT
Country:	US
Organization:	
Unit:	
Locality:	
State:	

Type	Status	Expiration	Revocati...
X.509v1	VALID	25-JUL-98	

Full name: CN=ROOT,C=US

- **Perform a “drag and drop.”** You can perform certain tasks using the tree structure, as opposed to the menu bar, by dragging an item (for instance, a Server Authorization) and dropping it on another item (say, an Identity). See the sections that follow for specifics about particular tasks that you can perform via “drag and drop.”

Identities

An **Identity** is a representation of any entity that does business with the Oracle Security Server. This includes users, programs or other systems, and the Certification Authority (CA). (See Chapter 1, Oracle Security Server Concepts, for more information about the CA.)

An Identity has two basic elements: a distinguished name (DN) (see “Distinguished Names (DNs)” within Chapter 1), and **credentials**, which are based on an X.509 certificate (see “Oracle Security Server Certificates,” also within Chapter 1).

The tasks that you can perform with regard to Identities are described in the following subsections.

Creating an Identity

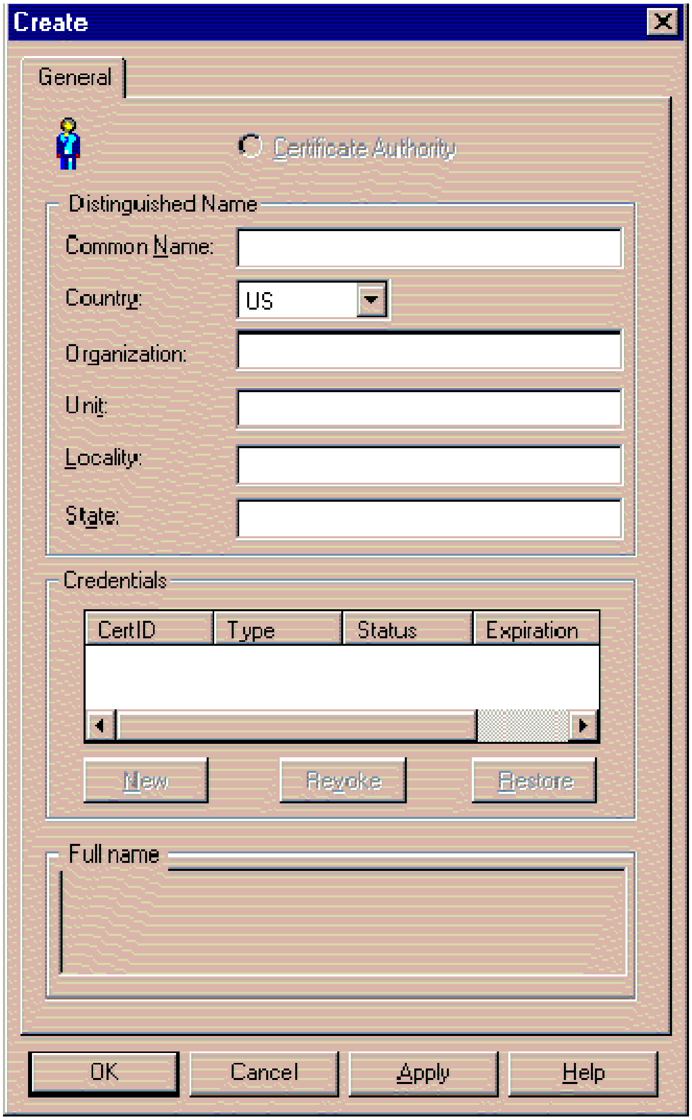
Create

If you wish to define a new Identity to the Oracle Security Server, select *Create* from the **Identity** pulldown on the Oracle Security Server Manager window.

The window shown in Figure 4–8 appears in response.

Refer to the on-line help for information about how to use this window. To bring up help for this topic, click the **Help** button on the window.

__Figure 4–8 Create Identity Window__



Country: US

Create Like

If you wish to define a new Identity based on an existing Identity:

- Select the existing Identity within the tree structure on the Oracle Security Server Manager window.
- Select *Create Like* from the **Identity** pulldown.

The window shown in Figure 4–9 appears in response.

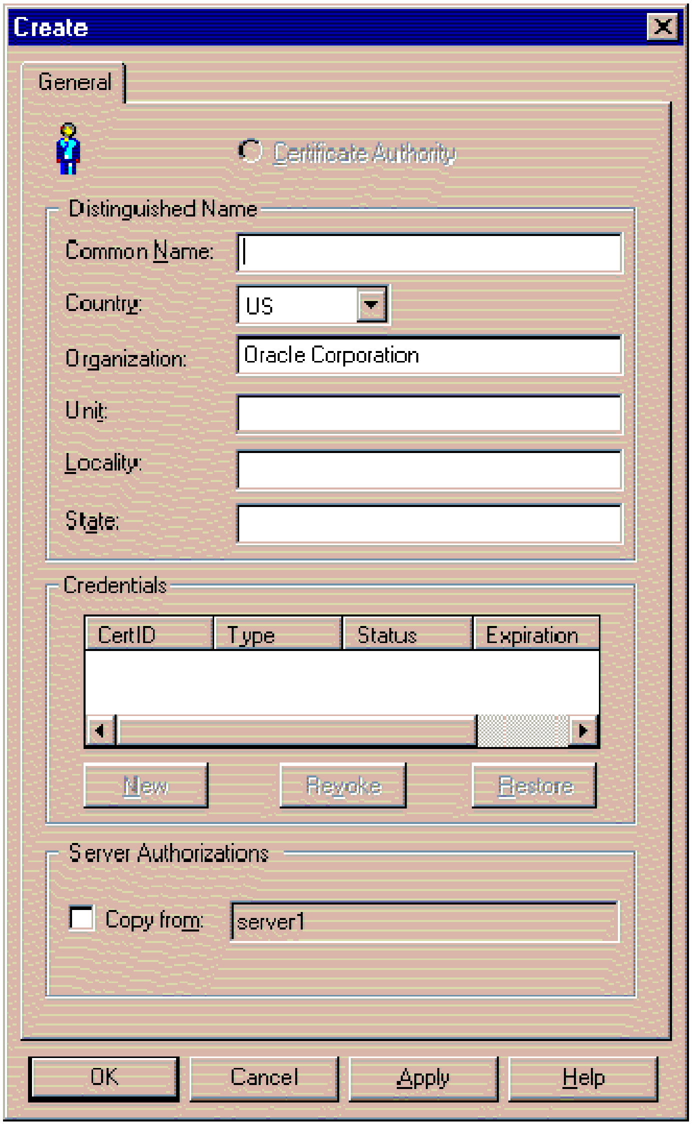
The window contains all of the information that you defined for the existing Identity except for its **Common Name**. Add and/or change field values as desired. Note that you can also tell the system to assign, to the new Identity, the Server Authorizations and Enterprise Authorizations that have been granted to the existing Identity, by clicking on the box next to **Copy from** within the *Attributes* area of the window.

