# Oracle® Calendar

Administrator's Guide

Release 2 (9.0.4)

**Part No. B10892-01**

June 2003

This book describes deployment, administration and maintenance procedures for Oracle Calendar.

ORACLE®

Oracle Calendar Administrator's Guide Release 2 (9.0.4)

Part No. B10892-01



Contributing Authors: Steve Carbone, Ingrid Pitchen, Robb Surridge, Jennifer Waywell, David Wood

Contributors: George Babics, Thierry Bonfante, Mario Bonin, Chady Chaar, Soo-Yong Chai, Tanya Correia, Marten den Haring, Manon Delisle, Bernard Desruisseaux, Andrew Edwards, Patrice Lapierre, Fréderic Leblanc, Benoit Martel, Alain Petit, Eric Plamondon, Michel Rouse, Costa Siourbas, Ridwan Tan

# Contents

## 3 Calendar Deployment



## 4 Directory Servers for Calendar

## 8 Calendar Users



## 9 Calendar Resources



## 10 Event Calendars

## 15 Node Maintenance



## 16 Monitoring Procedures



## 17 Oracle Calendar Application System



## A Disk Space and Memory



## B Adjusting Calendar Kernel Parameters

## C Security



## D International Support

## List of Figures

## List of Tables

# Send Us Your Comments

**Oracle Calendar Administrator's Guide Release 2 (9.0.4)**

**Part No. B10892-01**

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

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

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

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 633-3836 Attn: Oracle Collaboration Suite Documentation Manager
- Postal service:
  Oracle Corporation
  Oracle Collaboration Suite Documentation Manager
  500 Oracle Parkway, Mailstop 2op5
  Redwood Shores, CA 94065
  USA

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

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

# Preface

Oracle Calendar is scalable calendaring software, based on open standards, for efficiently scheduling people, resources and events. Among other features, it offers real-time lookups and free-time searches, multiple time zone support and UTF-8 encoding to support international deployments, e-mail and wireless alerts, multi-platform support and an extensible Authentication, Compression and Encryption (ACE) framework for enhanced security.

The Oracle Calendar server is the back end to an integrated suite of calendaring and scheduling products. Networked users can use a desktop client (Windows, Macintosh, Linux, Solaris), Web client or Microsoft Outlook to manage their calendars. Mobile users can synchronize their agendas with a variety of PDAs or, with the addition of Oracle's wireless technology, can send and receive calendar entries using a mobile phone.

Oracle Calendar is part of Oracle Collaboration Suite, offering integrated e-mail, voice mail, calendaring and wireless services. For more information on the other components of Oracle Collaboration Suite, please see Oracle's Web site or consult the relevant product documentation.

## Intended Audience

This *Administrator's Guide* is directed at any administrator whose task is the installation, configuration, use and maintenance of Oracle Calendar in general, or of any Oracle Calendar components. This guide documents deployment, configuration and maintenance procedures for calendar components. It is a companion volume to the *Oracle Calendar Reference Manual*, which provides detailed information concerning configuration parameters and command-line administration utilities.

# Documentation Accessibility

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

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

**Accessibility of Code Examples in Documentation** JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation** This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

# Structure

This manual contains 17 chapters, 4 appendices and a glossary:

### Chapter 1, "Oracle Calendar System Administration"

This chapter is an introduction to the administration requirements of Oracle Calendar, and the management tools and utilities available to you.

### Chapter 2, "Calendar Architecture"

This chapter examines the overall structure of the calendar server.

### Chapter 3, "Calendar Deployment"

This chapter outlines the deployment and installation of your calendar server.

### Chapter 4, "Directory Servers for Calendar"

This chapter contains an overview of how the Oracle Internet Directory interacts with the Oracle Calendar server, and information on using third-party directory servers when using the calendar server in a standalone installation.

### Chapter 5, "Calendar Server Administration"

This chapter describes how to manage your calendar server using command-line utilities or the Calendar Administrator.

### Chapter 6, "Setting Up Calendar Nodes"

This chapter describes how to setup and manage a calendar node.

### Chapter 7, "Calendar Node Networks"

This chapter describes how to manage your network of calendar nodes.

### Chapter 8, "Calendar Users"

This chapter describes the different tasks involved in managing users within a calendar server node.

### Chapter 9, "Calendar Resources"

This chapter describes the various tasks involved in creating and managing resources.

### Chapter 10, "Event Calendars"

This chapter describes the various tasks involved in creating and managing event calendars.

### Chapter 11, "Administrative Rights"

This chapter contains an overview of the administrative rights that can be assigned to users, and the methods available for assigning those rights.

### Chapter 12, "Groups"

This chapter describes the different tasks involved in managing groups and group members within a calendar server node network.

### Chapter 13, "Holidays"

This chapter describes the various tasks involved in creating and managing holidays.

### Chapter 14, "Alerts"

This chapter contains general considerations relating to the server-side implementation of reminders and notifications.

### Chapter 15, "Node Maintenance"

This chapter outlines the procedures for server maintenance, server back up and restore and user back up and restore.

### Chapter 16, "Monitoring Procedures"

This chapter describes how you can monitor the Calendar Server by viewing and interpreting log files.

### Chapter 17, "Oracle Calendar Application System"

This chapter provides an overview of the Oracle Calendar application system, including a summary of its components, architecture information and installation considerations.

### Appendix A, "Disk Space and Memory"

This appendix describes the disk space and memory requirements of the calendar server.

### Appendix B, "Adjusting Calendar Kernel Parameters"

This appendix details the modifications that must be made to certain kernel parameters and operating environments in order to ensure that sufficient resources are allocated to the calendar server.

### Appendix C, "Security"

This appendix describes the structure and configuration of the authentication, compression and encryption methods, details additional security considerations for installations using a directory server, and a number of other measures that may be employed to further protect calendar data.

### Appendix D, "International Support"

This appendix contains information relating to international installations of the calendar server.

# Related Documents

For more information, see the following manuals in the Oracle Calendar documentation set:

- *Oracle Calendar Reference Manual*
- *Oracle Calendar Application Developer's Guide*
- *Oracle Calendar Resource Kit*

See also the following manuals in the Oracle Collaboration Suite documentation set:

- *Oracle Collaboration Suite Concepts and Planning Guide*
- *Oracle Collaboration Suite Installation Guide*
- *Oracle Collaboration Suite Configuration Handbook*
- *Oracle Collaboration Suite Release Notes*

# Conventions

In this manual, Windows and NT are both used to refer to the Windows2000 and Windows NT operating systems.

In examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the Return key at the end of a line of input.

The following conventions are also used in this manual:

| Convention | Meaning |
|---|---|
| .<br>.<br>. | Vertical ellipsis points in an example mean that information not directly related to the example has been omitted. |
| . . . | Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted |
| **boldface text** | Boldface type in text indicates a term defined in the text, the glossary, or in both locations. |
| `monospaced font` | This typeface is used for any text that appears on the computer screen or text that you should type. It is also used for file and path names and functions. |
| **Cmd line** | Refers to a procedure executed on the command line (UNIX or NT) using a calendar server utility. |

| Convention | Meaning |
|---|---|
| **Web GUI** | Refers to a procedure executed using the Calendar Administrator, an Web administrative tool. |
| / | Forward-slashes are used to separate directories in a path name, following UNIX syntax. For Windows operating systems, substitute back-slashes "\" for all forward-slashes unless otherwise instructed. |
| < > | Angle brackets enclose user-supplied names and variables. |
| [ ] | Brackets enclose optional clauses from which you can choose one or none. |

# 1

# Oracle Calendar System Administration

This chapter is an introduction to the administration requirements of Oracle Calendar, and the management tools and utilities available to you.



## General Administration Concerns

Administration concerns for Oracle Calendar fall into four broad categories:

- Initial deployment planning, including sizing estimates and operating system configuration. Throughout this guide you will find a variety of information to aid you in designing a deployment that will meet your needs efficiently and effectively, including common deployment scenarios, recommended sizing guidelines, recommended kernel parameter settings for UNIX platforms and more.
- Installing the Oracle Calendar software and performing basic initial configuration. This information can be found in the Oracle Collaboration Suite Installation Guide.
- Performing ongoing system maintenance, such as tuning calendar configuration parameters according to anticipated and observed usage, backing up the calendar database and monitoring the system. Most of these tasks can be performed using a variety of command-line administration tools provided with the calendar server.
- Managing the information in the calendar database, such as adding and modifying calendar users, resources, event calendars, holidays, groups, etc.

Most of these tasks can be performed using either command-line utilities or the Calendar Administrator.

# Oracle Calendar Administrator

The Oracle Calendar administrator is an on-line server management tool that allows users and administrators to manage user accounts, resources, event calendars, groups and holidays in the calendar server database. In addition, the Oracle Calendar administrator provides easy access to basic system administration tasks such as viewing the status of calendar nodes and databases, and starting or stopping nodes and servers.

# Command-line Utilities

The Oracle Calendar server comes with a number of command-line utilities that provide scriptable control over a wide range of calendar server information and features. These command-line utilities are installed in the `/bin` subdirectory within your calendar server's installation directory.

For a complete list of the utilities included with your calendar server and full details on their operation, see the *Oracle Calendar Reference Manual*.

# 2

# Calendar Architecture

This chapter examines the overall structure of the calendar server. An introduction to the concepts and terminology involved is followed by an examination of the function of the calendar server daemons/services, and an illustration of the basic internal operations and processes involved in client connections.



## Overview

This section is a quick introduction to the concepts and terminology at the heart of the calendar server design. This information is intended as a preliminary to the architectural and structural information later in this chapter. More details on these subjects are presented in following chapters.

### Hosts

A host is the physical machine running an installation of the calendar server. It is possible to have more than one installation of the calendar server on the same host, except on Windows NT where only one installation of the server may be present on any one host. When more than one calendar server is installed on the same machine, the two are differentiates by the port number used to access them. This will be reflected by the <`hostname:portnumber`> combination used when accessing a server using a utility. For example:

```
uniping -host host1:5730
```

## Nodes

A node is a local database where the server stores information such as user records, meetings and events. Each node has a specific, unique identification number called the Node-ID. Multiple nodes may exist for the same calendar server. Nodes may also be connected into a node network, allowing users on separate servers to schedule meetings and events transparently with one another. When the calendar server is installed, a node is automatically configured.

## Clusters and master nodes

A cluster is a node network in which one node is designated the “master node”. The master node coordinates network management, finding user accounts on installations spanning multiple nodes and hosts. Use of a cluster type of network is optional.

## Installation options

When the Oracle Calendar Server is installed, by default it is integrated with the Oracle Internet Directory server (OiD) and other components of the Oracle Collaboration Suite. However, the Oracle Calendar Server can be deployed as a "standalone" application, storing all user information in a third party LDAP directory server. In either cases, the calendar server is said to be using an external directory.

The standalone installation also offers the possibility of installing the calendar server with no LDAP directory. In this case, the user information is stored in the calendar server’s database and the calendar server is said to be using an internal directory.

# Daemons/services

The calendar server can contain up to six UNIX daemons / multi-threaded Windows NT services. The calendar server daemons / services are:

- Oracle Calendar Engine
- Oracle Calendar Lock Manager
- Oracle Calendar Synchronous Network Connection
- Oracle Calendar Directory Access Server
- Oracle Calendar Corporate-Wide Services

- Oracle Calendar Server Manager

## Oracle Calendar Engine

The Engine (`uniengd`) accepts and services calendar requests. All communication with the calendar database is handled by the engine. At startup, a series of uniengd daemons / services are started automatically for each node.

The first uniengd process created is the uniengd Controller. It then creates multiple uniengd Listeners. A listener listens for incoming client connections and either launches uniengd session processes or creates uniengd session threads for each client it accepts (depending on the OS of the system). One of the uniengd session belongs to the Synchronous Network Connection daemon (unisncd described later in this chapter).

Since many of the calendar requests can be received at the same time, the servicing engines rely on the Lock Manager to ensure orderly access to the database of the local node (see Oracle Calendar Lock Manager later in this chapter). Therefore the Lock Manager must be running in order for the Engine to function. The Engine must be running in order to operate the calendar server.

## Oracle Calendar Lock Manager

The Lock Manager (`unilckd`) queues and processes the many requests for access to the calendar server's database. The Lock Manager must be running in order to operate the calendar server. On Unix, it is possible to have more than one lock manager per calendar server when more than one node is configured for the same calendar server. By default, one lock manager exists per node. See server parameter `[LCK] maxnodesperlistener` in Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*.

## Oracle Calendar Synchronous Network Connection

There is one Synchronous Network Connection daemon / service (`unisncd`) per server which services every node. It fulfills two roles in the server architecture. First, the `unisncd` is used to maintain open TCP / IP connections between nodes, and to grant those connections to clients that request access to another node. Each connection is unidirectional, from the current node to another node, but not vice versa. It is therefore important to ensure that node connections are set in both directions.

The Synchronous Network Connection daemon / service grants connections on a first in, first out basis. Requests that cannot be processed immediately are put in a

queue. The number of connections between two nodes may be increased or decreased to minimize both the number of connections used and the traffic and connection time between two nodes. If a Synchronous Network Connection daemon / service loses a connection due to network problems, it will attempt to reconnect later.

In its second function, it is used in implementations with an external directory server, the `unisncd` acts as a broker, granting connections to the Directory Access Server (`unidasd`).

### Configuration

For technical information concerning the configuration of the Synchronous Network Connections daemon / service, including a list of parameters and their default values, see Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*. Unless otherwise indicated, the Synchronous Network Connections daemon / service must be restarted in order for a configuration change to take effect.

## Oracle Calendar Directory Access Server

The Directory Access Server (`unidasd`) is used to maintain open connections to an LDAP directory server. Connections to the Directory Access Server are granted by the Synchronous Network Connection (`unisncd`) daemon / service. The `unidasd` daemon / service is not used when there is no external directory. By default, more than one `unidasd` is started when the calendar server is started.

The number of `unidasd` connections to establish to the directory server is defined by the parameter `numconnect` in section `[<YOURHOSTNAME>,unidas]`. If this parameter is set too low, the server may not be able to handle all requests made for directory server operations, in which case end users will get errors of the type “Unable to contact directory server”. For more details on the `numconnect` parameter, see Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*.

## Oracle Calendar Corporate-Wide Services

The Corporate-Wide Services (`unicwsd`) daemon / service replicates data between nodes, provides e-mail notification through an SMTP mail server and wireless notification through Oracle’s wireless services. It also processes server side reminders and synchronizes the calendar database with the directory server when necessary.

unicwsd communicates with other nodes using TCP/IP sockets and named pipe connections. These links provide fast communication. The TCP/IP sockets and named pipe connections are provided and managed by the Synchronous Network Connection daemon/service (unisncd).

By default, two unicwsd daemons/services are created when the calendar is started, one for the replication of calendar data between nodes and one for all the other tasks (mail, reminders, etc.).

## Configuration

For each of the following activities one CWS (Corporate-Wide Services) job can be enabled:

| | |
|---|---|
| Replication: | Node to node data replication |
| Messaging: | Messaging requests for e-mail, wireless alerts, *i*Meeting, etc. |
| SSR: | Server side reminders |
| Snooze: | Handling snoozed requests |
| DirSync: | Synchronizing with Oracle Internet Directory |
| EventSync: | Updating synchronization data for events recently modified. |
| GALSync: | Synchronizing the Global Access List. |

A CWS task is a process that handles one or more jobs.

The server can be configured to support one or many separate processes per job using server parameter [CWS]prioritizedjobs and the maximum number of nodes a Corporate Wide Server task can service can be configured using [CWS] maxnodepertask. For more details on theses parameters, see Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*.

For more information concerning the configuration of the Corporate-Wide Services daemon/service, including a list of all [CWS] parameters and their default values, see Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*. Unless otherwise indicated, the Corporate-Wide Services must be re-started in order for a configuration change to take effect.

## Processing

Whenever a client requires a service from the Corporate-Wide Services daemon, a request is put in a local queue and the appropriate unicwsd will service it

depending on the type of the request. One of the unicwsd daemon / service processes the requests by reading them from the queue.

When a remote node is temporarily shut down, replication requests destined for that node are accumulated in the queue and tagged as on hold until the node becomes available again. Similarly, if for any reason, mail cannot be sent, mail notification requests will accumulate in the queue. The utility unireqdump can be used to output the set of requests currently in the queue of the Corporate-Wide Services daemon / service. For more information on the utility unireqdump, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

## Oracle Calendar Server Manager

The Calendar Server Manager or CSM (unicsmd) provides the capability for remote administration of the calendar server. The following administrative operations will use the services provided by the CSM:

- Starting a remote server
- Stopping a remote server
- Obtaining the status of a remote server
- Stopping a node located on a remote server
- Starting a node located on a remote server

In order to start a server remotely, the server must be in “stand-by” mode. In this mode, all daemons / services are stopped except for unicsmd which is left up and running and waiting for a remote calendar request. If a server is stopped completely (all daemons / services are stopped including unicsmd), it will not be possible to restart it remotely.

At installation, the CSM is configured with the parameters in the [CSM] section of the $ORACLE_HOME/ocal/misc/unison.ini file. The port parameter specifies the TCPIP port on which the unicsmd daemon / service listens. This port needs to be opened (as is the port for uniengd) on any firewall for remote management of the server to be possible. Remote management of the server can be disabled by setting the enable parameter to FALSE.

In a standalone installation of the calendar server, the password parameter must be specified in the [CSM] section of the unison.ini file. This will be the password to access the local calendar server via unicsmd from a remote location. In configurations where the complete Collaboration Suite has been installed, the SYSOP password is used. The ACE (Authentication, Compression and Encryption)

framework is not used when authenticating remotely using the CSM. For more information on ACE, see Appendix C, "Security".

---

**Note:** There is only one password for remote access. Two administrators who remotely manage the same server will use the same password, and there will be no way of determining who did what.

---

The calendar utilities that use the services of the CSM are `unistart`, `unistop`, `unistatus` and the Calendar Administrator. When accessing a remote server, the local utility is invoked with the host name and port number of the remote server's CSM which must be specified using the `-csmhost` option. The CSM password (for standalone installations) or SYSOP password of the remote server must be supplied using the `-p` option. For example:

```
% unistart -n 120 -csmhost hercules:7688 -p pass1
```

The local utility communicates with the remote CSM which will invoke the remote version of the utility. Error codes and output produced by the remote utility will be passed back and displayed by the local utility. However, any log output will only be written to the log files on the remote host. You will need to sign-in to the remote host directly to access the log files. For more information on `unistart`, `unistop`, and `unistatus`, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

The Calendar Administrator can also be used to manage a calendar server remotely. As with the utilities, you will be required to enter the remote host name, CSM port and password.

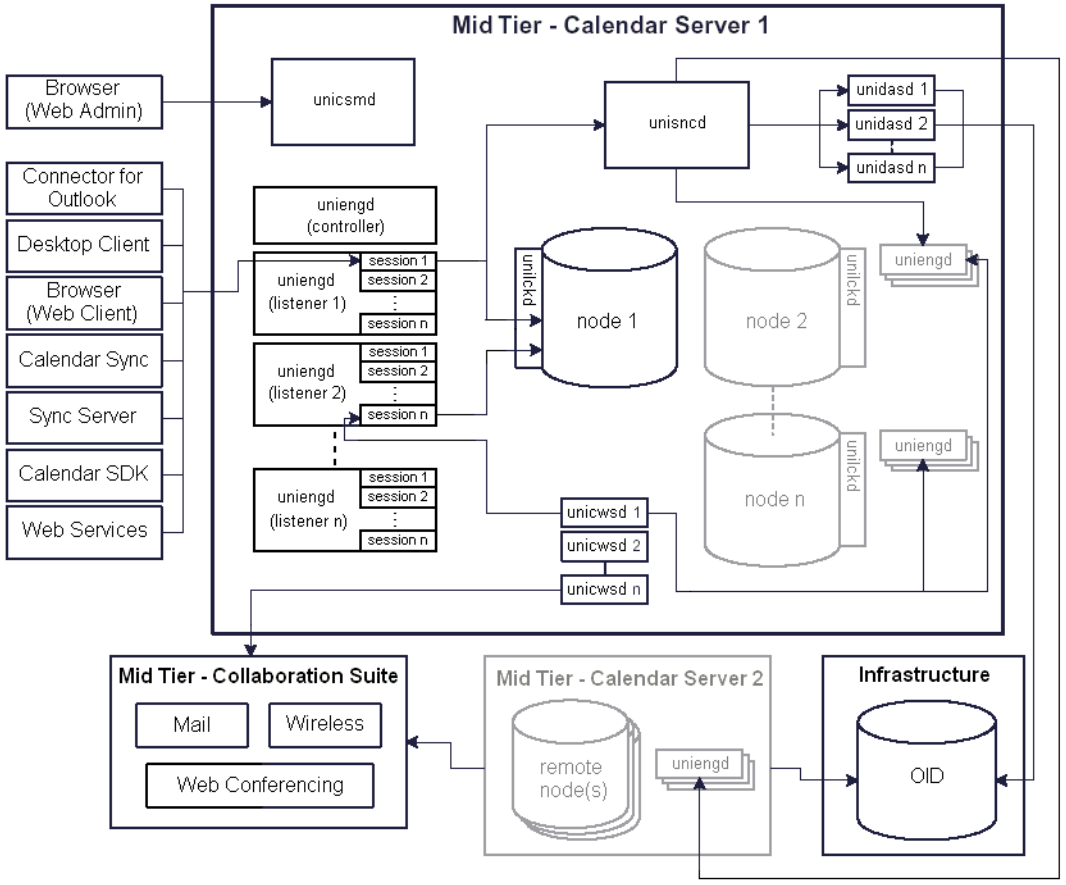
Note that the CSM is an optional part of the calender server. Whether `unicsmd` is up or down, it has no impact on the local calendar operations. In case of CSM failure, normal calendaring operations will not be affected. Only the ability to manage the server remotely will be affected.

The remote administration is cross-platform: a calendar server on a Windows NT system can be started remotely from a calendar server on a Solaris machine and vice-versa.

# Calendar server architecture

When the Collaboration Suite is installed, the default configuration uses the Oracle Internet Directory to store user attributes. All nodes in a node network must use the same directory server. Figure 2–1 shows the Oracle Calendar internal architecture.

***Figure 2–1 Calendar server internal architecture***



## Initial client connection

For each client it accepts, the uniengd listener either forks a uniengd session process or creates a uniengd session thread. All subsequent requests from a client are sent to the client's dedicated `uniengd` session. This is referred to as a "persistent connection".

> **Note:** Web clients do not adhere to this model. Instead, the Web client application running on the Web server establishes a pool of persistent connections to the calendar server at startup. The client divides these connections among the users actively making requests through their Web browsers.

## Client requests

In the course of a user session, calendar and user information may be viewed, created, modified, or deleted. These operations involve either reads from, or writes to, node databases. The client's `uniengd` session handles reads and writes on the local node (the node containing the user) differently from reads and writes on remote nodes.

**Reads and writes on the local node** To perform reads or writes on the local node, the client's `uniengd` session requests access to the node database (`$ORACLE_HOME/ocal/db/`<node name>) from the `unilckd`. When the `unilckd` grants access, the `uniengd` session performs the read or write operation. It then relinquishes access to the node database, and if necessary, returns data to the client.

**Reads from a remote node** To read information from a remote node, the client's `uniengd` session requests a connection to the remote node from the `unisncd`. When the `unisncd` hands it a connection to that node, the client's `uniengd` session sends the read request to the `uniengd` server at the other end of the connection (see "Connections to nodes" later in this chapter). The remote `uniengd` server then requests access to its local node database from its `unilckd`. When the `unilckd` grants access, the `uniengd` server retrieves the information, relinquishes access to the node database, and sends the information to the requesting `uniengd` session. The requesting `uniengd` session receives the information, returns the connection to the `unisncd`, and hands the information to the client.

**Writes to a remote node** Writes to a remote node arise when the user adds, modifies, or deletes calendar and/or user information. In this case, the `uniengd` session process or thread places a request in the queue of the local node (see "Node

request queue" later in this chapter for a discussion of the node queue). Another uniengd session associated with this local node services these requests in the same manner as reads on a remote node.

## Connections to nodes

unisncd establishes persistent connections to other nodes on startup, using information in the $ORACLE_HOME/ocal/misc/nodes.ini file to determine the number of connections to establish to each node. It establishes a connection by sending a request to the uniengd daemon / service (the uniengd daemon / service on the local host for nodes on the local machine, and the uniengd daemons / services on remote hosts for nodes on remote hosts). That uniengd daemon / service creates a uniengd session process or thread to handle requests to the specified node. A uniengd session sits on the end of each connection to a remote node. Each of these sessions obtains access to their node database from unilckd.

## Node request queue

Each node has a request queue associated with it. This request queue is maintained within the node database and used by all uniengd sessions associated with the node. These request queues contain requests for alerts and e-mail notification, and requests for replication of data (resulting from creation, modification, and / or deletion of calendar information).

unicwsd manages the request queue of each node on the local host. On startup, unicwsd sends a request to the uniengd daemon / service for a uniengd session for each node on the local machine. The uniengd daemon / service then creates one uniengd session process or thread per node to service each node's request queue. The diagram shows one of these uniengd sessions.

unicwsd examines each request in the request queue and determines how to handle it. If the request is for a mail notification, it hands the mail to the mail server. Similarly, it passes requests for wireless reminders and notifications to the Oracle9*i*AS Wireless server. If the request is for a write to a remote node, it hands the request to the uniengd session servicing the request queue of the node. unicwsd also deletes redundant requests in order to optimize performance.

## Connections to the directory server

On startup, unisncd reads the numconnect parameter from the [YOURHOSTNAME, unidas] section of the $ORACLE_HOME/ocal/misc/unison.ini file to determine the number of connections to establish to the LDAP directory server. It then sends a request to the unidasd

daemon / service. The `unidasd` daemon / service spawns a `unidasd` server for each of the requested connections. When a `uniengd` session requires directory server access, it requests a connection to a `unidasd` server from `unisncd`. `unisncd` passes a connection to the requesting `uniengd` session, which retrieves the necessary information from `unidasd` and subsequently returns the connection to `unisncd`.

# 3

# Calendar Deployment

This chapter outlines the deployment and installation of your calendar server. Prior planning is an integral part of a successful implementation in your organization. It is highly recommended that you read this chapter before installing the server to ensure an installation that is customized to your needs.

The following sections cover the information that you need to get your server up and running:

- Deployment
- Pre-installation checklist
- User licenses
- Installation notes

## Deployment

To realize the optimal Oracle Calendar server configuration for your organization, you must first evaluate who your users are, how they should be organized, and how the product will be installed and managed. Consider the following factors:

- Number of users
- Logical divisions of users
- Grouping users to create nodes
- Product administration

## Number of users

The first step in planning a successful deployment or “roll-out” of Oracle Calendar server is to determine the number of potential Oracle Calendar users in your organization. If growth is anticipated in your organization, factor this into your calculations. The final tally forms the basis for the value you supply for configured users in later calculations.

The categories of users are:

- **Configured users:** people with user accounts on an Oracle Calendar server node.
- **Logged-on users:** users who are connected to a node but not actively making requests of the database. This figure is generally estimated to be anywhere from 33–50% of configured users. Try to forecast how your users will use the calendaring application. For example, if everyone starts work at the same time, you might anticipate a period of peak usage in the morning where up to 75% of all users will be logged on at once. Also, a number of users may choose to stay logged on all day, keeping the calendaring application open in the background to permit quick and frequent access.
- **Active users:** logged-on users who are making an access request to the database. To estimate the number of active users at any point in time, take 10–25% of the total number of configured users. As with logged-on users, base this number on your highest estimate of peak usage.

> **Note:** Special consideration must be taken for users of Web clients. The Web client application running on the Web server establishes a pool of persistent connections to the calendar server at startup. Several users will share this pool of persistent connections when making requests through their Web browsers.

**Acme Co. example** To illustrate the planning process for your Oracle Calendar server implementation, we will use a fictitious company called Acme Co. The administrator at this company has chosen to make her estimates of logged-on and active users high to ensure that she has adequate resources and that the users can expect uniformly good performance.

***Table 3–1 Acme Company: user base***

| User category | Estimates |
|---|---|
| Configured users | 16,000 |

*Table 3–1 Acme Company: user base*

| User category | Estimates |
|---|---|
| Logged-on users | 8,000 (50% of configured users) |
| Active users | 4,000 (25% of configured users) |

## Logical divisions of users

Once you have estimated your user base, the next step is to group these users according to location and function. Here it is important to identify not only geographic divisions, but also functional or other administrative divisions within your organization. You should use both geographic and administrative divisions to group your users into nodes.

**Acme Co. example** In our Acme Co. example, the total user population of 16,000 is distributed in the following manner:

*Table 3–2 Acme Company: geographic and administrative user divisions*

| Location | Number of Users | Divisions |
|---|---|---|
| Los Angeles | 12,000 | 7,000 Engineering / 5,000 Administration |
| New York | 1,000 | 600 Marketing / 400 Administration |
| Chicago | 500 | 500 Marketing |
| Seattle | 2,000 | 1,500 Engineering / 500 Marketing |
| Vancouver | 500 | 500 Marketing |

## Grouping users to create nodes

A node is created when an instance of the calendar server is installed. A node is a calendar database containing agendas and information for users and resources. With the logical divisions among your user base clearly delineated, you are now ready to group your users into nodes. Before making these decisions, however, a number of factors must be considered:

- **Node size**

  The maximum capacity of an Oracle Calendar server node is 64,000 configured users; however, for performance reasons, this limit is not recommended except under exceptional circumstances of infrequent user connections. The recommended capacity per node and per server is heavily dependent on

hardware and configuration. In an environment supporting any combination of Oracle Calendar Desktop clients (Windows, Mac, Linux, Solaris) and Oracle Connector For Outlook, 10,000 users is a good working figure for maximum configured users. For environments consisting only of Web clients, use 20,000 as a base number. Use caution in determining an appropriate number for a mixed environment. For more information on tuning these numbers, contact a technical consultant or Oracle support representative.

See Appendix A, "Disk Space and Memory" for any additional node size restrictions.

- **Network issues**

  Although server-to-server calendar communication requires low network bandwidth, in order to obtain acceptable performance for users accessing a remote server, a network bandwidth of 64 Kbps or higher is suggested. If this is not possible, it may be wise to consider installing a local server.

- **Scheduling between nodes**

  More server resources are required when scheduling meetings between users on different nodes in an Oracle Calendar node network. For this reason, it is good practice to group users who work together on one node and thereby minimize the number of meetings involving users from other nodes.

- **User migration**

  Although it is possible to move individual users from node to node, the process can be lengthy and may alter or remove some information. Minimize the need to move individual users as a result of either reaching maximum node capacity, or the need to split a node according to logical divisions. For a more detailed discussion of the `unimvuser` utility, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

- **Administrative considerations**

  While the bulk of calendar server administration can be done remotely, there are tasks related to system maintenance that might require an on-site administrator. If you do not have personnel to manage back-up media and system problems at a branch office, then it is probably not a good idea to locate a server there.

  The time required to administer your calendar node network is also affected by the necessary repetition of some tasks. Certain features, such as holidays, are specific to each node. The tasks associated with these features, such as adding holidays, must be done separately on each node.

- **Directory server**

  Each node in a calendar node network that is linked to a directory server must point to the same directory server.

**Acme Co. example** Our administrator has attempted to integrate all of the preceding variables with her user base calculations, arriving at the following configuration. In achieving this balance, she has considered a number of factors specific to her situation:

- The Los Angeles user base (12,000) is too large to group in a single node, thus she opted to create two nodes, following the administrative divisions, on two separate servers.
- The Chicago office is expected to steadily decrease as the sales force there is relocated to New York's office. Since the administrator wants to minimize the time required for administrative tasks, and not manage two nodes, she groups the New York and Chicago users in one node on a server located in New York. Having users in two different time zones on the same node will require a minor configuration change for the Chicago users to enable them to set their time zone from the client (see the `[LIMITS] settimezone` parameter in Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*).
- The final two nodes will be located on one Unix machine in Seattle. Two instances of the Calendar Server will be installed, one for all Seattle users, and the other for the Vancouver users, each with their own port number. Although there is no performance advantage in splitting the users between two calendar server installations on a single host, a number of other factors have contributed to this decision. The on-site administrator is located in Seattle and the network bandwidth between the Seattle office and the Vancouver office is sufficient for the expected traffic. The Vancouver office is on a separate node, however, as it is expected to grow and eventually maintain its own support division. Thus, it is anticipated that the node will eventually migrate to a server in Vancouver.

The final configuration:

***Table 3–3 Acme Company: node distribution***

| Node | Server Location | Server name | User Base |
|---|---|---|---|
| Node 1 | Los Angeles | host1 | LA: Engineering Division (7,000) |
| Node 2 | Los Angeles | host2 | LA: Administration Division (5,000) |

*Table 3–3 Acme Company: node distribution*

| Node | Server Location | Server name | User Base |
|---|---|---|---|
| Node 3 | New York | host3 | NY: Marketing (600) and Administration (400) Divisions<br>Chicago: Marketing Division (500) |
| Node 4 | Seattle | host4 | Seattle: Engineering (1,500) and Marketing (500) Divisions |
| Node 5 | Seattle | host5 | Vancouver: Marketing Division (500) |

See Appendix A, "Disk Space and Memory", for information concerning memory and disk requirements for your installation.

## Product administration

As a final task in this deployment exercise, determine who will be responsible for the different tasks which are part of setting up and maintaining an Oracle Calendar system. The major tasks are:

- system administration on the UNIX or NT server (including monitoring and backups)
- adding, modifying, and deleting users, resources and event calendars
- manage administrative and public groups
- resource administration (assigning designates, creating categories)
- holiday administration
- front-line support
- client training

Different levels of administration tasks can be assigned to calendar users. For more information on administrative access rights and how to grant them, see Chapter 11, "Administrative Rights".

# Pre-installation checklist

To ensure a quick deployment and minimize later tuning, a number of configuration issues should be considered before installation. Calendar server behaviour can be controlled by parameters set in the $ORACLE_

HOME/ocal/misc/unison.ini file. For more information on these parameters, see Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*.

- **Kernel parameters**

  Operating system kernel parameters must be evaluated, and if necessary tuned, for each installation. See Appendix B, "Adjusting Kernel Parameters" for information concerning the relevant parameters for each supported operating system, the procedure used to alter the current values, and the formulae used to derive correct settings for your installation.

- **Client connections**

  The [ENG] maxsessions parameter determines the number of available client connections to the calendar server. Set it high enough to accommodate the traffic and expected usage of each node, but be aware that setting this value too high will waste system resources.

- **Resource Relative DN**

  Installations using external LDAP directory servers can specify a location in the LDAP hierarchy in which all resources will be located by default. Consult the documentation on the [LDAP] resourcerelativedn parameter for details.

- **Attachments**

  The [LIMITS] allowattachments parameter enables or disables the ability of the calendar clients to attach files to events or tasks. If attachments are allowed, they may be limited in size using the [LIMITS] maxattachmentsize parameter.

- **Group administration**

  The server offers four different group types: personal, members-only, public and administrative. All users have the right to create personal and members-only groups. The rights to create public and administrative groups must be assigned by the administrator. See Chapter 12, "Groups" for more details on the differences between the group types and the methods used to change default administration rights.

- **ACE security framework**

  The calendar server's Authentication, Compression and Encryption (ACE) framework is an extensible system ensuring the security and integrity of all information passing from server to server and between servers and clients. By default, the ACE framework is enabled and cannot be disabled unless you are

using a standalone installation of the calendar server; in which case use the `[ACE] frameworkenable` parameter to disable it.

- **Resource scheduling**

  Resources can either be set up on a first-come first-served basis where double-bookings are not permitted, they may be set up to allow conflicts to occur. Resources can also be restricted to only a few users or be available to all but requiring approval. The default value for the `[ENG] allowresourceconflict` parameter prohibits double-bookings. See Chapter 9, "Calendar Resources" for more details on managing resources.

The following table presents a list of the major items to consider before installing the calendar server. In the case of a standalone installation of the calendar server with no external directory, only the first 6 items need to be considered:

***Table 3–4 Installation information checklist***

| Item | Accepted Values | Mandatory or Optional | Default Value |
|---|---|---|---|
| Node-ID | Recommended Node-ID ranges:<br>1-5: Evaluation<br>6-100: Test<br>101-9999: Permanent<br>10000-59999: Future use<br>60000+: Reserved<br>(NOTE: this number must be unique across all connected nodes) | Mandatory | 1 |
| Node Alias | A descriptive word of up to 32 characters (no spaces) | Optional | N / A |
| Time Zone | See Appendix F, "Time Zone Table" of the *Oracle Calendar Reference Manual* | Mandatory | N / A |

***Table 3–4 Installation information checklist***

| Item | Accepted Values | Mandatory or Optional | Default Value |
|---|---|---|---|
| Number of Concurrent Users | Any number between 15 and 2000(NT) or 5000(UNIX) | Mandatory | 100 |
| Mail Notification | Enabled (Yes) or Disabled (No) | Mandatory | Yes |
| Mail Host | Any host | Mandatory if mail notification enabled | local host |
| Node (SYSOP) Password | Up to 15 alphanumeric characters in length | Mandatory | N/A |
| Base URL for Directory Server | A URL in this format: ldap://<LDAP host>:<LDAP port>/<Base DN> | Mandatory | ldap://<local host>:389/<no default for Base DN> |
| Base DN | The point in the directory hierarchy from which searches are performed | Mandatory | N/A |
| SuperUser DN | User with "unrestricted access". Must be a DN already in the directory server | Mandatory | none |
| Calendar server Administrators' Parent DN | Any DN, offset from the base DN | Optional | If present, `unison.ini` value; otherwise N/A |
| Calendar server Administrators' Group DN | A new group created under the base DN | Mandatory | If present, `unison.ini` value; otherwise "Calendar Server Admins" |

# User licenses

Your calendar server has an unlimited user license. Subject to hardware capabilities and the technical limitations on node size, you may create as many user, resource and event calendar accounts as you need.

# Installation notes

Ensure that you complete all the instructions the Oracle Collaboration Suite *Installation Guide*.

## Distributed installations

Multiple instances of the calendar server can be installed on the same Unix or Linux host (not on Windows). Whether one instance or many instances of the Oracle Calendar is installed on one host, each instance will include many components (even in the standalone mode) which will be installed on the same host. These components include the Calendar Server, Oracle Calendar Administrator, Oracle HTTP Server, Oracle Calendar Web client, Oracle Calendar Web Services and Oracle Calendar SDK. If you wish to run different components on different hosts (for example, to run Web calendar clients on a different host from the calendar server) you must keep the following in mind.

- When the Calendar Server is installed in a standalone mode, Web clients identify themselves to the calendar server using a shared key stored in both the Web client and server configuration files. This shared key must match exactly across all Web clients and server instances. As this key is generated automatically by the installation procedure, and is different for each install, you will have to perform this configuration manually. See the Appendix C, "Security" for more information on shared keys.
- If you wish to use multiple calendar server nodes, whether on the same host or distributed across multiple hosts, you must connect the nodes into a network. See Chapter 7, "Calendar Node Networks", for details.