Refer to the on-line help, which is the same as for the Create function, as necessary.

Creating Credentials for a New Identity

Once you have clicked the **OK** button on the Create Identity Window, the window shown in Figure 4–10 appears in response.

Figure 4–9 Create Identity Like Window



Distinguished Name

Common Name:

Country: US

Organization: Oracle Corporation

Unit:

Locality:

State:

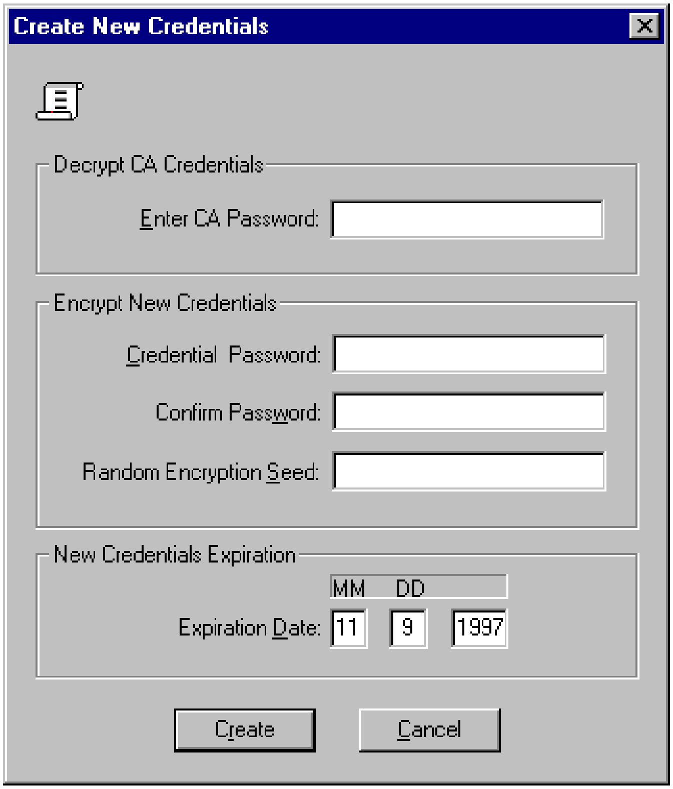
Credentials

CertID	Type	Status	Expiration

Server Authorizations

□ Copy from: server1

Figure 4–10 Create New Credentials Window



Expiration Date: 11 9 1997

Refer to the on-line help for information about how to use this window. To bring up help for this topic, click the **Help** button on the window.

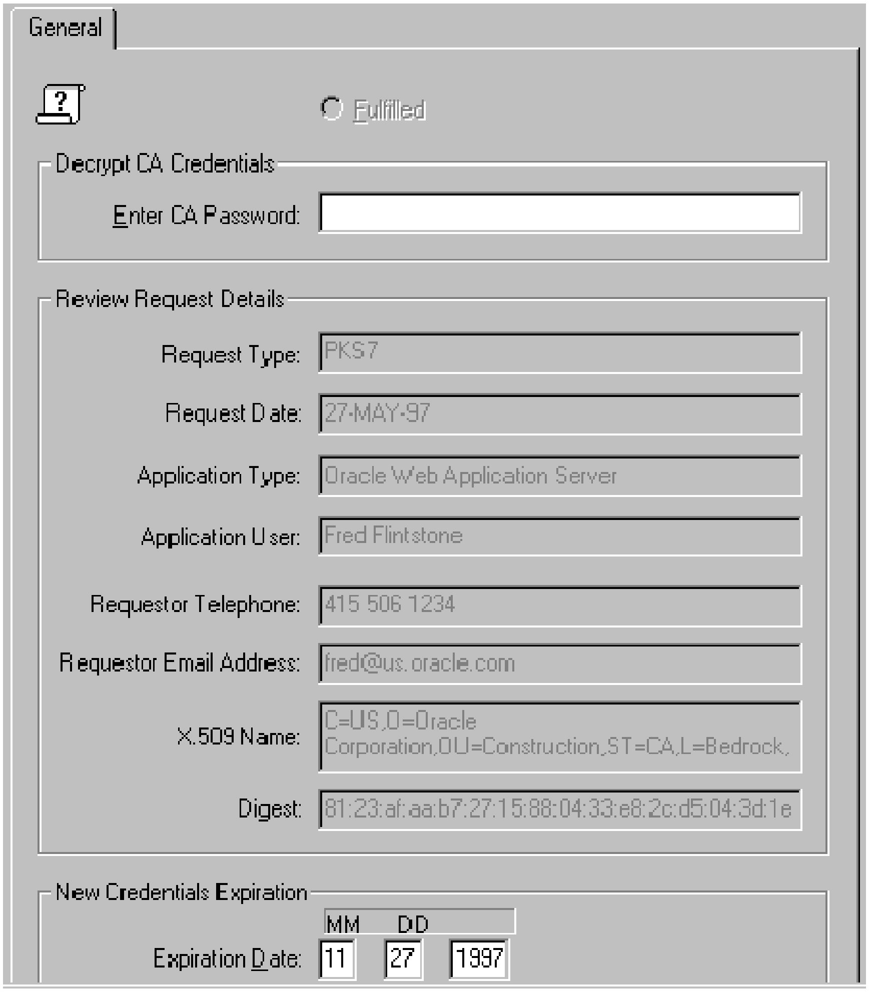
Once you have finished defining a new Identity, it should be present in the tree structure, within the appropriate folder under **oss/Identities/Approved**.