# 4

# Directory Servers for Calendar

The Oracle Calendar server can share information seamlessly with other back end infrastructure components such as e-mail servers through the use of Oracle Internet Directory. Benefits include centralized user administration and consistency of information across multiple applications.

This chapter contains an overview of how the Oracle Internet Directory interacts with the Oracle Calendar server, and a discussion of some more specific configuration issues. It also contains information on using third-party directory servers when using the calendar server in a standalone installation.

The following topics are dealt with in this chapter:

- Using a directory server
- Extending the directory schema
- Base DN
- LDAP groups and distribution lists
- Global Address List (GAL)
- Access control for administrators
- Binding
- Fail over
- Security using SSL
- Changing the attribute used for SSO login

# Using a directory server

Directory servers store information about users, and make this information available to various other services, such as e-mail or calendar applications. Directory servers like Oracle Internet Directory centralize user information; in the event that an administrator has to add or change user accounts, the task only needs to be done in one place. At the same time, user information is always kept consistent between all the applications that use the directory server; your user profile in your e-mail client, for example, will be the same as what you see in your calendar client.

Oracle Calendar server integrates with Oracle Internet Directory, which uses the Lightweight Directory Access Protocol (or LDAP) for accessing user records. Those records are kept in a hierarchical tree, in which each record is accessible through a particular path (called a *Distinguished Name*, or DN) that specifies the record's unique location in the tree. This is analogous to the way a filename and path specifies the location of a particular item in a file system.

# Extending the directory schema

Directory servers have schemas that define the information they store. These schemas consist, amongst other things, of *objects* and *attributes*. Objects are representations of real-world people or things, and attributes are defining characteristics of those objects. Directory servers come with preset schemas to represent people and things, providing attributes for names, addresses, phone numbers, and so on, that define a 'person'. You can also extend these schemas with new object classes and new attributes to reflect any information you care to store.

When you install Oracle Internet Directory as a part of Oracle Collaboration Suite, it's infrastructure is preset with all the objects and attributes needed by the calendar server. For more information on standalone installations of the Oracle Calendar Server with third party directory servers, see Chapter H, "Calendar Extensions to Directory Server Schema" of the *Oracle Calendar Reference Manual*.

# Base DN

The calendar server requires a base DN, which is a location in the LDAP directory server under which all calendar users, resources and administrators will be located, and under which all directory searches will be performed.

This base DN is defined by the `[LDAP] basedn` parameter in the `$ORACLE_HOME/ocal/misc/unison.ini` configuration file. The location for resources in

the LDAP directory relative to the calendar server base DN is specified in the unison.ini file by the [LDAP] resourcerelativedn parameter. Event calendar accounts are defined by [LDAP] eventcalrelativedn.

In a group search operation, it's possible to override the search base using the [LDAP]group_searchbase parameter in the $ORACLE_HOME/ocal/misc/unison.ini file. This is particularly useful if the directory group entries are stored in a central location.

For more information on server parameters, see Appendix C, "Server Parameters" in your *Oracle Calendar Reference Manual.*

# LDAP groups and distribution lists

The Calendar Server supports directory groups. To expose groups to calendar clients, set the [LDAP] group_enable parameter in the unison.ini file and add a corresponding filter to locate these groups using the [LDAP] groupfilter parameter. Also include the attribute that contains the URI to locate the members using the [LDAP]groupmemberlistattribute parameter. Users will be able to see all groups in the directory server, and any members of these groups who are also calendar users. These groups will appear to users as Public groups. By default, all groups in the directory server located under the calendar base DN will be listed in the calendar client. However, since the calendar client will only display calendar users, if a given LDAP group has only non-calendar users as members, that group will appear as an empty group, with no members.

Using the preceding parameters, you can control which groups in the directory should be accessible to the calendar user. One way is to create a custom object class (one with the name "calendargroup" for example) and add that custom object class to the group entries that you wish to expose. The parameter [LDAP]groupfilter should then be set to "(objectClass=calendargroup)".

If you have a deployment when group entries are centrally located, it's advisable to set the parameter [LDAP]group_searchbase to the subtree where the group entries are located. This will improve the performance of the group search operation.

The calendar server also supports dynamic groups. A dynamic group is a group where the membership information is stored in an LDAP URL. When the calendar server detects that the value of a group member is an LDAP URL, the calendar server will expand the URL to establish the group membership.

To expose dynamic groups to calendar clients, add a corresponding filter to locate that type of group using the `[LDAP] groupfilter` parameter. Also include the attribute that contains the URI to locate the members using the `[LDAP]groupmemberlistattribute` parameter.

Oracle Mail distribution lists can also be exposed to users as public groups if the Oracle Collaboration Suite is installed. Users will then be able to use the distribution lists to send e-mails (Oracle Connector for Outlook) or to invite a group of users (Oracle Desktop Calendar clients). To give users access to these groups, set the `[LDAP] group_dlfilter` and `[LDAP] group_dlenable` parameters. For more information on these server parameters, see Appendix C, "Server Parameters" in your *Oracle Calendar Reference Manual.* See also Chapter 12, "Groups".

# Global Address List (GAL)

The Global Address List, or GAL, is the interface used by Microsoft Outlook to select users (including non-calendar users), resources and event calendars from the directory server. When a user uses the GAL, it attempts to display all the entries in the directory. The user can filter out some entries. This filtering is performed on the client side by the Outlook interface after all the entries have been downloaded to the client.

The Oracle Connector For Outlook works with the Calendar Server to improve the efficiency of the GAL. The server caches the GAL entries in order to reduce the number of requests made to the directory server. Parameters can be set in the server configuration file to control the cache and how GAL requests are serviced.

In the `[CWS]` section of the `unison.ini` file, the parameter `galsyncinterval` controls the frequency of the cache refresh. In the `[ENG]` section, `gal_enable`, `gal_enumsize`, `gal_enableldapsearch`, `gal_refreshinterval`, `gal_view` let you control other aspects of the GAL. For more information on these server parameters, see Appendix C, "Server Parameters" in your *Oracle Calendar Reference Manual.*

# Access control for administrators

Using calendar administration tools, calendar administrators have the ability to modify entries in the directory server. The level of access depends upon the access control restrictions set on your directory server.

## Oracle Internet Directory

In a full Collaboration Suite installation, the Calendar Server creates two types of administrator entries: *calendar application entity* (CAE) and *calendar instance administrator*. The CAE entry is created under the subtree "`cn=Application Entities, cn=Calendar, cn=Products, cn=OracleContext`", where as the calendar instance administrator entry is created under "`cn=Admins, cn=Calendar, cn=Products, cn=OracleContext, <subscriberdn>`".

By default, the calendar instance administrator entry is not granted any explicit rights in the directory server. This entry is used for password validation only. The CAE entries have full access rights under the two calendar product containers, "`cn=Calendar, cn=Products, cn=OracleContext`" and "`cn=Calendar, cn=Products, cn=OracleContext, <subscriber DN>`". The CAE entry is also a member of the group "`cn=OracleDASEditUser, cn=groups, <subscriber DN>`" which grants it the right to modify user entries.

For a given instance of a calendar server, the distinguished names of the CAE and of the calendar instance administrator are defined in the `[LDAP]` section of that instance's `unison.ini` configuration file. The distinguished name of the CAE is defined by the `[LDAP] applicationentitydn` parameter and the distinguished name of the calendar instance administrator is defined by `[LDAP] admindn`.

## Other directory servers

In a standalone Calendar Server installation, where the calendar server is installed with a third party LDAP directory server, there are two ways to give directory access to administrator users.

One way is to grant default access control to an LDAP group which will be specified by the `[LDAP]admingroup` parameter. Calendar administrators can then inherit this right by becoming a member of this group. When the SYSOP grants administrative rights to a user, he must also add him to that group. For more information on the `[LDAP]admingroup` parameter, see Appendix C, "Server Parameters" in your *Oracle Calendar Reference Manual.*

The second way to give non SYSOP users access to the directory server is to provide a DN and a password which will be used by the calendar server to sign-in to the directory for directory administrative write operations. Set the parameter `[DAS] dir_usewritednforadmin` to `TRUE` and set parameters `[LDAP] writedn` and `[LDAP] writednpassword`. Make sure the specified DN has full write access to the calendar user, resource and event calendar information in the directory. For more details on these parameters, see Appendix C, "Server Parameters" in your *Oracle Calendar Reference Manual.*

By default, calendar administrators have full access rights for all entries under the calendar base DN. If you are familiar with your LDAP directory server's access control information, you may wish to configure it to restrict the administrator permissions.

Your directory server may or may not support access control restrictions at the necessary level of granularity. Consult your directory server documentation for details on configuring access control information. You must ensure that certain permissions remain at a minimum to avoid calendar client and server errors.

# Binding

In a standalone Calendar Server installation, when users query the calendar server for lists of other users, resources or groups, the calendar server binds to the directory server using the "anonymous" profile.

If your directory server does not allow anonymous binding, or if you want to prevent the calendar server from binding anonymously, you can use the `[LDAP] binddn` and `[LDAP] bindpwd` parameters in the calendar server configuration file `unison.ini` to specify a directory server user account and password with which to bind.

This bind DN will be used for all read access to the directory server (such as user and resource searches); however, users and administrators still bind to the directory server as themselves when performing modifications to user, resources or event calendar records.

Remember that you must encrypt the value of the bind password before including it in the `unison.ini` file. Use the `uniencrypt` utility with the **-s** argument, and include the entire output, enclosed within double quotes as the parameter value. For details on the use and syntax of `uniencrypt`, see Appendix E, "Utilities", in the *Oracle Calendar Reference Manual*.

# Fail over

You can specify an alternate directory server host to be used in the event that your main host becomes unavailable. The `unison.ini [LDAP] host` parameter lists the directory server hosts in preferred order; if multiple hosts are listed and the first host listed becomes unavailable, the calendar server will instead attempt to connect to the next host listed.

Since the calendar server maintains a persistent connection to the directory server, the calendar server will not attempt to reconnect to the primary directory server

when it becomes available again. Instead it will wait for the next recycle time, specified in the parameter `[DAS]dir_connectrecycletime`. Once the recycle time is reached, the calendar server will terminate the current directory server connection and attempt to re-establish a connection to the primary directory server. If the primary directory server is available at that time, it will establish the connection.

Consult the documentation on the `[LDAP] host` parameter in Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual* for full details.

# Security using SSL

For greater security, the connections between your calendar server and directory server will be protected by default using the Secure Sockets Layer (SSL) protocol. Without the use of SSL, passwords may be sent across the wire in clear text.

In a Collaboration Suite installation, the calendar server will configure and establish an SSL connection to the Oracle Internet Directory if it's available. In a standalone Calendar Server installation, the following procedures are required to establish an SSL connection to a 3rd party directory server.

1.Make sure your calendar server is not running before you proceed.

2.Run Netscape Navigator on the calendar server host, and access your directory server using its secure port. For example:

`https://<yourDSname>:<yourSSLport>/`

In the pop-up site certificate window, select **Accept** the certificate forever.

3.Find the cert7.db file on your system (likely in the `netscape/users/<username>` directory). Take note of this file's location.

4.Edit your `$ORACLE_HOME/ocal/misc/unison.ini` file, and add or modify the following parameters:

`[LDAP]`

`security=TRUE`

`secure-port=636` (encryption port set on your directory server)

`certdbpath="<pathname>"` (the value from step 3)

For more information on these parameters and their correct values for your operating system, consult Appendix C, "Server Parameters" in your *Oracle Calendar Reference Manual*.

5.Restart your calendar server. All further communications with the directory server will use SSL.

# Changing the attribute used for SSO login

In a Collaboration Suite installation, one directory server attribute is designated as the attribute that the Oracle Internet Directory uses to authenticate Single Sign-On (SSO) logins. The calendar server configuration file stores this information with the parameter `attr_uid` in the `[LDAP]` section.

If you change this attribute at any time on the directory server after you have installed the calendar server, it is vital that you perform the following calendar server configuration change. If you do not, users will be unable to sign in using Web clients, and the integrity of your calendar database may be threatened.

1. Stop all calendar servers in your network.
2. Edit the `$ORACLE_HOME/ocal/misc/unison.ini` file on each calendar server host in the network, and modify the value of the following parameter:

   ```
   [LDAP]
   attr_uid = <attribute_used_for_login>
   ```

   This parameter controls the directory server attribute the calendar server uses as a unique user identifier.
3. Restart all servers stopped in step 1.

# 5

# Calendar Server Administration

This chapter describes how to manage your calendar server. All administrative tasks and procedures are executed using command-line utilities or the Calendar Administrator.

Note that all server administration performed through the command-line utilities provided should be carried out on UNIX platforms as the `unison` user created by the server installation program. On the Windows platform, they should be run under the System Account. The calendar server daemons should all run under this user name as well. It is not recommended to run the calendar server as the `root` user.

This chapter documents the following administrative tasks:

- Using the Calendar Administrator
- Starting and stopping the calendar server
- Checking server status
- Viewing current user activity
- Changing the SYSOP (node) password
- Other administrative tasks

## Using the Calendar Administrator

A majority of the administration tasks can be performed using the Calendar Administrator WEB interface. This WEB application is normally installed in the `$ORACLE_HOME/ocad` directory. The Calendar Administrator can be used by the administrator (SYSOP) or calendar users who have administration rights.

The Calendar administrator interface is divided into two sections, accessible through tabs appearing on the top right of the page. Access to specific commands and functionality is determined by the user's assigned calendar administrative rights. The Calendar Management tab allows the user to manage users, resources, event calendars, and groups. The Server Administration tab lets the administrator start and stop individual nodes and calendar servers, and organize holiday management.

For more information on all tunable parameters available to configure your Oracle Calendar Administrator, see appendix D, "Calendar Administrator Parameters" which lists all parameters located in the initialization file `$ORACLE_HOME/ocad/bin/ocad.ini`.

# Starting and stopping the calendar server

If you are using an external LDAP directory server (the `unison.ini` parameter `[DAS] enable=TRUE`), your directory server must be running before you can start the calendar server. If you have enabled e-mail notification in the clients (the `unison.ini` parameter `[LIMITS] mail=TRUE`), a mail server should be running.

Start the calendar server by using the `unistart` utility. Stop it by using the `unistop` utility. For full information on use and syntax of this utility, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

**Example** To start the calendar server (bringing up five daemons if using an internal directory and six daemons if using an external LDAP directory), execute the following command:

```
% unistart
```

**Example** For an orderly shutdown of the calendar server, execute the following command:

```
% unistop
```

**Note:** The number of client connections, the number of processes running, and the volume of network traffic all affect the amount of time that the server takes to stop.

# Checking server status

To view the current status of the calendar daemons/services and servers, run the `unistatus` utility. For full information on use and syntax of this utility, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

**Example** To verify the status of the calendar server's daemons/services, run the following command:

```
% unistatus -d
    UID   PID  PPID STIME   TIME      COMMAND CLASS INFORMATION
    tin   6772 228  1:41:21 0:0:0.156 unisncd Listener
    tin   4368 228  2:32:23 0:0:0.187 unicwsd Controller  3 task(s)
    tin   6756 4368 2:32:27 0:0:0.125 unicwsd Task  SSR
    tin   7680 4368 2:32:27 0:0:0.203 unicwsd Task  Messaging
    tin   7196 228  1:41:28 0:0:0.46  unicsmd Listener
    tin   6712 228  1:41:17 0:0:0.78  unilckd Listener  0 DB sess
    tin   6692 228  1:41:18 0:0:1.875 uniengd Listener  3/100 sess
    unistatus: the calendar server is up
```

# Viewing current user activity

To view current logged-on users, run the `uniwho` utility. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

**Example** To view a list of current client connections, execute:

```
% uniwho
    PID           ADDRESS  NODEID      XITEMID  USER
   7721     193.77.49.162   20004      20004,2  CWSOP,na
  14668     193.77.49.223   20004    20004,295  Alexander,James
  10237      193.77.49.44   20004    20004,142  Addison,Thomas
  TOTAL STANDARD SHARED CONNECTIVITY
      3        2      0            1
```

# Changing the SYSOP (node) password

### Cmd line

To change the password of the SYSOP (the administrator of a node) or any other user, run the `unipasswd` utility from the command line. For full information on use and syntax of this utility, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

**Example** To change the password for node 34, execute:

```
% unipasswd -n 34
Enter old password:
Enter new password:
Re-enter new password:
unipasswd: Password changed successfully
```

---

**Note:** In an Calendar installation with no LDAP directory, nodes are created with no SYSOP password. The password must be set after the node is created. However, a SYSOP password is set for a node created where the calendar server installation is using a directory server.

---

# Other administrative tasks

Use command-line utilities for a variety of other less frequent tasks. Some of the utilities that you may want to be familiar with once the calendar server is up and running are:

- UNIDBBACKUP — to backup the calendar server.
- UNICHECK (UNIX only) — to verify the presence of all necessary files and directories, and check the settings for permissions, owner and group information.
- UNICLEAN (UNIX only) — to remove or correct any problems found running `unicheck`. Transient files will be removed and permission and ownership settings restored to the default.
- UNIDSUP — to report the status of the directory server.
- UNILOGONS — to display calendar server signon / signoff statistics.

- UNIREQDUMP — to view and delete requests in the Corporate-Wide Services (CWS) queue.
- UNIRMOLD — to remove old data from the calendar server database.
- UNIWHO — to display information on signed-on calendar users.

See Appendix E, "Utilities" in the *Oracle Calendar Reference Manual* for a description of the function and syntax of all utilities included with the calendar server.

# 6

# Setting Up Calendar Nodes

A node is a database containing agendas and information for users, resources and event calendars. A node network is a set of two or more connected nodes. More than one node can exist on a single calendar host. This situation commonly occurs where a group of users requires a different time zone, or when there is a logical division that the administrator wishes to maintain within a group of users in the same time zone.

This chapter covers the following topics:



## Creating a calendar node

Each node is identified by a unique numeric key called the node-ID. Most administrators set one or more descriptive node aliases that may also be used when connecting to the calendar server. A SYSOP (node administrator) password restricts access to the calendar account used for all node management tasks. Each node has a default time zone.

To create a node, you will need the following information:

- **Node-ID:** The node ID can be any number between 1 and 49999. When setting up a node, it is important to note that the node-ID cannot be changed once the node has been created. Furthermore, an existing local node will be deleted if a new local node is given a node-ID currently in use on the same computer. A warning prompt will be issued before this action is taken. **Node-IDs are unique locally and across the node network.** Two nodes with the same node-ID cannot be connected in a network.

- **Node Alias:** A descriptive word of up to 32 characters containing no spaces. When multiple nodes are configured on a server, users may need to indicate which node they want to connect to. Since, in general, a name is easier to remember than a numeric node-ID, aliases can be configured.
- **Node Time Zone:** Every node has a time zone associated with it. If you do not specify a time zone, your node will be created using the default time zone set during installation of the calendar server. See Appendix F, "Time Zone Table," in the *Oracle Calendar Reference Manual*, for a complete list of countries with their corresponding time zone notation.
- **SYSOP Password:** The password for the SYSOP, or node administrator. If you are not using a directory server, the password cannot have more than 15 alphanumeric characters in length. Otherwise the limit is 63 characters.
- **Directory Manager Password:** If you are using a directory server in a standalone calendar installation, the directory manager password must be provided when creating a node. This is the password associated with the LDAP directory server manager defined by the `[LDAP]mgrdn` parameter.

### To create a node:

1. Use the `unistop` utility introduced in Chapter 5, "Calendar Server Administration", to bring down the calendar server. Please note that the server must be down in order to create a node successfully.
2. Run the `uniaddnode` utility. For full information on use and syntax of `unistop` and `uniaddnode`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

**Example** To create a node with a node-ID of 144 and alias "Publications":

```
% uniaddnode -n 144 -a publications -w dmpasw
uniaddnode: Database initialization done
uniaddnode: node [144] has been successfully initialized
```

An entry similar to the following would now exist in the `$ORACLE_HOME/ocal/misc/unison.ini` file. Note that the name and version fields are for internal use and are automatically generated during node creation. The values in these fields must not be modified.

```
[144]
aliases = publications
name = N2
version = A.02.61
timezone = EST5EDT
```

3. Use the `unistart` utility introduced in Chapter 5, "Calendar Server Administration", to restart the calendar server. For full information on use and syntax of `unistart`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

# Deleting a calendar node

Deleting a node manually requires an advanced knowledge of the calendar server. Before attempting to remove a node, familiarize yourself with the contents of the chapters referenced in the following procedure.

### To delete a node manually:

1. Make an archive of the node if necessary. Use the archive tools available on the system you are using if you want to back up only the node you are deleting. If you want to back up the entire calendar database, see "Server back up and restore" in Chapter 15, "Node Maintenance", for more information.
2. Remove the node from the node network (if it is part of one) by editing the `$ORACLE_HOME/ocal/misc/nodes.ini` file and applying the change. Understand the contents of Chapter 7, "Calendar Node Networks", before attempting to do this.
3. If you are running a directory server, delete all users, resources and event calendars on the node (using `uniuser -ex`).
4. Shut down the calendar server.
5. Delete the entire `$ORACLE_HOME/ocal/db/nodes/<Nx>` directory, where `<Nx>` is the value of the `name` parameter in the appropriate node section of the `$ORACLE_HOME/ocal/misc/unison.ini` file. For example, if you are deleting the node with node-ID 144, `<Nx>` is the value of the `name` parameter in the `[144]` section of the `unison.ini` file. For details on `unison.ini` parameters, see Appendix C, "Server Parameters" in the *Oracle Calendar Reference Manual*.
6. Delete the corresponding node section in the `$ORACLE_HOME/ocal/misc/unison.ini` file. For example, if you are deleting the node with node-ID 144, delete the `[144]` section of `unison.ini`.
7. Restart the calendar server.

**Note:** If you are using a third-party directory server, you may want to remove all references to reserved calendar users for the deleted node. Use your directory server's `ldapmodify` utility or other tools available with your directory server.

# 7

# Calendar Node Networks

A node network is a set of two or more connected nodes.

Once a connection between two or more nodes has been defined, all searches produce listings of configured users and resources from both local and remote nodes. This basic information is maintained on each computer in the node network. All calendaring data for each user and resource, however, resides only on that item's local node, eliminating space and consistency problems created by replicated databases. All exchanges of this information between nodes is done in real time, making local or remote location on a network completely transparent to the user.

When setting up a node, it is important to note that the node-ID cannot be changed once the node has been created. Furthermore, an existing local node will be deleted if a new local node is given a node-ID currently in use on the same computer. A warning prompt will be issued before this action is taken. **Node-IDs are unique locally and across a node network.** Two nodes with the same node-ID cannot be connected in a network.

This chapter details the following node network management tasks:

- Connecting nodes
- Setting up a master node
- Moving a node
- Coexistence of LDAP and non-LDAP nodes
- Caveat: LDIF differences between UNIX and NT
- Coexistence of nodes with and without Oracle9iAS Wireless

# Connecting nodes

The network configuration is stored in one file (`$ORACLE_HOME/ocal/misc/nodes.ini`), and is managed using the `uninode` utility or the Oracle Calendar Administrator WEB interface. The file must reside on only one of the host members of the node network, and commands can only be executed from this host. Both the `uninode` utility and the centralized administration tool are used to add or delete nodes in the node network, and to set the number of TCP/IP connections between the nodes, which are maintained by the Synchronous Network Connection (SNC) daemon/service.

The number of connections to establish between each pair of nodes in a node network is dependent in large part on the size and configuration of your installation. As a general guideline, smaller implementations are well served by a configuration in which a single node has two connections to each node in the network. All connections are one-way, so a network of 3 nodes would have a total of 12 connections:

- Node A has 2 connections to node B and 2 connections to node C
- Node B has 2 connections to node A and 2 connections to node C
- Node C has 2 connections to node A and 2 connections to node B
- Total number of connections = 12

A different set of guidelines applies to larger installations which fit within the following configuration parameters:

- hardware configuration adequately supports the demands of the software
- clients used are not Web-based (i.e. Windows, Macintosh or Motif clients, or Oracle Connector for Outlook)
- configured users per host does not exceed 5,000
- logged-on users per host does not exceed 2,500
- logged-on users per node does not exceed 1,000
- logged-on users per node is greater than 250
- connected nodes per host does not exceed 4
- number of nodes in a network does not exceed 10

For any installation in this category, the general recommendation is to establish 4 connections each way between a local node and a node on a remote machine, and 3

connections each way between nodes on the same machine. Most installations can probably optimize this further.

The uninode utility is used for all node management tasks for the calendar server. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

### To connect two or more calendar server nodes:

1. Select the server which will be used to manage your node network.
2. Run uninode -init to create and initialize a $ORACLE_HOME/ocal/misc/nodes.ini file.

```
% uninode -init
checking password for node 24, please wait...
Enter SysOp password:

connected to clio, node 24
extracted existing connection information

created the "$ORACLE_HOME/ocal/misc/nodes.ini" file
initialization succeeded
```

3. The newly created $ORACLE_HOME/ocal/misc/nodes.ini file contains the following header with a summary of uninode syntax and connection rules.

```
#Description    File (nodes.ini)
#-------------------------------------------------------
#INCLUDE NODE:  + H=Vancouver/N=10/ALIAS=Finance
#EXCLUDE NODE:  - H=Toronto/N=20
#NODE FOR MAIL: + H=Montreal/N=30/S=unison/G=unison/OU1=CS&T/OU2=R&D
#ABSOLUTE RULE: all:2
#RELATIVE RULE: Vancouver->Montreal:+1
```

4.

**NT**

Use a text editor to add the nodes to be connected and the rules governing the connections to the nodes.ini file. See "Syntax" and "Connections and Rules" later in this chapter for a discussion of the nodes.ini syntax, connections, and rules. Once you exit the text editor, run uninode -apply to create the node network.

**UNIX**

Run `uninode -edit` and use a text editor to add the nodes and the rules governing their connections to the `nodes.ini` file. See "Syntax" and "Connections and Rules" later in this chapter for a discussion of the `nodes.ini` syntax, connections, and rules.

```
+ H=clio/N=24
+ H=clio/N=25
included:2
~
no errors detected
2 node(s) to ADD
edit the temporary node file again? (y/n) n
LAST CHANCE TO ABORT, process changes? (y/n) y

checking if all nodes are up
connected to clio, node 24
connected to clio, node 25

Processing node 24
connected to clio, node 24
connected to clio, node 25
added 24->25, TCP/IP connection

Processing node 25
connected to clio, node 25
connected to clio, node 24
added 25->24, TCP/IP connection
Do you want to update the directory of items (Actual =
0/Expected = 7)? (y/n) y
```

```
placed a request in the CWS queue to get node 24 user
directory

0 connection error(s), 0 processing error(s)

Applying connection configuration: Successful
```

## Syntax

The `$ORACLE_HOME/ocal/misc/nodes.ini` file contains the list of the nodes and the list of rules that describe the network configuration. Any lines in the file which begin with the symbol "#" are considered comments and are ignored.

The minimal syntax for a node is:

```
+ H=<HOSTNAME>/N=<NODE-ID>
```

or

```
- H=<HOSTNAME>/N=<NODE-ID>
```

The <HOSTNAME> can be either a fully-qualified domain name, a machine name, or a numeric IP address. Do not, however, mix these in the same `nodes.ini` file. If you choose to use fully-qualified domain names, you must continue to use fully-qualified domain names throughout the file to avoid problems.

A node can either be included (+) in the network or excluded (-) from the network.

The following fields can be used to specify a node:

| Field | Description | Mandatory or optional |
|---|---|---|
| H | Host name | mandatory |
| N | Node-ID | mandatory |
| ALIAS | Alias for Node-ID | optional |
| GR | Group name | optional |

If an alias is specified, it will be easier for users on all nodes of the network to identify where remote users are located, as this information will be displayed by the calendar client.

The group name is given by the administrator and is used to refer to a group of nodes. The interaction between the nodes of a specific group should be greater than

with nodes of other groups. In most cases, a group name will represent a geographical area or a company subdivision.

Three predefined groups can be used:

- "`all`" refers to all included (+) and all excluded (-) nodes
- "`included`" refers to all included (+) nodes
- "`excluded`" refers to all excluded (-) nodes

## Connections and Rules

Two kinds of rules can be used. The first is used to specify the default number of connections between all nodes or between nodes within a group.

For example, say we have the following nodes in our `nodes.ini` file:

```
+H=mis-can1/N=1
+H=mis-usa1/N=2
+H=mis-eur1/N=3
+H=mis-eur2/N=4
```

To specify that two connections be established from each node to each of the other nodes, we use the predefined group "`included`" and add the following line.

```
included:2
```

The second kind of rule specifies the number of connections, from one node or group to another node or group.

```
N1->N2:X
```

N1 and N2 may either be host names, node-IDs, or group names. X may either be an absolute number of connections (0, 1, 2, 3...), or a relative number of connections (+1, -1, +2, ...). Rules are interpreted from the first to the last rule of the file. Consequently, the rules should be arranged from the most general to the most specific.

For example, to apply a more specific rule to this set of nodes, the group (`GR`) field can be useful in selecting these nodes.

```
+H=mis-can1/N=1/GR=Canada
+H=mis-usa1/N=2/GR=USA
+H=mis-eur1/N=3/GR=Europe
+H=mis-eur2/N=4/GR=Europe
included:2
Europe:+1
```

In the preceding example, we are able to add an additional connection (for a total of 3) to each of the European nodes relative to the absolute value defined on the preceding line.

Had we not wanted to use groups, we could also have said:

```
+H=mis-can1/N=1
+H=mis-usa1/N=2
+H=mis-eur1/N=3
+H=mis-eur2/N=4
included:2
mis-eur1 ->mis-eur2:3
mis-eur2 ->mis-eur1:3
```

Note that in this case we must specify the number of connections in each direction as SNC connections are unidirectional.

### Adding a node to the network

Replace the exclusion sign (-) of the host with the inclusion sign (+).

### Deleting a node from the network

Replace the inclusion sign (+) of the host with the exclusion sign (-).

> **Warning: Deleting a node from the node network, even temporarily, will result in the loss of all remote records created on this node.**

### Increasing or decreasing the number of connections between nodes

To modify the number of connections between nodes, make the necessary changes to the rule entry. It is possible to add, delete or modify a rule entry.

Example: To increase the number of connections from Los Angeles to Cupertino by 2, add the following rule to the end of the file:

```
angeles->cupertino:+2
```

## Tuning connections

It is possible to tune the socket and / or transaction options of some or all of the connections to specific hosts and / or nodes. This may be useful in node networks with nodes distributed over multiple machines, where there are significant variations in performance among the connections. These variations may be due to differences in machine speeds, the speeds of the network links, the loads on the various machines, and / or the loads on the various calendar servers.

Tune the socket and / or transaction options by adding sections to the `unison.ini` file to define the relevant connections.

Each section heading has the syntax:

`[<host>, unieng, <node>]`

where

- <host> is either the name of a host, or the keyword `ALL` to indicate all hosts in the node network
- <node> is either a node-ID, or the keyword `ALL` to indicate all included nodes in the node network

For example, the section `[murphy,unieng,all]` contains parameters which tune all connections to all nodes on host murphy.

Each section contains one or more of the following parameters:

```
tr_block = 0
tr_recv_timeout = 60
tr_send_timeout = 0
so_rcvbuf = 0
so_sndbuf = 0
so_keepalive = FALSE
```

Each of these parameters overrides the value of the equivalent parameter in the `[SNC]` section of the `unison.ini` file. For example the `tr_block` parameter overrides the `[SNC]` `snc_tr_block` parameter. See the equivalent parameters in the `[SNC]` section in Appendix C, "Server Parameters" in the *Oracle Calendar Reference Manual* for an explanation of each of these.

**Note:** The `tr_block` parameter refers to the block size for communications between two `uniengd` servers and not, as for the `snc_tr_block` parameter, for communications between a `uniengd` server and a `unidasd` server.

# Setting up a master node

You can set up a master node for your node network to control network management and ease the finding of user accounts on installations spanning multiple nodes and hosts. Use of a master node is optional.

**Note:** Your node network **may not** at any time contain more than one master node.

### To set up a master node:

1. Stop all calendar servers that host the nodes in your network.
2. Edit the `unison.ini` file on the calendar server host that contains the node you wish to configure.
3. Set the value of the `[CLUSTER]` `masternode` parameter to the node-ID of the desired node.
4. Restart your calendar servers.

# Moving a node

Entire nodes can be moved from one host to another. The following must be taken into account when moving a node:

- Big-endian vs. little-endian hosts
- Versions of the calendar servers
- The node network
- Moving from internal to external directory

### Big-endian vs. little-endian hosts

Moving a node between a big-endian host and a little-endian host (or vice-versa) requires a database conversion. Contact Oracle support for more information about

the conversion utilities `unib2lendian` (big-endian to little-endian), and `unil2bendian` (little-endian to big-endian) utilities.

***Table 7–1 Oracle Calendar server 5.4 platforms***

| Big-endian | Little-endian (Intel processors) |
|---|---|
| Solaris | Linux |
| HP-UX | Windows |

## Versions of the calendar servers

Moving a node from an earlier version of the calendar server to a later version requires the use of the `unidbconv` database conversion utility. Note that it is not possible to move a node from a later version of the calendar server to an earlier version.

## The node network

The procedure for moving a node between node networks differs from that for moving a node within a node network. This difference is in the utilities the procedure uses to update the node network information.

While most node network management is performed using `uninode` or the Calendar administrator, removing a node from a node network using either of these tools results in data loss. Namely, any calendar data created by a user in the node that is removed is deleted from all other nodes in the network. This is fine when moving a node between node networks. However, when you move a node within a node network, you want all nodes in the network to preserve all data related to the node. In this case you prevent the data loss by using the `unidbfix` utility to update the node network configuration. See "Performing the move" later in this chapter for the exact procedure to follow in each case.

---

**Note:** If you are running a directory server, all nodes in a node network must point at the same directory server.

---

## Moving from internal to external directory

If you are moving a node from a calendar server that uses an internal directory to one that uses an LDAP directory server (or vice versa), you should consult Oracle support for assistance.

## Performing the move

The following procedures describe moving a node between hosts which:

- are both either big- or little-endian
- are both running the same version of the calendar server
- are both using the internal directory or both using a directory server

The first procedure describes moving a node within a node network. The second procedure describes moving a node from one network to another. Although both procedures describe moving a single node, each can be adapted to moving several nodes.

### To move a node within a node network:

1. Stop both calendar servers. Do not restart either until instructed to do so later in this procedure.
2. Run `unidbfix -c` on the node you want to move to ensure that the database is not corrupted.
3. Make an archive file of the `$ORACLE_HOME/ocal/db/nodes/`<Nx> directory, where <Nx> is the value of the `name` parameter that appears in the section of the `unison.ini` file configuring the node you want to move.
4. Copy the archive file to the new host and, using the same archiving tool, restore the directory. Verify that the `$ORACLE_HOME/ocal/db/nodes/`<Nx> directory now exists on the new host.

   If a node with the <Nx> name already exists on the target server, you may choose to rename the node you are moving. Use the lowest unused letter-number combination. For example, if the last node on the target server is named `N6`, rename the new node `N7`.
5. Remove the section configuring the node from the `$ORACLE_HOME/ocal/misc/unison.ini` file on the old host and add it to the `$ORACLE_HOME/ocal/misc/unison.ini` file on the new host.

   Remember that if you chose to rename the node in step 4, you must also rename the section configuring the node. Use the same letter-number combination you selected in step 4.
6. If you are moving a node between two different big-endian platforms or two different little-endian platforms (e.g. from HP-UX to Solaris), copy the `unison.dbd` and `vista.ctb` files from the `$ORACLE_`

HOME/ocal/db/nodes/Nempty/perm/ directory on the old host to the same directory on the new host.

### Steps 7 to 12 update the node network information

7. Stop all other calendar servers in the node network. Do not restart any of these until instructed to do so later in this procedure.
8. Run `unidbfix -export -n all` on each of the hosts in the node network. This creates a `remotenode.ini` file in each of the node database directories (i.e. in each `$ORACLE_HOME/ocal/db/nodes/`<node>`/perm` directory). This file contains information about all nodes remote to node <node>.
9. Edit the node entry for the moved node in the `remotenode.ini` file of each node in the node network, replacing the old host name with the new host name.
10. Run `unidbfix -k` on the newly-moved node to create the key files.
11. Run `unidbfix -import -n all` on each host in the node network. This updates the node database for each node in the node network.
12. Edit the `$ORACLE_HOME/ocal/misc/nodes.ini` file for the node network to reflect the host name change.
13. If you are running a directory server, update the directory server using the `ldapmodify` tool, changing the old host name to the new host name on the moved node. This attribute exists for each SYSOP special user.
14. Remove the `$ORACLE_HOME/ocal/db/nodes/`<Nx> directory on the old host.
15. Start all calendar servers stopped during this procedure.

### To move a node between node networks:

1. Remove the node from its current network. Run `uninode -edit` to edit the `nodes.ini` file and apply the change.
2. Stop both calendar servers. Do not restart either until instructed to do so later in this procedure.
3. Run `unidbfix -c` on the node you want to move to ensure that the database is not corrupted.
4. Make an archive file of the `$ORACLE_HOME/ocal/db/nodes/<Nx>` directory where <Nx> is the name of the node you are moving.

**Warning: To avoid overwriting an existing node, be certain that the name of the node you are moving does not already exist on the new host.**

5. Copy the archive file to the new host and, using the same archiving tool, restore the directory. Verify that the `$ORACLE_HOME/ocal/db/nodes/<Nx>` directory now exists on the new host.
6. Remove the section configuring the node from the `$ORACLE_HOME/ocal/misc/unison.ini` file on the old host and add it to the `$ORACLE_HOME/ocal/misc/unison.ini` file on the new host.
7. If you are running a directory server, update the directory server using the `ldapmodify` tool, changing the old host name to the new host name on the moved node. This attribute exists for each SYSOP special user.
8. Remove the `$ORACLE_HOME/ocal/db/nodes/`<Nx> directory on the machine which originally contained the node.
9. Start all calendar servers stopped during this procedure.
10. Add the node to the node network. Run `uninode -edit` to edit the `nodes.ini` file and apply the change.

# Coexistence of LDAP and non-LDAP nodes

For nodes with and without LDAP connections to coexist in a network, the most recent version of the calendar server in the node network must manage that network. Furthermore, if there is more than one of the most recent version, and any use a directory server, one of those must manage the node network. Recall that all nodes in a node network must point at the same directory server.

The following procedure outlines the steps you must execute when you have a node network where all nodes currently connect to an external directory, and you want to introduce internal-directory nodes into the network. All calendar servers are assumed to be the most recent version.

To integrate the new internal-directory nodes into the node network, execute the following steps:

1. Back up all nodes in the existing node network. See "Server back up and restore" in Chapter 15, "Node Maintenance", for more instructions.