Approving Credentials for an Externally Defined Identity

If you wish to approve credentials for an entity that has been defined outside of the Oracle Security Server (for instance, by an Oracle WebServer), click on that Identity within the **oss/Identities/Requested** folder of the tree structure.

The window shown in Figure 4–11 appears in response.

Figure 4–11 Approve Credentials Window



○ Fulfilled

Decrypt CA Credentials

Enter CA Password:

Review Request Details

Request Type: PKS7

Request Date: 27-MAY-97

Application Type: Oracle Web Application Server

Application User: Fred Flintstone

Requestor Telephone: 415 506 1234

Requestor Email Address: fred@us.oracle.com

X.509 Name: C=US,O=Oracle Corporation,OU=Construction,ST=CA,L=Bedrock,

Digest: 81:23:af:aa:b7:27:15:88:04:33:e8:2c:d5:04:3d:1e

New Credentials Expiration

Expiration Date: MM 11 DD 27 1997

Note that the data within Figure 4–11 is sample data; the data that appears within this window on your system will vary.

Refer to the on-line help for information about how to use this window. To bring up help for this topic, click the **Help** button on the window.

Once you have approved credentials for an externally defined Identity, that Identity should have moved, within the tree structure, from a folder under **oss/Identities/Requested** to one under **oss/Identifies/Approved**.

Revoking Credentials

If you wish to revoke the credentials for a particular Identity (because, for instance, the Identity is associated with a user who no longer works at your company), click on the symbol for that Identity within the tree structure.

The window shown in Figure 4–8 appears in response.

Click on the line that appears within the Credentials area, and then click the **Revoke** button.

Restoring Credentials

If you wish to restore the credentials for a particular Identity, click on the symbol for that Identity within the tree structure.

The window shown in Figure 4–8 appears in response.

Click on the line that appears within the Credentials area, and then click the **Restore** button.

Deleting an Identity

If you wish to delete an Identity:

- Select that Identity within the tree structure on the Oracle Security Server Manager window.
- Select *Drop* from the **Identity** pulldown.

The window that appears in response asks if you are sure that you want to remove the specified Identity. Click **Yes** if you are sure, or **No** to return to the main window without performing the delete.

If you decide to go ahead with the deletion, the given Identity will no longer appear within the tree structure.

> **WARNING:** If you delete the CA's Identity, you will have to re–establish all of the other Identities that you have defined to the Oracle Security Server, because they contain the CA's digital signature. (As you can see on Figure 4–7, the **Certificate Authority** radio button is filled in on the Identity window associated with the CA.) The *only* reason you should *ever* consider doing this is if the CA's private key is somehow compromised. The Oracle Security Server Manager will ask you if you are *absolutely* sure before proceeding.

Servers

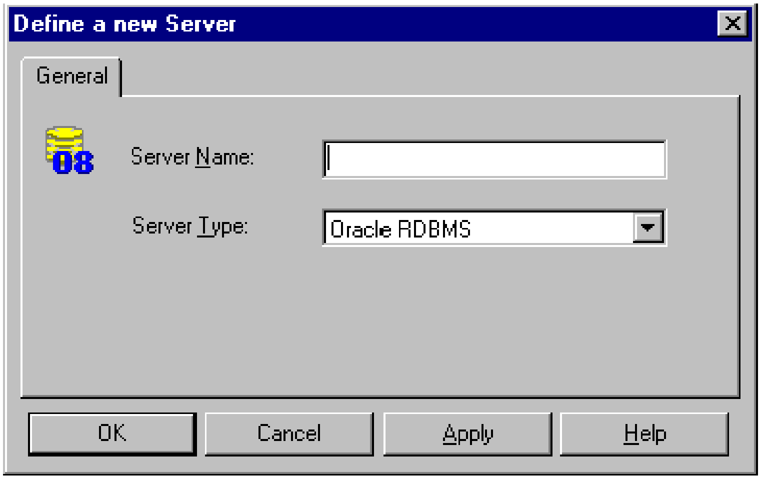
A **server**, in this context, is simply a representation of an Oracle8 Server.

Creating a Server

If you wish to define a new Server to the Oracle Security Server, select *Create* from the **Server** pulldown on the Oracle Security Server Manager window.

The window shown in Figure 4–12 appears in response.

Figure 4–12 Create Server Window



Define a new Server

Server Name:

Server Type: Oracle RDBMS

Refer to the on-line help for information about how to use this window. To bring up help for this topic, click the **Help** button on the window.

Note: The name of a Server must match the global name of the associated database.

Once you have finished defining a new Server, it should be present in the tree structure, within the **oss/Authorizations/Server Authorizations** folder.

Deleting a Server

If you wish to delete a server:

- Select that Server within the tree structure on the Oracle Security Server Manager window.
- Select *Delete* from the **Server** pulldown.

The window that appears in response asks if you are sure that you want to remove the specified Server. Click **Yes** if you are sure, or **No** to return to the main window without performing the delete.

If you decide to go ahead with the deletion, the given Server will no longer appear within the tree structure.

Server Authorizations

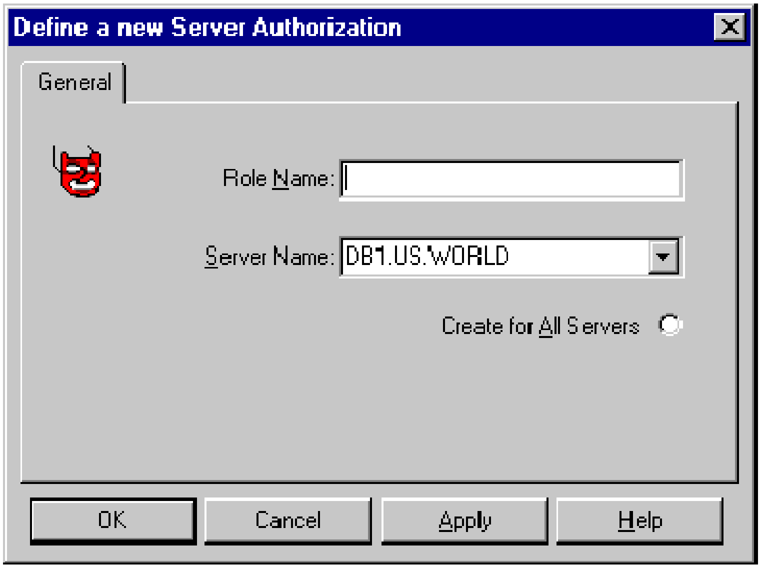
A s**erver authorization** is a representation of a role that has been “identified globally” at an Oracle8 Server.

Defining a Server Authorization

If you wish to define a new Server Authorization to the Oracle Security Server, select *Create* from the **Server Authorization** pulldown on the Oracle Security Server Manager window.

The window shown in Figure 4–13 appears in response.

Figure 4–13 Create Server Authorization Window



Define a new Server Authorization

Role Name:

Server Name: DB1.US.WORLD

Create for All Servers ○

Refer to the on-line help for information about how to use this window. To bring up help for this topic, click the **Help** button on the window.

Once you have finished defining a new Server Authorization, it should be present in the tree structure, within the **Roles** folder under the entry for each of the Servers with which the new Server Authorization is associated. These Server entities reside under the **oss/Authorizations/Server Authorizations** folder.

Deleting a Server Authorization

If you wish to delete a Server Authorization:

- Select that Server Authorization within the tree structure on the Oracle Security Server Manager window.
- Select *Delete* from the **Server Authorization** pulldown.

The window that appears in response asks if you are sure that you want to remove the specified Server Authorization. Click **Yes** if you are sure, or **No** to return to the main window without performing the delete.

If you decide to go ahead with the deletion, the given Server Authorization will no longer appear within the tree structure.

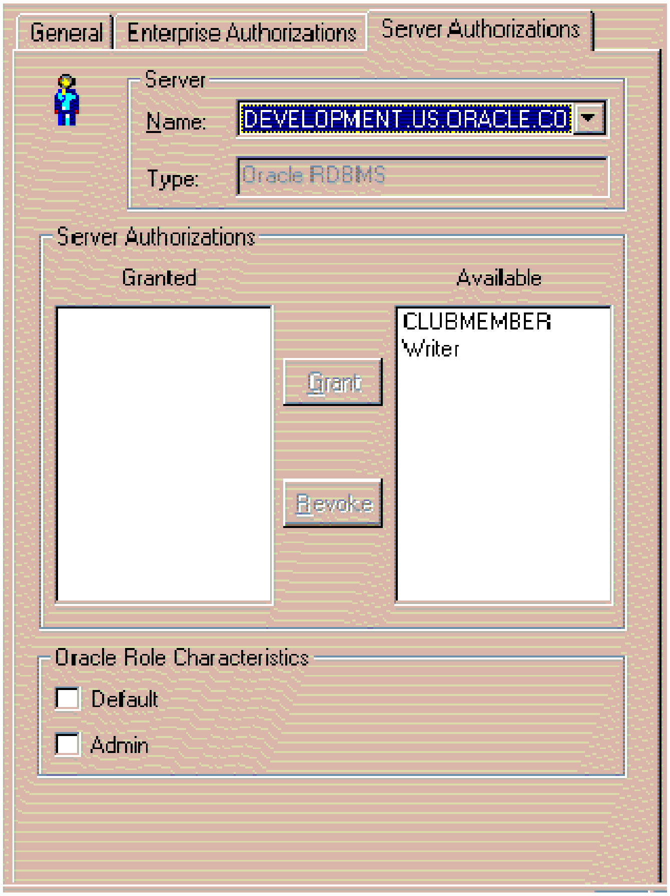
Granting and Revoking Server Authorizations

If you wish to grant and/or revoke Server Authorizations in association with a particular Identity:

- Select that Identity within the tree structure on the Oracle Security Server Manager window.
- Click the *Server Authorizations* tab on the window that appears.