2. Shut down all servers hosting nodes that use a directory server (where the `unison.ini` parameter `[DAS] enable=TRUE`).
3. Edit the parameter `[ENG] dir_internal_nodes` in the `unison.ini` file, on all calendar servers in the node network which use a directory server, to include all non-LDAP nodes.

   For example: with four nodes in your network, nodes 10000 and 10001 on the calendar server using a directory server, and nodes 10002 and 10003 on calendar servers using internal directories, the `unison.ini` file on the calendar server using a directory server would contain the following parameter:

   ```
   [ENG]
   dir_internal_nodes = {10002, 10003}
   ```

   **Warning: Incorrect use of the [ENG] dir_internal_nodes parameter can have serious consequences.**

4. Bring up all servers once the changes are complete.
5. Run `unidssync` to synchronize the LDAP nodes with the directory server.
6. Run `uninode` to add the non-LDAP nodes to the node network.
7. Run `unidssync` on a regular basis (at least once a week for most installations) to keep the LDAP nodes synchronized with the directory server.

# Caveat: LDIF differences between UNIX and NT

Slight differences in the UNIX and NT LDIF file formats must be understood in order to successfully transfer data from a NT to a UNIX server. Before importing NT-generated LDIF files to UNIX, ensure that:

- Any control characters are removed (must change CR/LF to NL).
- The "NT User" object class is removed.

# Coexistence of nodes with and without Oracle9*i*AS Wireless

If a master node is present in your node network, clients will query that master node for available server functionality such as wireless capabilities. If your master node server is set up with Oracle9*i*AS Wireless but the other nodes in your network are not, then clients of users whose accounts reside on those other nodes will

behave as if wireless capabilities are enabled, but users will encounter errors trying to use those capabilities. Likewise, if your master node is not set up with Oracle9*i*AS Wireless but the other nodes in your network are, then all users' clients will hide or disable wireless capabilities (since the master node tells them no wireless capabilities are enabled on the server).

It is therefore recommended that you ensure that your master node has the same wireless capabilities as the other nodes in your network.

# 8

# Calendar Users

This chapter describes the different tasks involved in managing users within a calendar server node. Whether user information is stored internally or in an LDAP directory, the administrative procedures required are similar.

Administrators of installations using LDAP directories must be familiar with user creation and management on their directory server, or should refer to the appropriate on-line help. In an external directory context, users are generally added to calendar server nodes from the directory server, but it is also possible to migrate existing information from a calendar node to a directory server.

The following topics are dealt with in this chapter:

- Creating calendar users
- Managing calendar users
- Deleting calendar users
- Managing user defaults
- Setting up e-mail notification
- Granting access rights from one user to another
- Other user configuration options

## Creating calendar users

Each person who plans to use calendar services must have a profile on the calendar server. Once a user's profile has been created and added to a node, that person can then use any calendar client to connect to the server and manage his/her personal agenda.

## Adding users from a pre-populated directory server

Before you can give a calendar account to users, they must already exist in the directory server. Adding calendar services implies using the existing directory data to create calendar profiles on the calendar server node.

### Web GUI

Creating calendar accounts for existing directory users can be done through the Calendar Administrator. Sign in to the node on which you want to create the account, select **Users** from the main screen, and click **Provision Calendar Service to User** on the Users page. The Directory Users page will appear where you can list users in the directory who do not have a calendar account. Search for the user or users to whom you wish to provision calendar services. You can click **Go** to list all users or select a filter and enter a value in the Search edit box to limit the search. When the user is listed, click the "Provision Calendar Service" icon in the Actions column for that user. To provision services for more than one user at a time, select them in the Select column and then click the **Provision Calendar Service to User** button at the upper right of the list.

### Cmd line

Adding calendar users from an existing directory server using the utilities is a two-step process. The first step is to identify all directory server users who are not calendar users. The `unidssearch` utility will search the directory server DNs and return all entries without the attribute `ctCalXItemId`. These users can then be added to a calendar server node using the `uniuser` utility. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

**Determine which users on the directory server have not yet been added to a calendar server node:** Use `unidssearch` to search the directory server. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

```
% unidssearch -c 10
A DID=cn=Lan Nguyen, ou=Research, o=Acme, c=US
A DID=cn=James Alexander, ou=Research, o=Acme, c=US
A DID=cn=Chris Robbins, ou=Research, o=Acme, c=US
A DID=cn=Thomas Addison, ou=Administration, o=Acme, c=US
A DID=cn=Claire Roslyn, ou=Administration, o=Acme, c=US
A DID=cn=Denis Tremblay, ou=Administration, o=Acme, c=US
A DID=cn=Maija Laine, ou=Finance, o=Acme, c=US
A DID=cn=Elizabeth McKinley, ou=Finance, o=Acme, c=US
A DID=cn=Walter Chen, ou=Finance, o=Acme, c=US
```

```
A DID=cn=Oliver Maxwell, ou=Finance, o=Acme, c=US
```

**To add users one at a time:** Use `uniuser -add -user`. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

```
% uniuser -user -add "DID=cn=James Alexander, ou=Research, o=Acme, c=US" -n 134
Enter SysOp password:
uniuser: added: "cn=James Alexander, ou=Research, o=Acme, c=US"
```

### To add several users:

1. Create a file of all users in the directory server who are not calendar users. The number of non-calendar users returned by a search may be limited by maximum search result settings on the directory server. You can also limit the scope of the search, as in the following example where 5 users are selected from the directory server. The greater-than symbol redirects the output of `unidssearch` to a file named `userslist`.

   ```
   % unidssearch -c 5 > userslist
   ```

2. The file created may then be modified, filtered or added to as required and according to a set format and syntax. Additions are made in X.400 format. For a complete description of the X.400 keys, fields and syntax, see the `uniuser` documentation in Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

   ```
   A DID=cn=Chris Robbins, ou=Research, o=Acme, c=US/G=Chris
   A DID=cn=Thomas Addison, ou=Administration, o=Acme, c=US/G=Thomas
   A DID=cn=Claire Roslyn, ou=Administration, o=Acme, c=US/G=Claire
   A DID=cn=Denis Tremblay, ou=Administration, o=Acme, c=US/G=Denis
   A DID=cn=Maija Laine, ou=Finance, o=Acme, c=US/G=Maija
   ```

3. Attach all users in the "userslist" file to the specified node.

   ```
   % uniuser -ex userslist -n 134
   Enter SysOp password:
   uniuser: added "cn=Chris Robbins, ou=Research, o=Acme, c=US/G=Chris".
   uniuser: added "cn=Thomas Addison, ou=Administration, o=Acme,
   c=US/G=Thomas".
   uniuser: added "cn=Claire Roslyn, ou=Administration, o=Acme, c=US/G=Claire".
   uniuser: added "cn=Denis Tremblay, ou=Administration, o=Acme, c=US/G=Denis".
   uniuser: added "cn=Maija Laine, ou=Finance, o=Acme, c=US/G=Maija".
   ```

## Adding users to the internal calendar server directory

In a standalone installation of the calendar server, where no LDAP directory server is used, user profiles are added to the internal calendar directory. A user password must be supplied.

### Web GUI

Use the Calendar Administrator to add users to a node. Click the **Calendar Management** tab and then on **Users**. To add a new user click **Create Calendar Account** on the far right. Enter the user information and click **Apply**.

### Cmd line

Use the `uniuser` utility with the **-add** and **-user** options to add users to the calendar server's internal directory. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

### Example

```
uniuser -user -add "S=Addison/G=Thomas/I=W/O=acme" -n 786 -p syspsw
uniuser: added "Addison,Thomas,W"
```

## Adding calendar users to a directory server

Instead of adding users from the directory server to the calendar server, the administrator may wish to take one or more existing calendar databases and export the user and resource data in an LDIF format that is then used to populate the directory server.

Contact Oracle support for assistance and utilities to handle the migration of all calendar users to the directory server.

# Managing calendar users

Administrators can modify user attributes using the `uniuser` utility or using the Calendar Administrator. Attributes include user information such as the user name, e-mail address, telephone number, and other personal information. It also includes access rights (designate, viewing rights, etc.), administrative rights (manage holidays, users, server, etc.) and alert attributes (notifications).

A user's calendar account can be disabled. When an account is disabled, the owner cannot sign-in and does not have access to his agenda. The calendar data of

disabled accounts is not deleted and will be accessible again once the user account is re-enabled.

### Web GUI

Use the Calendar Administrator to view and modify a user's calendar attributes. To modify an existing user, click the **Calendar Management** tab and then on **Users**. Search for the user you want to modify using the search box. Click **Go** to list all users or select a filter and enter a value in the Search edit box to limit the search then click **Go**. Find the user in the list and click the pencil icon in the **Actions** column. To modify attributes that are not calendar-specific, use your Oracle Internet Directory administration tools or third-party directory servers.

### Cmd line

You may view and modify a user's calendar attributes using the `uniuser` utility with the **-mod** option or **-s** option. To modify attributes that are not calendar-specific, use your Oracle Internet Directory administration tools.

To disable a calendar account, use `uniuser` with the `-mod` option. Example:

```
% uniuser -mod -user "S=Smith/G=John" -m "ENABLE=FALSE" -n 23
```

To grant access rights from one user to another use the `uniaccessrights` utility. See **Granting access rights** later in this chapter. To change a user password, the `unipasswd` utility can be used. User passwords can also be modified using the Oracle Internet Directory administration tools.

For full information on the use and syntax of `uniuser` and `uniaccessrights`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

## Deleting calendar users

When a user is deleted from a node, the user's directory entry and records are removed from the local node. This means that all data owned by the user, including any meetings or groups, will be deleted. The user will no longer appear in others' agendas, nor will any meetings and other calendar entries owned by this user remain. Any remote directory listings and remote copies of calendar data owned by the user will also be removed.

To preserve meetings or other calendar entries owned by the user, you should transfer ownership of these meetings to another user before the deletion.

### Web GUI

Use the Calendar Administrator to transfer calendar data from one user to another and then to delete the user from the server. Click the **Calendar Management** tab and then on **Users**. Search for the user you want to delete using the search box. Click **Go** to list all users, or select a filter and enter a value in the Search edit box to limit the search then click **Go**.

To transfer data from this user to another, first click the pencil icon in the Actions column for this user. Click **Transfer Calendar** at the bottom of the menu on the left. In the "Recipient for Calendar" page, click **Go** to list all users, or select a filter and enter a value in the Search edit box to limit the search then click **Go**. Find the target user, and click on the Transfer icon in the Actions column for that user. From the "Calendar Data to Transfer" page, select the types of calendar entries you want to transfer and click **Apply**.

Once the data has been transferred you can proceed with the deletion. Click the **Users** tab on the top left. To find the user to delete, use the Search filter and edit box. To delete one user or more, select the users by clicking on the corresponding checkbox in the **Select** column and then click **Delete** at the top right.

### Cmd line

Remove one or more users from the calendar server node using the `uniuser -del -user` (single deletion) or `uniuser -ex` (multiple deletions) commands. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

If the user is the owner of calendar entries (meetings, day events, etc) that you do not wish to delete, you might transfer these entries to another user before deleting the user. Use `uniuser` with the **-transfer** option to transfer ownership of selected entries in the user's agenda to another user.

For example, if a manager who controlled group scheduling leaves the company, you might transfer all non-personal entries from his agenda to the person replacing him. Use the **-event**, **-task**, **-group** and **-folder** options of `uniuser` to define which type of calendar data to transfer.

To preserve a copy of the user's complete agenda in a file prior to deletion from the node, use the `unicpoutu` utility. Use the `unicpinu` utility to copy this information back into a calendar server node.

## Moving calendar users

Due to a variety of potential circumstances — organizational changes, employee relocation, or the need to redistribute node capacity — you may need to move one or more users from one node to another.

### Cmd line

Use the `unimvuser` utility. It is advisable to run `unimvuser` during off-peak hours for the calendar server. Always use the most recent version of `unimvuser` in your node network. For full information on use and syntax, including a number of crucial warnings and considerations, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

# Managing user defaults

To set client display preferences, administrative rights, default viewing privileges or other parameters for a group of users, define a default user profile before adding users to the node. This default user profile may also be applied to existing users.

### Defining a default user profile:

- All configuration parameters for the user profile are stored in the `$ORACLE_HOME/ocal/misc/user.ini` file. Edit this file using a text editor supplied with your operating system.
- Values can be set and changed according to the information and limits defined in Appendix A, "User and Resource Parameters" of the *Oracle Calendar Reference Manual*.
- To make changes, delete the old value and insert a new value.
- The default value is assumed if the parameter is not included in the `$ORACLE_HOME/ocal/misc/user.ini` file.

### Applying a default user profile:

- The profile is applied during user creation (using the `uniuser` utility, or the Calendar Administrator).
- The default user profile is outlined under the section heading `[GEN]` in the `$ORACLE_HOME/ocal/misc/user.ini` file. Multiple profiles can be created from this template and appended to the file under different section heading names. These profiles can then be specified during user creation or modification

using the `uniuser` utility with the **-s** option. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# Setting up e-mail notification

The e-mail addresses allow users to notify each other of created, modified or deleted calendar entries and are used for sending e-mail reminders.

When you add users with `uniuser`, you may specify their e-mail addresses by using the `EMAIL` key / value pair. For example:

```
% uniuser -add "S=Kafka/G=Franz/EMAIL=fkafka@mail.org" -n 23
Enter SysOp password:
uniuser: added "Kafka, Franz"
```

For full information on the use and syntax of `uniuser`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# Granting access rights from one user to another

An administrator can grant a user the right to access the agenda of another user, resource, or event calendar. These rights include event viewing rights, task viewing rights, scheduling rights and designate rights. A designate is a user assigned the right to modify the agenda of another user or resource. Granting scheduling rights to a user means giving him the right to invite you.

You can set access rights using the Calendar Administrator or the `uniaccessrights` utility.

### Web GUI

Use the Calendar Administrator to grant access rights from one user (grantor) to another (grantee). Click the **Calendar Management** tab and then on **Users** at the top left. Search for the user who will be the grantor. Click **Go** to list all users, or select a filter and enter a value in the Search edit box to limit the search then click **Go.** Find the user in the list and click the pencil icon in the **Actions** column to open the profile for this calendar user. Click **Access Rights** on the left.

In the Access Rights page, any user to whom you already have granted rights can be displayed. Click **Go** to list them all. To grant rights to a user for the first time, click **Grant Rights** on the right. Search for the user who will be the grantee and click the Grant Rights icon in the Actions column to change the rights granted to this user. In the "Access Rights to Calendar" page, change the rights and click **Apply**.

### Cmd line

Use the uniaccessrights utility with the -mod option to grant access rights from one user to another or from one user to many users. For example:

```
% uniaccessrights -mod -grantee "S=OBrian" -grantor "S=Martin/G=Don" -host
gravel -p sysop1 -eventview "PERSONAL=ALL" -taskview "all=true"
```

For full information on the use and syntax of uniaccessrights, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# Other user configuration options

## Global and published calendars

Calendar users can share their agenda with other users through the Oracle Calendar Web client. Calendar sharing is determined through two user attributes: Global Read Access and Published Type.

Global Read Access only applies to users of Oracle's Web clients. Users with this attribute turned ON can share their agendas with any other Internet user by mailing them a URL defined by the Web client. For more information on this feature, see your Web client documentation and on-line help.

The Published Type attribute defines whether other calendar users can view this user's or resource's agenda directly using their calendar clients. PUBLISHED calendars can be viewed through any native and Web client; users with this attribute set to NOTPUBLISHED cannot be opened by any other user.

For event calendars, this attribute is set to EVENTCALENDAR and cannot be changed. Event calendars appear in a different list in Web clients. For more information on event calendars, see Chapter 10, "Event Calendars"

The default setting of this attribute for users is NOTPUBLISHED.

Set the Published Type and Global Read Access attributes using the Calendar Administrator or the uniuser utility.

# 9

# Calendar Resources

This chapter describes the various tasks involved in creating and managing resources. The administrative controls and functionality of resource management are similar to those of user management, with some important differences in an LDAP context.

The following topics are dealt with in this chapter:

- Creating calendar resources
- Managing calendar resources
- Deleting calendar resources
- Managing calendar resource defaults
- Granting designate and other access rights to users
- Booking calendar resources
- Searching for calendar resources

## Creating calendar resources

A resource is an inanimate object, such as a conference room or a piece of equipment, that has its own account on the calendar server. When creating an event in their agenda, users can reserve resources by inviting them in the same way that they invite other users. Resources can be managed by local users who act as designates. Resources can be set up to permit reservations on a first come first served basis to prevent double-bookings, to permit more than one reservation at a time, or to require approval by a resource manager.

Resource accounts can also be used to create calendars for tracking related enterprise-wide information, such as company holidays or employees' travel

schedules. For example, to create a Travel Planner for an organization, add a resource to the node and name it “Travel Planner”. Whenever an employee is scheduled to travel, he/she will create an event in his/her personal agenda and invite the resource. The result is a calendar for the resource Travel Planner containing all entries related to employees' travel.

### Web GUI

You can add resources to a node using the Calendar Administrator. Click the **Calendar Management** tab and then on **Resources** at the top left. Click **Add Resource** on the far right. Enter the resource information and then click **Apply**.

### Cmd line

**To add a resource:** Use `uniuser -add -resource` to add a single resource. A password for the resource must be supplied. For full information on use and syntax, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

```
% uniuser -add "R=conference room/N=4/PSW=abcd123" -resource -n 786 -p syspsw
uniuser: added "conference room"
```

**To add several resources:**

1. Create a file listing the resources that you wish to add to a node. The information for each resource must be entered following the format and syntax detailed in the `uniuser` documentation in Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

   ```
   A R=conference room/N=104/CA=24/S=Alexander/G=James/PSW=abc1
   A R=projector/N=2/S=Addison/G=Thomas/PHONE=123-4567/PSW=def2
   ```

2. Add the resources in the file (named “res1” in the following example) to the specified node.

   ```
   % uniuser -ex res1 -resource -n 444
   Enter SysOp password:
   uniuser: added "conference room"
   uniuser: added "projector"
   uniuser: added "lab"
   ```

## Adding resources in an LDAP directory context

In a standalone installation of the calendar server, using an third-party LDAP directory server, you may specify a Resource Relative DN for your installation using

the server parameter `[LDAP] resourcerelativedn`. If you do, all resources will be stored by default in that location in the LDAP tree. Alternatively, you may specify a full DN for resources as you create them. For more details on the `resourcerelativedn` parameter, see Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*.

While users are typically in the directory server before they are added to a calendar server node, resources are added directly to the calendar server and directory server in a single operation.

### Web GUI

You can add resources to a calendar server node easily using the Calendar Administrator. Click the **Calendar Management** tab and then on **Resources** at the top left. Click **Create Resource** on the far right. Enter the resource attributes and then click **Apply**.

### Cmd line

**To add a resource:** Use `uniuser -add` to add a single resource. For full information on use and syntax, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*. This example specifies a full DN for the new resource.

```
%  uniuser -add -resource "DID=cn=Room614,o=Acme,c=US" -n 134
Enter SysOp password:
uniuser: added "cn=Room614,o=Acme,c=US"
```

This example specifies only a resource name, leaving the resource's location in the LDAP directory to be determined by the Resource Relative DN and the calendar server base DN.

```
%  uniuser -add -resource "R=Room614" -n 134
Enter SysOp password:
uniuser: added "R=Room614"
```

#### To add several resources:

1. Create a file of the resources that you wish to add. The information for each resource must be entered following the format and syntax documented in the `uniuser` documentation in Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*. This example specifies full DNs for the new resources.

   ```
   A DID=cn=Room24, o=Acme, c=US
   A DID=cn=projector3, o=Acme, c=US
   A DID=cn=lab5, o=Acme, c=US
   ```

2. Add the resources in the file (named “res1” in the following example) to the specified node.

```
% uniuser -ex -resource res1 -n 444
Enter SysOp password:
uniuser: added "cn=Room24, o=Acme, c=US"
uniuser: added "cn=projector3, o=Acme, c=US"
uniuser: added "cn=lab5, o=Acme, c=US"
```

The resource now exists in the directory server and on the calendar server node.

# Managing calendar resources

### Web GUI

You can manage resources easily using the Calendar Administrator. Click the **Calendar Management** tab and then **Resources** at the top left. Search for the resource you want to modify. You can click **Go** to list all resources or select a filter and enter a value in the Search edit box to limit the search. When the resource is listed, click the pencil icon in the Actions column to modify the resource’s attributes or click the key icon to change the password.

### Cmd line

List resources using the `uniuser` utility with the `-ls` and the `-resource` options. List resource attributes using the `-info` and the `-resource` options. To modify resources, use `uniuser -mod -resource`. For full information on use and syntax, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*. For example:

```
% uniuser -resource -ls "R=HPLaser*" -format "%r% Contact: %g% %s%" -n 12
 -p mypasswd
% uniuser -resource -mod "R=oakroom" -m "N=301/PSW=abc123" -n 23
```

# Deleting calendar resources

When a resource is deleted from a node, the resource's directory entry and records are removed from the local node. The resource will no longer appear as invited in user's agendas, nor will it appear in any directory listings.

Calendar entries owned by the resource can be transferred to another resource before deleting the resource, using `uniuser` with the `-transfer` option.

Another option is to take a copy of the resource's agenda using the `unicpoutr` utility, and subsequently copy it back to any calendar server node using the `unicpinr` utility. For full information on use and syntax of these utilities, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

### Web GUI

You can delete resources easily using the Calendar Administrator. Click the **Calendar Management** tab and then on **Resources** at the top left. Search for the resources you want to delete using the search edit box or select **Advanced Search**. Select the resources you want to delete from the list using the checkboxes in the Select column and click **Delete** on the top right.

### Cmd line

Remove the resource(s) from the calendar server node using the `uniuser -del -resource` (single deletion) or `uniuser -ex` (multiple deletions) commands. For full information on use and syntax, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

## Managing calendar resource defaults

You may define a default resource profile to apply as you add resources to the database. This profile can also be applied to existing resources.

### Defining a default resource profile:

- All configuration parameters for the resource profile are stored in the `$ORACLE_HOME/ocal/misc/resource.ini` file. Edit this file using a text editor supplied with your operating system.
- Default values can be changed according to the information and limits defined in Appendix A, “User and Resource Parameters” of the *Oracle Calendar Reference Manual*.
- To make changes, delete the old value and insert a new value.
- The default value is assumed if the parameter is not included in the `$ORACLE_HOME/ocal/misc/resource.ini` file.

### Applying a default resource profile:

- The resource profile is applied during resource creation (using either the `uniuser` utility or the Calendar Administrator).

- The default resource profile is outlined under the section heading `[GEN]` in the `$ORACLE_HOME/ocal/misc/resource.ini` file. Multiple profiles can be created from this template and appended to the file under different section heading names. These profiles can then be specified during resource creation or modification using the `uniuser` command. For full information on use and syntax, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.
- The default profile of one or more resources can be modified using the `uniuser -ex -resource` command.

# Granting designate and other access rights to users

An administrator can grant a user the right to access the agenda of a resource. These rights include event viewing rights, task viewing rights, scheduling rights and designate rights.

When a designate user is assigned to a resource, he can open the resource's agenda to add, delete or modify entries. Designate and other access rights are set according to the default resource profile file (resource.ini) at the time the resource is added to the node.

If a resource grants scheduling rights to a user, it means the user has the right to book the resource. If the default access rights for all users is set such that no one has the right to book a resource, the resource is said to be restricted.

You can set access rights using the Calendar Administrator or the `uniaccessrights` utility.

### Web GUI

Use the Calendar Administrator to grant access rights from a resource to a user. Click the **Calendar Management** tab and then on **Resources** on the top left. Search for the resource and then click the Update (pencil) icon in the **Actions** column. Click **Access Rights** on the left side. Click **Grant Rights** on the top right. Search the user or users who will be granted the access rights (grantees). Select the users using the checkmark boxes in the Select column then click **Grant Rights** on the right.

### Cmd line

Use the `uniaccessrights` utility with the **-mod** option to grant access rights from a resource to a user or from a resource to many users. For example:

```
% uniaccessrights -mod -grantee "S=OBrian" -grantor "R=ConfRoom1" -host gravel
-p sysop1 -eventview "NORMAL=ALL"
```

For full information on the use and syntax of `uniaccessrights`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# Booking calendar resources

Resource reservation settings are controlled at the level of individual resources. By default, a resource is set to be reserved on a *first come first served* basis. The first user to reserve the resource will get the accepted reply automatically. Any subsequent request to reserve the resource for the same time slot will be refused.

A resource can also be set to *allow double-booking*. When a resource is allowed to be booked by more than one user for the same time slot, the resource designate can choose which user gets the resource by accepting one of the requests and declining the others.

Some resources may be set up such that anyone requesting a resource will need *approval* by a resource manager before the resource can be booked. This mechanism includes sending an e-mail to the approver which notifies him of the request for the resource. The e-mail message will include a link to the calendar Web client where the approver can either accept or decline the request. This e-mail will be sent in the language set for the resource.

When a resource is *restricted*, no one can reserve the resource. To allow a few users to reserve a restricted resource, the right to book the resource must be granted from this resource to each of these specific users.

You can set resource scheduling attributes easily using the Calendar Administrator or the `uniuser` utility. For more information on the use and syntax of `uniuser`, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

## Setting the resource to allow double-booking

Setting the `ALLOW-CONFLICT` attribute to `YES` will allow the resource to be booked by more than one user for the same time slot. Set the `ALLOW-CONFLICT` attribute back to `NO` to let the resource be booked on a first come first served basis. Use `DEFAULT` to resort to the default set by the server parameter (unison.ini) `[ENG] allowresourceconflict`. For more information on this server parameter, see Appendix C, “Server Parameters” of the *Oracle Calendar Reference Manual*.

### Web GUI

A resource can be set to *allow double-booking* using the Calendar Administrator. Click the **Calendar Management** tab and then click **Resources** at the top left. Search for

the resource you want to modify by selecting a Search filter and entering a value in the search edit box. Click **Go** to list the resources. When the resources are listed, click the pencil icon in the Actions column. Scroll down to the Scheduling Options section. Select "Yes -- Requires Approval" in the Allow Double-booking pull-down list.

### Cmd line

To allow double-booking for a resource, set the resource ALLOW-CONFLICT attribute to YES using the uniuser utility with the **-mod** option. Example:

```
% uniuser -n 1 -resource -mod "R=Pool Table" -m ALLOW-CONFLICT=YES
```

## Setting up the approval mechanism for a resource

The Approval mechanism for resources can be set up using the Calendar Administrator or using the uniuser utility.

### Web GUI

To set resource approval using the Calendar Administrator, sign-in and click the **Calendar Management** tab and then **Resources** at the top left. Search for the resource you want to modify and then click the pencil icon in the Actions column. Scroll down to the Scheduling Options section. Select "Yes -- Requires Approval" in the Allow Double-booking pull-down list. For the approval mechanism to work, you must checkmark the "Notify Approver by E-mail" box and enter the e-mail address of the approver in the box labeled "Approver E-mail". Change the language that will be used for sending the e-mail using the pull down list.

### Cmd line

To set resource approval using the uniuser utility, use the **-mod** option. Three attributes must be set, ALLOW-CONFLICT must be set to YES, NOTIFY-APPROVER must be set to TRUE and APPROVER-EMAIL must be set to the approver's e-mail address. Example:

```
% uniuser –n 1 –resource –mod "R=Projector" -m
"ALLOW-CONFLICT=YES/NOTIFY-APPROVER=TRUE/APPROVER-EMAIL=resmanager@test.com"
```

## Restricting a resource

Restricting a resource is done using the resource's CanBookMe attribute. Setting it to FALSE will make the resource restricted. To allow a user to reserve a restricted resource, the CanBookMe access right for the resource must be granted to the user.

### Web GUI

You can restrict a resource easily using the Calendar Administrator. Click the **Calendar Management** tab and then on **Resources** at the top left. Search for the resource you want to modify and then click the pencil icon in the Actions column. Click **Access Rights** on the left menu. Click the **Modify Default Access Rights** button on the top right. In the Scheduling section, un-check the "Can Invite Me" option. This will be reflected in the Scheduling Options section of the Resource Information Web page for this resource where it will say "Restricted resource".

To grant booking rights to specific users, from the same **Resources** page, search for the resource then click the pencil icon in the Actions column. Click **Access Rights** on the left menu, then click **Grant Rights** on the top right. Find a user or all the users using the search box. When the user or the list of users is displayed, checkmark the user or users to whom you want to grant access rights in the Select column, then click **Grant Rights** at the top right of the list. Go down to the Scheduling section and select "Can Invite Me". Click **Apply**.

### Cmd line

Use the `uniuser` utility with the **-s** option to set a resource to be restricted. This option lets you specify a section of the configuration file `$ORACLE_HOME/ocal/misc/resource.ini` to use for determining the default values to be used for editing the attributes of a resource.

For full information on the use and syntax of `uniuser -s`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*. For example:

1. Create a new section `[RESTRICT_RESOURCE]` in the `resource.ini` file with the `CanBookMe` parameter set to `FALSE`. Simply add these two lines at the end of the file:

```
[RESTRICT_RESOURCE]

CanBookMe=FALSE
```

You can list the values in any section of the `resource.ini` file:

```
% uniuser -resource -defaultls -s "RESTRICT_RESOURCE" -n 23 -p sysop1
CanBookMe = FALSE
```

2. Proceed with the modification for the OakRoom resource:

```
% uniuser -resource -mod "R=OakRoom" -s "RESTRICT_RESOURCE" -n 23 -p sysop1
uniuser: "R=OakRoom/N=100/UID=oakroom/ID=308/NODE-ID=23" has been modified.
```

3. Then use `uniaccessrights` utility with the **-mod** option to grant access rights from the resource to a user. For example:

```
% uniaccessrights -mod -grantee "S=OBrian" -grantor "R=OakRoom" -p sysop1
-scheduling "canbookme=true" -n 23
```

4. Or to many users. For example, to all the people in Engineering:

```
% uniaccessrights -mod -grantee "OU1=Engineering" -grantor "R=OakRoom" -p sysop1
-scheduling "canbookme=true" -n 23
```

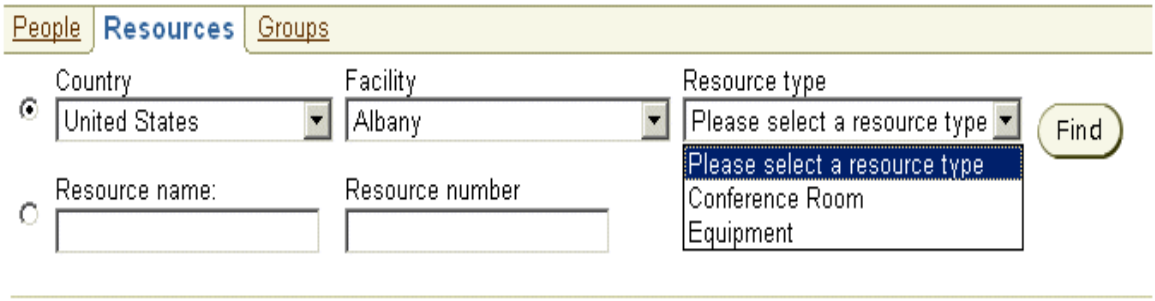
For full information on the use and syntax of `uniaccessrights`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# Searching for calendar resources

The calendar server facilitates the task of searching for resources through the use of resource categories and resource capacity for calendar clients that support it. When a new resource such as a conference room or a vehicle is added, a capacity value can be entered. The capacity can be used by the calendar users to select resources. The Calendar Administrator interface and the Oracle Connector For Outlook both allow users to filter resources based on the capacity.

When a user searches for resources using any calendar client, he can narrow his search by first selecting the country, then the facility and finally the type of resource (rooms, equipment, etc.). Figure 9–1 shows how a user can search for a resource from a calendar client by selecting entries from three pull down lists. Resource categories are used to store this information in the calendar server database.

***Figure 9–1 Searching for a resource in the Oracle Calendar Web client***



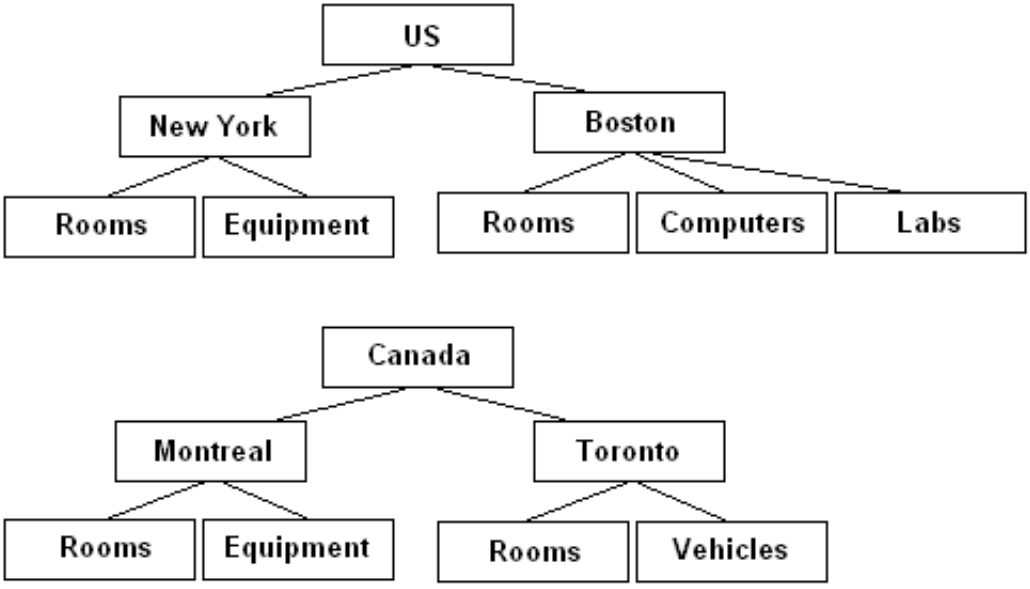
## Defining resource categories

In order to provide this feature for end users, the list of resource categories must be defined in the category configuration file `$ORACLE_HOME/ocal/misc/category.ini`. A second category configuration file `$ORACLE_HOME/ocal/misc/categorytype.ini` exists in the same directory and should not be modified. The list of available countries, facilities and resource types will be displayed in the calendar clients according to the `category.ini` file. The administrator must edit this file and enter the data manually. These two files are needed to allow searching for resources using resource categories.

The categories must be organized in a tree structure which specifies the countries, the facilities in each country, and the types of resources available at each facility.

Figure 9–2 shows an example of resources in an organization with offices in 2 countries, in 4 different cities, and with different types of resources.

***Figure 9–2 Resource organization tree***




The file `category.ini` contains information describing each node of the resource tree and defines the names of the categories that will appear in the pull-down lists

in the user interface (for example, the country “United-States”, the facility “New-York”, the resource type “Computer”). Each category name can be defined in different languages allowing localized calendar clients to display the list of categories in the proper language.

These files must be encoded in the UTF-8 format. To convert a string to UTF-8, use the `unistrconv` utility or use a UTF-8 editor. For full information on use and syntax of this utility, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

Once you have defined the way to structure the resources in your organization, edit the `category.ini` file to enter the information. You may want to keep a copy of the sample `category.ini` file before changing it.

In order for the categories to apply to all resources in a node network, the `category.ini` file must be copied on all hosts in the network.

### Structure of the category.ini file:

The `category.ini` file contains the list of all countries, the list of all facilities for each country and the list of all types of resources available at each facility. Every entry in the three lists must have a corresponding section in the file. Each section is identified by a section name enclosed in square brackets, (ex. `[RL_NEWYORK]`). Section names must not exceed 32 characters.

Each section contains a list of labels to be displayed in the different languages available to end users. Each of these label strings must be encoded in UTF-8 format. The `typeid` parameter in each section specifies whether the entry is part of the list of countries, of facilities or of resource types. All section names must be different.

#### Country sections

Enter sections for each country. For example, if your organization has offices in three countries, USA, England and Canada, then you should enter three sections, one for each country: `[RL_USA]`, `[RL_ENGLAND]` and `[RL_CANADA]`. Within these sections, the `typeid` parameter should be set to “`RL_COUNTRY`“and `parentid` to “`RL_ROOT`“. For example:

```
[RL_USA]
typeid = "RL_COUNTRY"
parentid = "RL_ROOT"
description.en = "USA"
```

Use the `description.xx` parameters to enter the country’s label in more than one language. The translated labels will be used for calendar clients of different

languages. For example, `description.fr` defines the French label, `description.de` defines the German label, etc. Each of these label strings must be encoded in UTF-8 format.

For example:

```
[RL_ENGLAND]
typeid = "RL_COUNTRY"
parentid = "RL_ROOT"
description.en = "England"
description.fr = "Angleterre"
description.es = "Inglaterra"
```

The language codes are the following:

| | |
|---|---|
| `en` | (English) |
| `fr` | (French) |
| `it` | (Italian) |
| `es` | (Spanish) |
| `de` | (German) |
| `pt` | (Portuguese) |
| `ja` | (Japanese) |
| `ko` | (Korean) |
| `zh-cn` | (Traditional Chinese) |
| `zh-tw` | (Simplified Chinese) |
| `pt-br` | (Brazilian Portuguese) |

### Facility sections

Enter sections for each facility. For example, if your organization has two offices in the USA, then you should enter two sections with the `typeid` parameter set to "`RL_FACILITY`" and `parentid` to "`RL_USA`". The section name should identify the facility. The name of the city it is located in or the name of the department that occupies it can be used. For example:

```
[RL_NEW_YORK]
typeid = "RL_FACILITY"
parentid = "RL_USA"
description.en = "New York Office"
```

```
[RL_SANFRANCISCO]
typeid = "RL_FACILITY"
parentid = "RL_USA"
description.en = "San Francisco Laboratory"
```

#### Resource type sections

Enter sections for each type of resource available at a facility. For example, if your office in New York has three types of resources (Conference rooms, Equipment, Vehicles), then you should enter three sections with the `typeid` parameter set to "`RL_RESOURCETYPE`" and `parentid` set to "`RL_NEW_YORK`". The section name should identify the resource type and the facility and should be unique. For example:

```
[RL_NY_VEHICLES]
typeid = "RL_RESOURCETYPE"
parentid = "RL_NEW_YORK"
description.en = "Vehicle"
description.es = "vehículo"

[RL_NY_CONFROOM]
typeid = "RL_RESOURCETYPE"
parentid = "RL_NEW_YORK"
description.en = "Conference room"
description.es = "Salón de Conferencias"

[RL_NY_EQUIPMENT]
typeid = "RL_RESOURCETYPE"
parentid = "RL_NEW_YORK"
description.en = "Equipment"
description.es = "Equipos"
```

For each facility, in all countries, resource types must be entered this way with a section name reflecting the resource type and the facility, and the `parentid` parameter set to the facility's section name.

## Assigning a category to a resource

Once all resource types have been entered in the `category.ini` file, each resource entered in the database can then be associated with a resource type category. Associating a category to a resource will ensure that this resource is found when the user selects the right category type from the pull-down list.