The window shown in Figure 4–14 appears in response.

Figure 4–14 Server Authorizations Tab on Identity Window



Name: DEVELOPMENT.US.ORACLE.CO

Type: Oracle RDBMS

Available:
CLUBMEMBER
Writer

□ Default

□ Admin

Refer to the on-line help for more information about Server Authorization granting and revocation. To bring up help for this topic, click the **Help** button on the window.

You can also assign a Server Authorization to an Identity using “drag and drop,” by dragging the symbol for the given Server Authorization to the symbol for the Identity to which you wish you assign the authorization.

Enterprise Authorizations

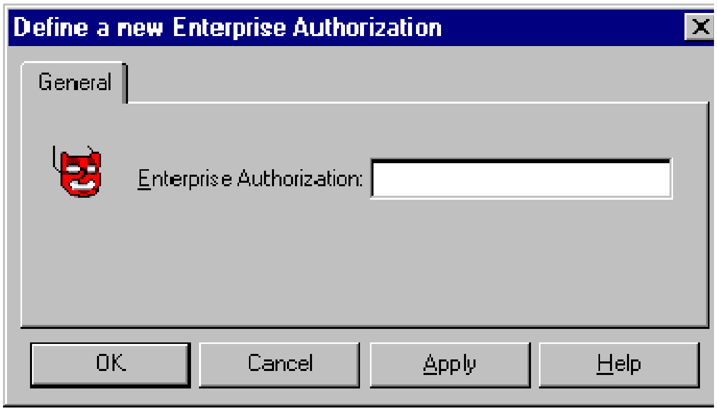
An **enterprise authorization** is a role that a global user can perform across multiple Oracle8 databases.

Defining an Enterprise Authorization

If you wish to define a new Enterprise Authorization to the Oracle Security Server, select *Create* from the **Enterprise Authorization** pulldown on the Oracle Security Server Manager window.

The window shown in Figure 4–15 appears in response.

Figure 4–15 Create Enterprise Authorization Window


Define a new Enterprise Authorization
General
Enterprise Authorization:
OK
Cancel
Apply
Help

Refer to the on-line help for information about how to use this window. To bring up help for this topic, click the **Help** button on the window.

Once you have finished defining a new Enterprise Authorization, it should be present in the tree structure, within the **oss/Authorizations/Enterprise Authorization** folder.

Deleting an Enterprise Authorization

If you wish to delete an Enterprise Authorization:

- Select that Enterprise Authorization within the tree structure on the Oracle Security Server Manager window.
- Select *Delete* from the **Enterprise Authorization** pulldown.

The window that appears in response asks if you are sure that you want to remove the specified Enterprise Authorization. Click **Yes** if you are sure, or **No** to return to the main window without performing the delete.

If you decide to go ahead with the deletion, the given Enterprise Authorization will no longer appear within the tree structure.

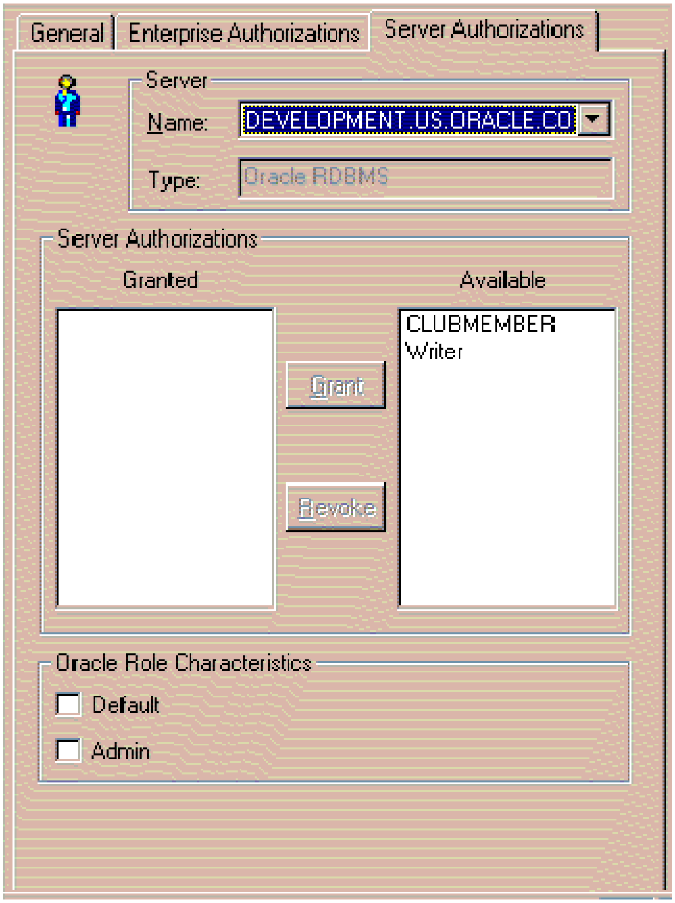
Adding and Deleting Server Authorizations for an Enterprise Authorization

If you wish to add Server Authorizations to, and/or delete Server Authorizations from, a particular Enterprise Authorization:

- Select that Enterprise Authorization within the tree structure on the Oracle Security Server Manager window.
- Click the *Server Authorizations* tab on the window that appears.

The window shown in Figure 4–16 appears in response.

Figure 4–16   Server Authorizations Tab on Enterprise Authorization Window



Name: DEVELOPMENT.US.ORACLE.CO

Type: Oracle RDBMS

Server Authorizations

Available:
CLUBMEMBER
Writer

□ Default

□ Admin

Refer to the on-line help for more information about adding and deleting Server Authorizations. To bring up help for this topic, click the **Help** button on the window.