### Cmd line

To associate a resource to the proper categories, the `uniuser` utility can be used with the **-mod** option. For example:

```
% uniuser -resource -n 1 -mod "R=ConfRoom202" -m "CATEGORY=RL_
NY_CONFROOM"
```

Once categories have been assigned to all resources, you can list the resources based on categories using `uniuser -resource -ls`. For example:

```
% uniuser -resource -n 1 -ls "CATEGORY=RL_NY_CONFROOM"
```

For full information on use and syntax on this utility, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

### Web GUI

You can set categories for a resource easily using the Calendar Administrator. Click the **Calendar Management** tab and then on **Resources** at the top left. Find the resource or resources you wish to edit by using the Search box. Once the resource is listed, click the pencil icon to edit the resource’s attributes. Select **Categories** on the left hand side. The Available Categories will be listed. Click a category name to see the sub-categories. Click the check boxes to select categories to associate to the resource. Click View All to view all categories available.

# 10

# Event Calendars

This chapter describes the various tasks involved in creating and managing event calendars. Event calendars are similar to user accounts, but represent schedules such as sports matches, concerts, or other events that may be of interest to your user base. The administrative controls and functionality of resource management are also similar to those of user management. An event calendar can be accessed by signing in with the account password using one of the Oracle Calendar Desktop clients.

The following topics are dealt with in this chapter:

- Creating event calendars
- Managing and deleting event calendars
- Populating event calendars
- Granting designate rights to users

## Creating event calendars

An event calendar is an administrative calendar account which exists to inform your user base about upcoming events that may be of interest to them. Users have read-only access to the events created in event calendars. However, using the Oracle Calendar Web client, users may copy events and appointments from the event calendar into their own calendars. Administrators or users who have been given special administrative rights may create event calendars. See Chapter 11, "Administrative Rights" for more details on how to give calendar users the right to manage event calendars.

### Web GUI

Use the Calendar Administrator to create event calendars easily. Click the **Calendar Management** tab and then on **Event Calendars** at the top left. Click **Add Event Calendar** on the far right.

### Cmd line

**To add an event calendar:** Use `uniuser -add -eventcal` to add a single event calendar. A password for the event calendar must be supplied. For full information on use and syntax of `uniuser`, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

```
% uniuser -add "N=Training classes/PSW=abcd12" -eventcal -n 786 -p syspsw
uniuser: added "Training classes"
```

**To add several event calendars:**

1. Create a file listing the event calendars that you wish to add to a node. The information for each event calendar must be entered following the format and syntax documented in the `uniuser` documentation in Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

   ```
   A N=Training Classes
   A N=Soccer Matches
   ```

2. Add the event calendars in the file (named "evcal1" in the following example) to the specified node.

   ```
   % uniuser -ex evcal1 -eventcal -n 444
   Enter SysOp password:
   uniuser: added "Training Classes"
   uniuser: added "Soccer Matches"
   ```

## Other directory servers

If your calendar configuration consists of a standalone installation using a 3rd party directory server, a user account must first be created for this event calendar in the directory server. Once it exists in the directory, you can create a calendar account for the event calendar. In this configuration, the existing directory data is used to create a calendar profile for the event calendar. Providing a password is not mandatory. The Calendar Administrator will list all entries existing in the directory server. From this list, the event calendar is found and calendar services can be added to the account.

# Managing and deleting event calendars

### Web GUI

You can manage event calendars easily using the Calendar Administrator. Click the **Calendar Management** tab and then on **Event Calendars** at the top left. Search for the event calendar you want to modify and then click one of the icons in the Actions column.

### Cmd line

View event calendar attributes using the `uniuser` utility with the **-ls** and **-eventcal** options. To modify users, use `uniuser -mod -eventcal`. For full information on use and syntax, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

# Populating event calendars

Only administrators or users with designate rights may modify events in an event calendar. Users who have been given special administrative rights can also manage the agendas of event calendars. Unlike calendar users or resources, Event Calendars cannot be invited to meetings.

### Web GUI

As an administrator, use the Calendar Administrator to populate event calendars easily. Click the **Calendar Management** tab and then on **Event Calendars** at the top left. Search for the event calendar you want to populate and then click the Manage Events icon in the Actions column. You can then enter or modify meetings, day event or notes in the event calendar.

### Calendar clients

To add new events to an event calendar, you may also sign in as that event calendar using one of the Oracle Calendar Desktop clients and change the content of the agenda. Remember to grant other users the right to view the events you create if you choose this method of managing events in your event calendars. Do this either by setting each event’s access level to "Public" when you create it, or by setting the event calendar’s default access right profile to allow all users to view "Normal" events.

Users with designate access using the Oracle Calendar Desktop clients or the Oracle Connector for Outlook, can sign-in to their own calendar account and then manage

the contents of an event calendar by working as designate. See *Granting designate rights to users* later in this chapter.

### Cmd line

Use the `uniical` utility to add entries to an event calendar. For full information on the use and syntax of `uniical`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

## Granting designate rights to users

An administrator can grant to a user designate access to an event calendar. When a designate user is assigned to an event calendar, he can open the event calendar's agenda to add, delete or modify entries. Designate rights can be set according to the default event calendar profile file (eventcal.ini) at the time the event calendar is added to the node.

You can set designate access rights using the Calendar Administrator or the `uniaccessrights` utility.

### Web GUI

Use the Calendar Administrator to grant access rights from an event calendar to a user. Click the **Calendar Management** tab and then on **Event Calendars** at the top left. Search for the event calendar and then click the pencil icon in the **Actions** column. Click **Access Rights** on the left side and then on **Grant Rights** on the far right.

### Cmd line

Use the `uniaccessrights` utility with the **-mod** option to grant access rights from an event calendar to a user or from an event calendar to many users. For example:

```
% uniaccessrights -mod -grantee "S=OBrian/OU1=teachers" -grantor "N=Training
Classes" -host gravel -p sysop1 -designate "ALL=TRUE"
```

For full information on the use and syntax of `uniaccessrights`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# 11

# Administrative Rights

Oracle Calendar offers a wide range of administrative rights that can be assigned to calendar users, covering calendar management (users, resources, event calendars, holidays, etc.) and server administration (starting and stopping nodes and servers, configuring initialization files, etc.).

This chapter contains an overview of the rights that can be assigned to users, and the methods available for assigning those rights.



## Administrative rights

Most of the administrative operations for Oracle Calendar can be performed by regular calendar users. The administrator (calendar system operator) must first grant the users administration rights. Different sets of rights can be granted to different users based on what they should manage: users, resources, event calendars, groups, node network, or calendar server.

These administrative rights limit the operations any given user can perform using the Calendar Administrator and command-line calendar utilities.

For example, it may be useful to give designated employees in Human Resources administrative rights for holidays. Those employees could then add, modify and delete holidays in the Calendar Administrator by signing-in with their own user name and password.

For a complete list of the individual rights that can be assigned in Oracle Calendar 9.0.4, see the `uniadmrights` utility in Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# Scope of administration

Each user's profile of administrative rights also has a *scope* that defines which nodes the user's rights apply to. The **Node** scope limits all administrative rights to the node on which the user's own calendar account exists. The **Network** scope extends the user's administrative rights to all nodes that are in the network which includes that user's node.

For example, in a scenario where nodes 30 and 40 are in a node network together, a user on node 30 with a Node scope may only modify users and resources on node 30. A user on node 40 with Network scope may modify users and resources on both nodes.

# Assigning rights to users

You can assign administrative rights to a user through the Calendar Administrator or the command-line `uniadmrights` utility.

Note that the ability to manage other users' administrative rights is itself covered by an administrative right. In addition, you may only assign rights that you possess, and you may only assign Network scope if you possess Network scope. For example, in order to assign resource creation administration rights to a user, you must possess *both* resource creation administration rights *and* the right to manage users' administrative rights.

### Web GUI

Use the Calendar Administrator to grant administrative rights to users. Click the **Calendar Management** tab and then on **Users** at the top left. Search for the user you wish to grant administrative rights to and then click the pencil icon in the Actions column. Click **Administrative Rights** in the menu on the left.

### Cmd Line

Use the `uniadmrights` utility. For example:

```
% uniadmrights -u "S=Heller/G=Joseph" -n 22 -user "create=true/modify=true"
-resource "all=true" -csm "all=false"
```

The preceding command line grants Joseph Heller on node 22 the ability to create and modify users (but not to delete, set access rights or passwords); all rights for resources (including the rights to create, modify, delete, set passwords, and set access, viewing and designate rights); and removes all access rights to the Calendar Server Manager, denying this user the ability to start and stop nodes and servers.

For more details on the `uniadmrights` utility, including syntax, accepted key-value pairs and a complete list of all available access rights, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual.*

# 12

# Groups

This chapter describes the different tasks involved in managing groups and group members within a calendar server node network.

The following topics are dealt with in this chapter:



## Managing groups

Calendar groups of users and resources can be created by an administrator to be used by everyone in the calendar network. Private groups can be created by any user for their personal usage. Calendar groups allow users to schedule meetings efficiently with other users and resources.

Groups can include members on any remote node. There are four types of groups:

### Private groups

- available only to the users who created the groups
- created and modified by users in the Group Management dialogue box of a calendar client
- the right to create these groups is available to all calendar users

### Members-only groups

- available only to members of the group
- created and modified by users in the Group Management dialogue box of a calendar client

- the right to create these groups is available to all users

### Public groups

- available to all users
- the user will have the right to create, modify and delete his own public groups
- ownership of these groups is exclusive and cannot be transferred
- can be created and modified only by users who have been granted the administrative rights to do so

### Administrative groups

- available to all users
- administrative groups are owned by the SYSOP, not the users who create them
- can be created and modified only by users who have been granted the administrative rights to do so

Private, public and members-only groups are managed by the users who created them. Users can create or modify these groups using their Calendar client. Administrative groups can be managed by administrators using the group utility `unigroup` or the Calendar Administrator client.

The server administrator can give special group administration rights to calendar users. For full information on how to grant users group management rights, see Chapter 11, "Administrative Rights".

### Web GUI

Use the Calendar Administrator to create groups easily. Click the **Calendar Management** tab and then on **Groups**. To add a new group click **Add Group** on the far right. To modify an existing group or to add members to it, search for the group you want to modify using the search box. When the group is listed click the corresponding pencil icon in the **Actions** column. Click **Users** or **Resources**. The list of existing members in the group will be listed. To delete a user or resources, click the **Remove User** or **Remove Resource** icon in the **Actions** column. To remove more than on member, check mark each one in the **Select** column and then click **Delete Users** or **Delete Resources** button at the bottom right. To add new members to the group, click **Add Users** or **Add Resources** on the far right.

### Cmd line

**To add a group:** Use `unigroup` with the `-add` option to create a new group. Use the `-mod` option to modify the group name or other attributes. Use the `-attach` option to add members to the group. Use the `-ls` and `-members` utility can be used to list public and administrative groups and their members. For full information on use and syntax of `unigroup`, see Appendix E, “Utilities” of your calendar server *Reference Manual*.

```
% unigroup -add "NAME=Marketing" -n 8 -p sesame
unigroup: NAME=Marketing/ID=4096/NODE-ID=8/TYPE=admin
```

## Directory server groups and group filters

Oracle calendar clients allow users to view and create groups of calendar users. If any groups exist in the LDAP directory server, these groups will also be available for viewing (but not modification) in the calendar client. The Oracle server also allows users to view Oracle Mail distribution lists as groups. This includes any mailing lists present in Oracle Internet Directory. Only the Oracle Connector For Outlook gives access to these groups for sending mail or inviting users to meetings.

For more info on directory groups and how to configure them, see "LDAP groups and distribution lists", on page  on page 4-3 of Chapter 4, "Directory Servers for Calendar".

# 13

# Holidays

Holidays are special events that appear in the calendars of all users, resources and event calendars in a node. Holidays can be created one-by-one by any user with the proper administrative rights, through the Calendar Administrator or one of the Oracle Calendar Desktop clients. Alternatively, administrators can import holidays for their locales directly into the calendar server using the `uniical` utility.

This chapter describes the various tasks involved in creating and managing holidays.

## Managing Holidays

### Assigning Holiday Administration Rights

The SYSOP, by default, has holiday management rights. The administrator can also assign holiday administration rights to a regular calendar user. This can be done either through the Calendar Administrator or the command line. For more details on granting administration rights to users, see Chapter 11, "Administrative Rights".

#### Web GUI

To assign holiday administrative rights to a given user, go to the **Users** section of the Calendar Administrator and search for the desired user. Select the pencil icon in the **Actions** column for that user. On the next page, click **Administrative Rights** from the menu on the left.Note that this option will only appear if you are signed in to the Calendar Administrator as the SYSOP, or as a user with the right to manage other users' administrative rights. Scroll down to the **Node Management** section where one of the options is Manage Holidays.

### Cmd line

Use the `uniadmrights` utility to assign holiday administration rights. For full information on use and syntax, see the Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

For example:

```
% uniadmrights -u "S=Sitchin/G=Zechariah" -n 165 -node "holiday=true"
```

## Creating and Modifying Holidays

### Oracle Calendar Desktop Client

Sign in as the Sysop or as a user to whom were granted holiday administration rights in the previous step. Select the holiday management menu item (Tools | Manage Holidays...). See your client's online help for details.

### Web GUI

Sign in to the Calendar Administrator as the Sysop or as the user who has holiday administration rights. Select the **Server Administration** tab on the top right. Click **Nodes** on the top right. Search for the node if it is not already listed. Click the node's **Manage Holiday** icon in the **Actions** column. A calendar will be displayed in which you can add, modify or delete holidays. Click the Add Holiday icon at the top to add new holidays. To modify an existing holiday, find it using the navigation links (Previous or Next) on the top right and then click its link in the calendar.

### Cmd line

Use the `uniical` utility to import a file containing any number of holidays in iCalendar format. For details on the `uniical` utility and the required iCalendar format, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

# 14

# Alerts

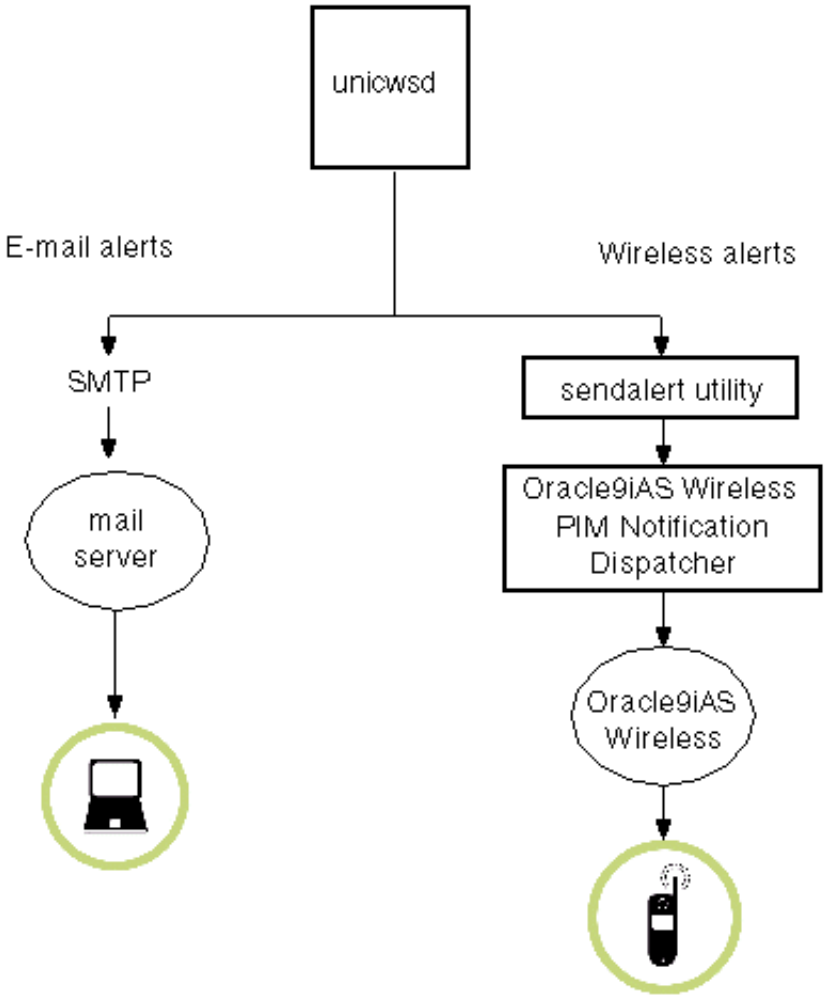
Oracle offers two kinds of alerts: *reminders* and *notifications*. Reminders are messages sent a specified amount of time in advance of a meeting or event, to alert users to the upcoming events in their agendas. Notifications are messages sent when a meeting or event is created, modified or deleted, to let the attendees know of the change to their schedules.

This chapter contains general considerations relating to the server-side implementation of reminders and notifications. See also the documentation for Oracle9*i*AS Wireless, which provides reminders and notifications through a number of channels including voice mail, FAX and Short Message Service (SMS).



The Corporate-Wide Services daemon is responsible for delivering reminders and notifications through e-mail and wireless services. When it has an alert to deliver via e-mail, it sends the alert by SMTP to the mail server specified by the `unison.ini [CWS] smtpmailhost` parameter (if present), which delivers the message to its intended recipient. When it has an alert to deliver via SMS or any other technology supported by Oracle9*i*AS Wireless, the CWS calls the `sendalert` program specified by the `unison.ini [CWS] smsnotifyprogram` parameter (if present), which delivers the message to an instance of Oracle9*i*AS Wireless PIM Notification Dispatcher.

For more information on these server parameters, see Appendix C, "Server Parameters" in the *Oracle Calendar Reference Manual*.


unicwsd
E-mail alerts
Wireless alerts
SMTP
sendalert utility
mail server
Oracle9iAS Wireless PIM Notification Dispatcher
Oracle9iAS Wireless

## Filtering e-mail alerts

The CWS includes MIME headers in all e-mail alerts to allow users to easily filter them, either to specific folders or to another application, such as a pager delivery system. All e-mail alerts include the following MIME header:

```
X-Oracle-Calendar: 1
```

All e-mail reminders include the following MIME header:

```
X-Oracle-Calendar-Reminder: 1
```

# Reminders

## Types

Some Oracle clients offer reminders that pop up or appear in the user's agenda in advance of a meeting; these types of reminders are implemented on the client side. The calendar server is only involved in the case of reminders sent by e-mail or through Oracle9*i*AS Wireless. Server Side Reminders (SSR) are messages used to remind the user of an upcoming meeting. They are usually sent a few minutes before the start of the meeting. There is no special server configuration related to reminders.

## Configuring for users

SSR can be configured for a user. For example, a user can select to receive reminders for normal events but not for daily notes. These options can be set for a user, using the `uniuser` utility. Example:

```
% uniuser -user -mod "S=Kundera/G=Milan" -m
"REMINDER-SERVERALERT=TRUE/REMINDERDAILYNOTE-SERVERALERT=FALSE" -n 23
```

For a complete list of the reminder attributes for users, see Appendix A, "Calendar User and Resource Parameters" of the *Oracle Calendar Reference Manual*. For full information on use and syntax of `uniuser`, see Appendix E, "Utilities" of the *Oracle Calendar Reference Manual*.

Reminder attributes can also be set for all users using the user profile file `user.ini`. For more information on how to use a default user profile, see Managing user defaults in Chapter 8, "Calendar Users".

## Format

The format of the reminder delivered by the calendar server is determined by template files installed in `$ORACLE_HOME/ocal/etc/reminder`, according to the delivery mechanism (e-mail or wireless) and language of the user (if set). These files may be customized, but if you plan to modify a reminder template file, note that the data in these files must be stored in the UTF-8 character set.