You can also assign a Server Authorization to an Enterprise Authorization using “drag and drop,” by dragging the symbol for the Server Authorization to the symbol for the Enterprise Authorization to which you wish you assign the authorization.

Nesting Enterprise Authorizations

If you wish to nest one or more Enterprise Authorizations within a given Enterprise Authorization:

- Select that Enterprise Authorization within the tree structure on the Oracle Security Server Manager window.
- Click the *Enterprise Authorizations* tab on the window that appears.

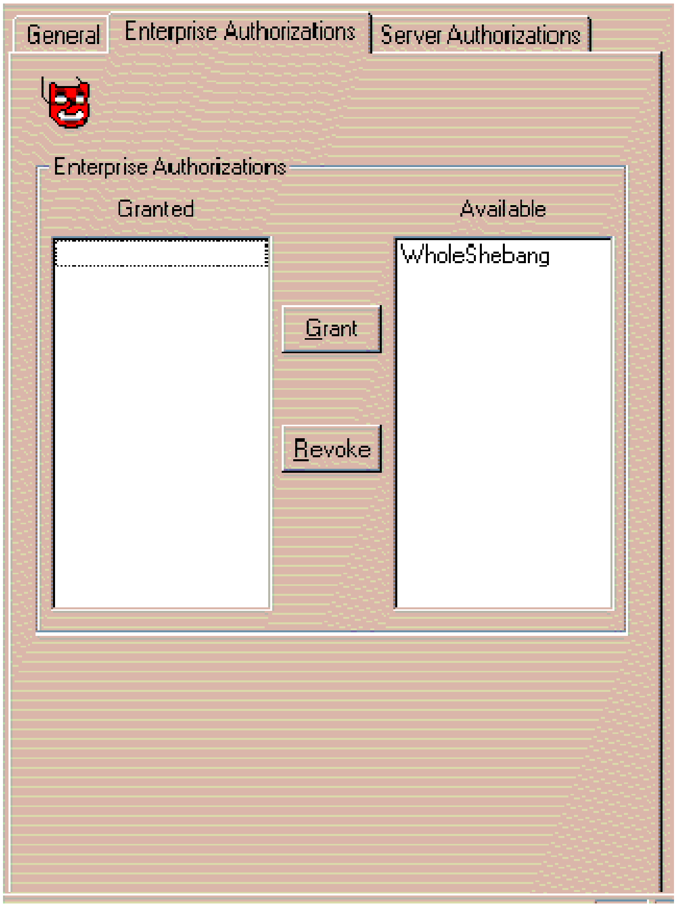
The window shown in Figure 4–17 appears in response.

Refer to the on-line help for more information about nesting Enterprise Authorizations. To bring up help for this topic, click the **Help** button on the window.

Once you have nested an Enterprise Authorization within another Enterprise Authorization, the system shows that Enterprise Authorization within the **Enterprise Authorizations** folder under the given Enterprise Authorization within the tree structure. A deletion results in an Enterprise Authorization disappearing from that folder.

CAUTION: Be careful not to nest Enterprise Authorizations within each other. In other words, if you nest Enterprise Authorization **X** within Enterprise Authorization **Y**, do not later nest Y within X. The results are unpredictable.

Figure 4–17 Enterprise Authorizations Tab on Enterprise Authorization Window



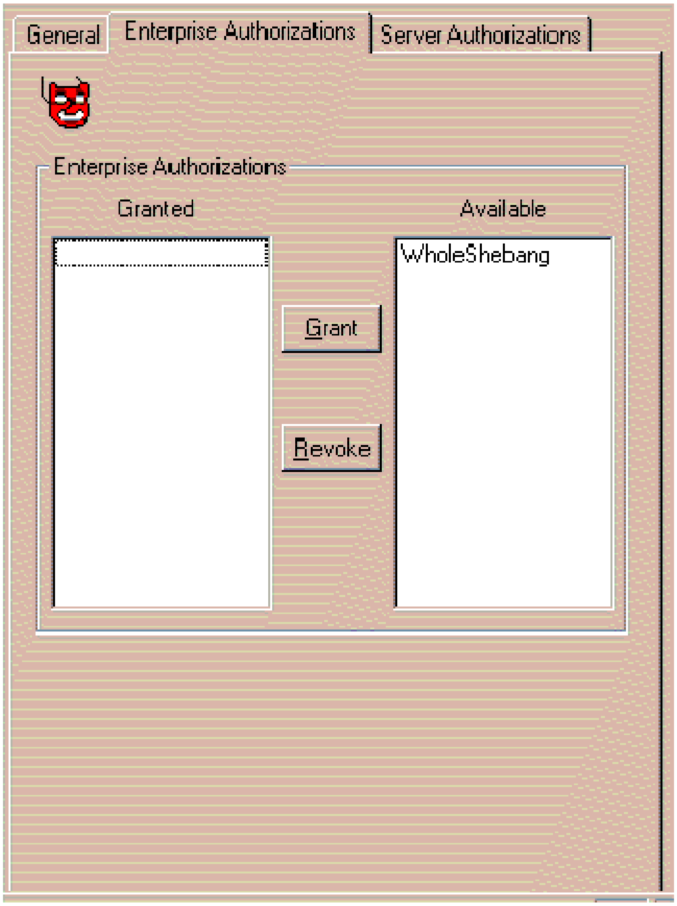
Granting and Revoking an Enterprise Authorization

If you wish to grant and / or revoke a particular Enterprise Authorization in association with a particular Identity:

- Select that Identity within the tree structure on the Oracle Security Server Manager window.
- Click the *Enterprise Authorizations* tab on the window that appears.

The window shown in Figure 4–18 appears in response.

Figure 4–18 Enterprise Authorizations Tab on Identity Window



Refer to the on-line help for more information about Enterprise Authorization granting and revocation. To bring up help for this topic, click the **Help** button on the window.

Once you have granted an Enterprise Authorization to an Identity, the system shows that Enterprise Authorization within the **Enterprise Authorizations** folder under the given Identity within the tree structure. A deletion results in an Enterprise Authorization disappearing from that folder.

You can also assign an Enterprise Authorization to an Identity using “drag and drop,” by dragging the symbol for the Enterprise Authorization to the symbol for the Identity to which you wish you assign the authorization.

Glossary

Asymmetric Cryptography

See *Public–Key Cryptography*.

Authentication

The process of proving the identity of a principal.There are three basic ways that you can be authenticated to a computer:

- Tell the computer something you know (such as a password).
- Show the computer something you have (for instance, a card key).
- Let the computer measure something about you (for example, your thumbprint).

Authenticity

The assurance that a message was transmitted by the sender.

Authorization

The process of granting permission for a principal to access a resource.

Block Cipher

A cryptographic algorithm that operates on plaintext in groups of bits.

BSAFE

A security toolkit sold by RSA that enables the addition of cryptographic security to any application.

CA

See *Certification Authority (CA)*.

CA Hierarchy

Multiple layers of CAs in which each higher level of CAs vouches for the authenticity of the certificates and / or CRLs from the next lower level of CAs.

Certificate

A formatted data item signed by a trusted party to attest to the validity of the item's information. Public key certificates use a CA's signature to attest that the enclosed public key belongs to the principal identified by the enclosed name.

Certificate Revocation List (CRL)

A list of certificates that have been revoked.

Certification Authority (CA)

A trusted third party that signs a certificate. In the Oracle Security Server, the Oracle Security Repository serves as the certification authority.

Checksum

A short piece of data that is added to a message so that the receiver can check to see if the message was distorted during transmission. Alternatively, to generate the checksum. The Oracle Security Server uses the MD5 algorithm to generate a hash value that is used as a checksum.

Cipher

See *Cryptographic Algorithm*.

Ciphertext

The encrypted form of data.

Cleartext

See *Plaintext*.

Client

A computer or a process that wants to use the services of a system facility or a computer.

Confidentiality

The assurance that only an authorized receiver can read a message.

Credentials

A term used within the Oracle Security Server Manager to refer to an X.509 certificate associated with a particular entity.

CRL

See *Certificate Revocation List (CRL)*.

Cryptanalysis

The art and science of breaking ciphertext.

Cryptanalyst

A person who performs cryptanalysis.

Cryptographer

A person involved in cryptography.

Cryptographic Algorithm

A general procedure for transforming data from plaintext to ciphertext and back again.

Cryptography

The science of providing security for information through the reversible transformation of data.

Cryptology

A branch of mathematics that encompasses both cryptography and cryptanalysis.

Cryptosystem

The combination of a cryptographic algorithm and all possible plaintexts, ciphertexts, and keys.

Database Server

A computer or a process that accepts and processes requests for database information from clients.

Data Encryption Standard (DES)

See *DES*.

Decrypt

To reverse the encryption process: in other words, to restore ciphertext to its original form so that the original message is easily readable.

DES

A block cipher that uses a 56–bit key to encrypt or decrypt data in 64–bit blocks.

Digital Signature

A checksum or hash of a message encrypted with the sending party's private key. The signature is added to the message; the receiving party can use the signature to receive assurance that the original data was not modified in transit and to verify that the data came from the nominal sender.

Distinguished Name (DN)

A string that uniquely identifies a principal, a role, or a path.

DN

See *Distinguished Name (DN)*.

Encrypt

To transform data so that it is unreadable by anyone without the correct decryption key. Encrypted data is also called ciphertext.

Enrollment

The process of making a principal known to a particular application. For example, in the Oracle Security Server, enrollment occurs when a principal's identity is added to the Oracle Security Server Repository, a database server for security data.

Enterprise Authorization

A role that a global user can perform across multiple Oracle8 databases.

Entity

A person, an object, or an event about which information is stored in a database. For example, in the Oracle Security Server, communicating parties such as users and principals are entities.

Global User

A user who needs access to more than one Oracle8 database.

Hash Function

A function that takes a variable–length input string and converts it to a fixed–length output string.

Hash Value

The output string from a hash function. See also *Message Digest*.

Hybrid Cryptosystem

A cryptographic system in which two parties who wish to communicate with each other use a public–key encryption algorithm to authenticate each other and a more streamlined private–key algorithm to transmit bulk data.

IDEA

A block cipher that uses a 128–bit key to encrypt or decrypt data in 64–bit blocks.

Identity

A representation of any entity that does business with the Oracle Security Server.

Integrity

The assurance that a message will not be deleted or altered without explicit authorization that the message's sender.

International Data Encryption Algorithm (IDEA)

See *IDEA*.

Key

A variable parameter of a cryptographic algorithm.

MD5

A hashing algorithm that compresses a message of arbitrary length into a 128-bit message digest.

Message Digest

The output string from a hash function. See also *Hash Value*.

Message Digest 5 (MD5)

See *MD5*.

Mutual Authentication

A process whereby two communicating parties authenticate each other.

Nonce

A unique character string, which usually includes the current date and time, that is only used once.

Nonrepudiation

The condition established by a digital signature under which the sender of a message cannot later claim that it did not send the message.

One–Way Hash Function

A hash function that works in one direction: it is easy to compute a hash value from a pre–image, but it is hard to generate a pre–image that hashes to a particular value.

Oracle Security Server Authentication Adapter

The component of the Oracle Security Server that interfaces with the Oracle Security Repository and oversees the authentication and authorization processes.

Oracle Security Server Manager

The component of the Oracle Security Server that enables administrators to add, modify, and delete information in the Oracle Security Repository.

Oracle Security Server Repository

The component of the Oracle Security Server that stores certificates and roles.

Plaintext

The unencrypted, readable form of data.

Pre–Image

The input string to a hash function.

Principal

A communicating party that has been enrolled in the Oracle Security Server.

Privacy

The ability to keep anyone but the intended recipient from reading a given message.

Private Key

An encryption key that is used only by a limited number of communicating parties, because it needs to be kept secret.

Private–Key Cryptography

A type of cryptography that is based on a single key.

Private–Key Encryption

A technique for encrypting information such that the same key is used in encrypting and decrypting a given message.

Privilege

Authorization for an entity to perform certain actions on certain programs or objects. For example, John may have the SELECT privilege on table EMP within database ITR.

Public Key

The key that is distributed to parties that wish to communicate with the owner of the private key.

Public–Key Cryptography

A type of cryptography that is based on public / private key pairs.

Public–Key Encryption

A technique for encrypting information such that the key used to decrypt the message is different from the key used to encrypt the message.

RC4

A stream cipher that uses a key of any length between 1 and 2048 bits inclusive to encrypt or decrypt a block of text of arbitrary length.

Role

A collection of one or more privileges.

RSA

A public–key cryptosystem that can be used for both encryption and authentication; also, the name of the company that owns the cryptosystem.

Secret–Key Cryptography

See *Private–Key Cryptography*.

Server

A computer or a process that accepts and processes requests from clients. In Oracle documentation, “server” often refers to the Oracle database server.

Server Authorization

A role that has been “identified globally” at an Oracle8 Server.

Session Key

A key that is used to encrypt and/or decrypt the data transmitted during one and only one communication session.

Sign

To add a digital signature to a message.

Signature

See *Digital Signature*.

Single Sign-On

A system capability that enables users to access a number of applications without having to log on and/or present a password to each application.

Stream Cipher

A cryptographic algorithm that operates on plaintext one bit or byte at a time.

Strength

With regard to a cryptographic algorithm, the difficulty an attacker would have deriving plaintext input to that algorithm from the ciphertext output from that algorithm without prior knowledge of the key.

Symmetric–Key Cryptography

See *Private–Key Cryptography*.

TIPEM

A security toolkit sold by RSA that enables the addition of cryptographic security to mail and other messaging applications.

Trustpoint

One or more identities that are considered trustworthy and that can be used to validate other identities. Also, the certificate of a CA, which has been signed by a CA that is higher in the CA hierarchy and theoretically more trustworthy. Also, the CA itself.

Validate

To determine that the signer of a digital signature is legitimate.

Verify

To check to see if the data in a signed message has not been changed and that the data came from the nominal sender.

Wallet

A data structure that contains an X.509 certificate and a public/private key pair.

Web Server

A server that receives anonymous requests from unauthenticated hosts on the Internet and delivers requested information in a quick and efficient manner.

X.500

ITU-T Recommendation X.500 [CCI88c], which defines a directory service.

X.509

ITU-T Recommendation X.509 [CCI88c], a subset of X.500 that specifies the syntax used within Oracle Security Server digital certificates.

Bibliography

[Diffie & Hellman] Whitfield Diffie and Martin E. Hellman: “New Directions in Cryptography.” *IEEE Transactions on Information Theory*, v. IT-22, n. 6, November 1976, pp. 644-654.

[Krawczyk] Hugo Krawczyk: “SKEME: A Versatile Secure Key Exchange Mechanism for Internet.” IBM, Hawthorne, NY, 1995.

[MD5] R.L. Rivest, “The MD5 Message Digest Algorithm,” RFC1321, 1992.

[PKCS1] *PKCS #1: RSA Encryption Standard*. RSA Laboratories, Redwood City, CA, 1993.

[PKCS7] *PKCS #7*: *Cryptographic Message Syntax Standard*. RSA Laboratories, Redwood City, CA, 1993.

[RSA FAQ] *Frequently Asked Questions: Cryptography*--*The Latest from RSA Labs*. RSA Laboratories, Redwood City, CA, 1996.

[Schneier] Schneier, Bruce: *Applied Cryptography*. John Wiley & Sons, 1996.

[X.500] *ITU-T Recommendation X.500* (1993), ISO / IEC 9594: 1995.

[X.509] *CCITT Draft Recommendation X.509*, Omnicom PPI, Philips Business Information, Inc., Potomac, Maryland.

[X.509A] *Draft Amendments DAM 4 to ISO/IEC 9594-2:1995(E), DAM 2 to ISO/IEC 9594-6:1995(E), DAM 1 to ISO/IEC 9594-7:1995(E), DAM 1 to ISO/IEC 9594-8:1995(E)*

Index

Send Us Your Comments

Oracle Security Server Guide, Version 2.0.3

Part No. A54088-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

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

- electronic mail - *ossdoc@us.oracle.com*
- FAX - +1 (415) 506-7226. Attn: Oracle Security Server
- postal service

  Oracle Corporation
  Oracle Security Server Documentation
  500 Oracle Parkway
  Redwood Shores, California 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

____

____

____