To convert a string to UTF-8, use the `unistrconv` utility or use a UTF-8 editor. For full information on use and syntax of this utility, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

## Controlling reminder behaviour

### CWS operation

You can control the frequency with which the Corporate-Wide Services daemon / service checks for reminders using the `unison.ini [NOTIFY] checkreminderinterval` parameter, and the amount of time that the CWS will spend checking any given node for reminders using the `[NOTIFY] limitremindercheck` parameter.

### Old reminders

Reminders will not be sent for any event if the reminder time is set to a time before the current time when that event is created. For example, if a new meeting is created with a start time five minutes from now, and a reminder set to be delivered ten minutes before the start time of that meeting, that reminder will not be sent by the calendar server.

In any other case, if the CWS finds that the scheduled delivery time of a reminder is before the current time, that reminder will still be delivered up to 30 minutes after the intended delivery time. You can customize this value using the `unison.ini [NOTIFY] ignoreoldreminders` parameter. For more information on this server parameter, see Appendix C, "Server Parameters" in the *Oracle Calendar Reference Manual.*

# Notifications

## Types

E-mail and wireless notifications are handled differently.

When a user creates, modifies or deletes an event using an Oracle calendar client, he or she may choose to notify the attendees by e-mail. This e-mail is passed from the client to the server which will queue the request in the CWS for delivery.

Wireless notification, however, is not decided by the user creating, modifying or deleting the event. Instead, users can specify in their calendar clients (if their clients support the feature) whether they wish to be notified via wireless services when meetings or events to which they are invited are created, modified or deleted.

## Configuring for users

Alert notifications can be configured for a user with the `uniuser` utility or through the Calendar Administrator. Alerts can be disabled on a per user basis. Alerts can be suspended for a configurable period of time in the day. Alerts that are sent during this period are either discarded or held until the suspension period ends. Users can also select which type of calendar entries should trigger an alert.

### Web GUI

Use the Calendar Administrator to set alert parameters for a user. Click the **Calendar Management** tab and then on **Users**. Search for the user you want to modify using the search box. When the user is listed click the corresponding pencil icon in the **Actions** column. Click **Alerts** in the menu on the left.

### Cmd line

To set alert parameters for a user, use the `uniuser` utility with the `-mod` option. For example, to get notifications for changes made to meetings:

```
% uniuser -user -mod "S=Kundera/G=Milan" -n 23 -m
"ALERT-NOTIFMEETING=TRUE/ALERT-SUSPENDRANGEACTION=HOLD/ALERT-SUSPENDRANGE=23:00-
06:30"
```

For a complete list of the alert attributes for users, see Appendix A, "Calendar User and Resource Parameters" of the *Oracle Calendar Reference Manual*. For full information on use and syntax of `uniuser`, see Appendix E, “Utilities” of the *Oracle Calendar Reference Manual*.

Alert attributes can also be set for all users using the user profile file `user.ini`. For more information on how to use a default user profile, see Managing user defaults in Chapter 8, "Calendar Users".

## Format

E-mail notifications are passed to the SMTP server with the same text formatting used by the calendar client.

## Controlling notification behaviour

### Disabling e-mail notification in some clients

You can disable e-mail notification entirely in the Oracle Calendar Desktop clients by setting the `unison.ini [LIMITS] mail` parameter to `FALSE`.

### Limiting the number of recipients

If necessary in order to avoid strain on your mail server, you can limit the number of recipients for any given notification message using the `unison.ini [LIMITS] maxmaildistr` parameter.

# Setting up wireless services

When you install the calendar server, you will be asked whether you want to configure wireless services, and if so, you will be prompted for all the necessary information. If you choose not to configure your calendar server for wireless services at installation time, you can use the following manual procedure to set up wireless services afterward.

1. Stop all calendar servers in your network.
2. Edit the `$ORACLE_HOME/ocal/misc/unison.ini` file on each host.
3. Set the following parameter values:

```
[NOTIFY]
alert_sms = TRUE

[CWS]
smsnotifyprogram = sendalert
smsnotifyprogramparam = "-host <hostname> -port <portnumber>"
```

In place of the <hostname> and <portnumber> variables in the preceding `smsnotifyprogramparam` parameter example, supply the host name and port number of an instance of Oracle9*i*AS Wireless PIM Notification Dispatcher.

If you do not know the host name and port number of an instance of Oracle9*i*AS Wireless PIM Notification Dispatcher, you can find out using the

Wireless system management area of Oracle Enterprise Manager. For more details, consult the Oracle9*i*AS Wireless documentation.

4. Restart all calendar servers stopped in step 1.

# 15

# Node Maintenance

A regular schedule of node maintenance is the best protection against unscheduled down time and loss of data. Following the procedures outlined later in this chapter will minimize problems and ensure that your calendar server runs smoothly and without interruption.

This chapter outlines the following tasks:

- Server maintenance procedures
- Server back up and restore
- User back up and restore

## Server maintenance procedures

### Daily monitoring procedures

The following system monitoring procedures should be performed on a daily basis:

- Check that all relevant daemons/services are operational.
- Check that ample space is left in the `$ORACLE_HOME/ocal` directory or file system. For more information on calculating the storage requirements for your node, see Appendix A, "Disk Space and Memory".
- Verify that the previous night's backup has run.
- Search for unusual entries in the log files in the `$ORACLE_HOME/ocal/log` directory. This task can be automated by "grep"ing/searching the log files for specific errors, and e-mailing the results to the calendar server administrator.
- Check for recent writes to the `$ORACLE_HOME/ocal/log/dbv.log`. This file is created only if there is a problem and should be manually removed once the

problem is resolved. If the file is present and is not empty, you might analyze the contents and use the `unidbfix` utility, or consult your support provider for further assistance.

### NT

Windows NT's Performance Monitor tool can be used to chart or log the performance and activity of the calendar services. Windows NT's Event Viewer records any problems encountered running the application.

### Cmd line

The `unistatus` utility displays the current status of the calendar server. The `uniwho` utility can be used to display the list of users currently logged on to the calendar server. Use the `-nolist` if you only want to see the total number of signed-in calendar users. For full information on use and syntax of these utilities, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

### Web GUI

The Calendar Administrator can also be used to view the server status. Click the **Server Administration** tab and then on **Servers**. The server or servers in the same network will be listed. From this page, you can already see which servers are up and which ones are down based on the icons in the **Actions** column. Click the View icon in the **Actions** column for the server you want to view. The Identification section displays whether the server is running and the number of users currently logged on. Other server settings (indicating whether user passwords can be changed, whether the server is connected to a Directory Server, etc.) are also displayed.

## Special monitoring procedures

It is possible to turn on logging of specific calendar activities using server parameters. Most of these options should be turned on for short periods of time as it increases the amount of data written to log files and can cause these files to grow rapidly. Statistical data can be compiled regarding user connections, activity information of the `unicwsd` daemon / service, directory server access, etc.

To view elapsed time and CPU statistics for each client connection, set `[ENG] stats=TRUE` in `unison.ini`. When a client connection is closed, stats results are appended to the `$ORACLE_HOME/ocal/log/stats.log` file. Once the period being analyzed has passed, you must not forget to set the parameter `[ENG]stats` back to `FALSE` to disable logging, as the file grows quickly.

See also parameters [CWS]log_activity and log_modulesinclude, [ENG]stats, activity and dac_failederrlog in Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual*.

## Daily maintenance procedures

A nightly backup of the calendar database ($ORACLE_HOME/ocal/db) and configuration files ($ORACLE_HOME/ocal/misc) is your best protection against database corruption that may occur as a result of a power failure or disk crashes. While database corruption is rare, even under the aforementioned conditions, nightly backups serve as a safeguard in the event that your database cannot be restored. For more information, see "Server back up and restore" later in this chapter.

## Monthly maintenance procedures

The following system maintenance procedures should be done after hours on a monthly basis:

- Archive the log files. Remember to shut down the server before archiving the log files, and to restart the server once the task is completed.
- If you are using a directory server, run unidssync as required to ensure that the information in the node(s) is synchronized with that in the directory server. For full information on use and syntax of the unidssync utility, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.
- To improve performance and minimize disk space requirements, the unirmold utility should be run monthly to remove all events and tasks older than 12-18 months. For full information on use and syntax of unirmold, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

## Other maintenance procedures

- Verify the consistency of the server database(s) using the unidbfix utility.
- The unidbfix utility should be run in check mode once a week with the calendar server running, and in fix mode once a month with the calendar server down. If the weekly check discovers an error, it should be corrected immediately using unidbfix in fix mode; if the weekly check produces a warning, maintenance can be delayed until the monthly fix.

  It is possible to stop one node at a time. This allows you to run unidbfix on a single stopped node while the rest of the nodes are still active. Use the -n

option to specify which nodes to fix. More than one instance of the `unidbfix` utility can be run at the same time on different nodes. For full information on use and syntax of `unidbfix`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

- If you are using a directory server, run `unidsdiff` to detect and resolve any discrepancies in the mapping between users and resources in the directory server with those in the calendar server node. For full information on use and syntax of `unidsdiff`, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*. You should perform this synchronization procedure every 2 to 4 weeks or as required when making a batch of changes to the calendar node, particularly when deleting users. You may also synchronize your calendar and directory servers through the Calendar Administrator.
- The administrator can append text at the end of notification and reminder e-mail messages sent out by a server. See Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual*, for details on how to enable message banners for mail notifications and reminders using the `[CWS]banner` parameter.

# Server back up and restore

To minimize the impact on your users, back up your calendar server only during periods of low user activity. If you use an external directory server, back up your directory server concurrently with your calendar server to minimize inconsistencies should it become necessary to restore a backup.

You have three options for backing up your calendar server:

- the `unidbbackup` utility
- stopping the calendar server and running the `uniarch` utility
- stopping the calendar server and copying or zipping the database files directly

The `unidbbackup` utility is recommended, as it provides on-line or 'hot' backups allowing users to login during a backup. An on-line backup cannot be achieved by simply copying the database files while the server is still running, as the files on disk are not necessarily an accurate reflection of the state of the database at any given time. If you choose to copy the database files directly, you must stop your server to allow all database contents to be written to the disk first.

While `unidbbackup` is running, users can sign-in and sign-out. They may view but not modify their agenda. If more than one node exists on a host, each node is locked and backed up in succession. The `-lockall` option can be used to lock all the

specified nodes at the same time instead of one by one. This will improve the data consistency for connected nodes. The `unidbbackup` utility can be used to make a backup of a single node using the `-n` option.

`unidbrestore` is the complementary utility used for database restoration. For full information on the use and syntax of the `unidbbackup` and `unidbrestore` utilities, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

### To back up a calendar host:

- Execute the `unidbbackup` utility through the command line. A backup will be made of all database and configuration files on your calendar server. If more than one node exists on the host, `unidbbackup` will back up each node in turn. To make backups of specific nodes only use the `-n` option.

### To restore a calendar host:

1. Shut down the server.
2. Run `unidbrestore` to restore the backup. Your calendar database and configuration files will be restored to the `$ORACLE_HOME/ocal` directory on the host.

> **Warning:** This operation restores only the database and configuration files. Calendar data stored in a directory server must be restored separately. If you have any reason to expect that inconsistencies may exist between the data in the calendar server and that in the directory server, use the `unidsdiff` and `unidssync` utilities to identify and resolve all discrepancies. For full information on use and syntax, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

Archived backups should be managed to ensure full data recovery capabilities without sacrificing large amounts of disk space. Remove backups that are no longer needed.

# User back up and restore

It is possible to restore a single user, resource or event calendar through the Calendar Administrator or using the `unirestore` utility. The restore is done using the backup files made using the `unidbbackup` utility.

### Web GUI

Use the Calendar Administrator to restore a user, resource or event calendar account. Click the **Server Administration** tab and then on **Nodes**. Click the pencil icon in the **Actions** column for the node where you will restore the calendar account. Click **Restore Calendars**. Enter the path to the backup file and select the type of calendar account you are restoring (user, resource or event calendar). Click **Apply** to proceed to the next step. Search for the user, resource, or event calendar to be restored.

### Cmd line

Use `unirestore` to restore a calendar account. Use the `-path` option to specify the path to the directory containing the backup `db` directory. Use the `-u` option to specify the UID of the user, resource or event calendar to be restored. Example:

```
% unirestore -u "smithj" -path "/backups/cserver/jan0799" -noAddAttendee
-host hubert3 -p abcdef12 -n 10
```

For full information on the use and syntax of the `unirestore` utility, see Appendix E, "Utilities" in the *Oracle Calendar Reference Manual*.

# 16

# Monitoring Procedures

The log files, found in the $ORACLE_HOME/ocal/log directory, are a useful starting point for troubleshooting problems related to the calendar server's operations or performance.

This chapter covers the following topics:

- Viewing log files
- Interpreting log files

## Viewing log files

To view a log file, go to the $ORACLE_HOME/ocal/log directory and open the file using a text editor. Note that log files for utilities are created the first time the utility is run.

***Table 16–1 Calendar server log files***

| Filename | Description |
|---|---|
| act.log | Tracks calendar usage and monitors possible security violations. To track all signons and signoffs, set the [ENG] activity parameter in unison.ini to TRUE. The size of the act.log file should be closely monitored, since it can increase quickly. |
| csm.log | For the Oracle Calendar Server Manager. |
| cws.log | For the Corporate-Wide Services. Set the [CWS] trace parameter in unison.ini to TRUE to log each transaction performed by the CWS. This will cause the size of the cws.log file to increase quickly, and should only be used for a short time for testing or debugging purposes. |

*Table 16–1 Calendar server log files*

| Filename | Description |
|---|---|
| `das.log` | For the Directory Access Service. |
| `dasstats.log` | For Directory Access Service statistics. |
| `dbi.log` | For node (database) initialization. |
| `dbv.log` | Database operation file. Created only if there is a problem. |
| `dsstats.log` | For directory server (LDAP) calls. |
| `eng.log` | For the Engine. |
| `lck.log` | For the Lock Manager. |
| `ocad.log` † | For the Oracle Calendar Administrator. |
| `script.log` | For all UNIX utilities. |
| `snc.log` | For the Synchronous Network Connections. |
| `stats.log` | Tracks CPU consumption, user wait times, and network traffic for calendar server user sessions. Session statistics are output once a client session is terminated normally. To enable this logging, set the `[ENG] stats` parameter in `unison.ini` to `TRUE`. The size of the `stats.log` file should be closely monitored since it can increase quickly. |
| `<utility>.log` | For various utilities that create and update self-named log files when they are run. |

† The `ocad.log` file is located in the `$ORACLE_HOME/ocad/bin` directory.

## Interpreting log files

Much of the content of the calendar server log files is self-explanatory, namely the sections referring to the status of the various daemons/servers. Interpreting other sections may require the knowledge and resources of a qualified support representative. If you are uncertain about the content of a log file, use Oracle MetaLink to search for log file information or to log a Technical Assistance Request (TAR) online.

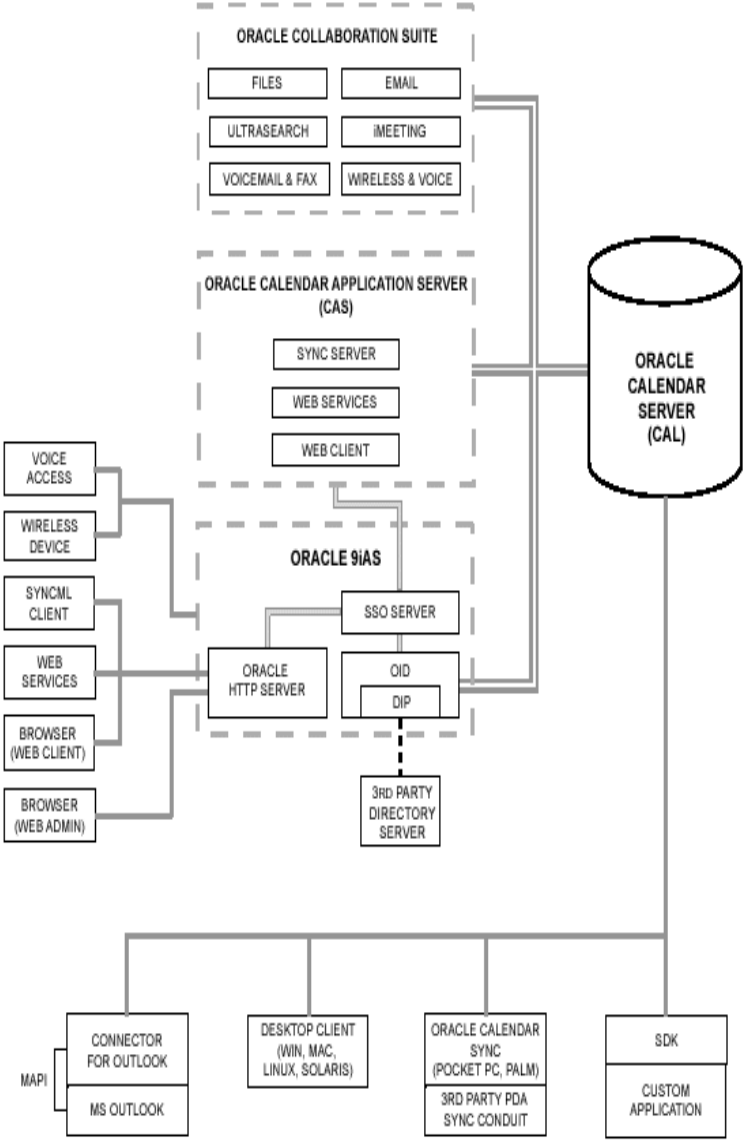
# 17

# Oracle Calendar Application System

This chapter provides an overview of the Oracle Calendar application system, including a summary of its components, architecture information and installation considerations.

## About the Oracle Calendar Application System

Oracle Calendar includes several components, or plug-ins, that are managed in an application server framework called the Oracle Calendar application system (OCAS). OCAS provides a set of shared proprietary APIs that interface with the Oracle Calendar server and run alongside the Oracle HTTP Server (OHS), as shown in the following illustration.


ORACLE COLLABORATION SUITE
FILES
EMAIL
ULTRASEARCH
iMEETING
VOICEMAIL & FAX
WIRELESS & VOICE
ORACLE CALENDAR APPLICATION SERVER (CAS)
SYNC SERVER
WEB SERVICES
WEB CLIENT
ORACLE CALENDAR SERVER (CAL)
VOICE ACCESS
WIRELESS DEVICE
SYNCML CLIENT
WEB SERVICES
BROWSER (WEB CLIENT)
BROWSER (WEB ADMIN)
ORACLE 9iAS
SSO SERVER
ORACLE HTTP SERVER
OID
DIP
3RD PARTY DIRECTORY SERVER
MAPI
CONNECTOR FOR OUTLOOK
MS OUTLOOK
DESKTOP CLIENT (WIN, MAC, LINUX, SOLARIS)
ORACLE CALENDAR SYNC (POCKET PC, PALM)
3RD PARTY PDA SYNC CONDUIT
SDK
CUSTOM APPLICATION

Oracle Calendar's features are provided by loading each of the components on startup. Each component populates the system registry with functionality such as HTTP request handlers, data sync handlers and data access service providers.

# Working with OCAS Components

This section includes descriptions and configuration information for the components of OCAS.

## About OCAS components

**Oracle Calendar Web client** Allows users to share agendas, schedule meetings and book resources and equipment through a wide variety of on-line and off-line access options, including Web, Microsoft Outlook through the Oracle Connector for Outlook, Oracle Calendar Desktop clients for Windows, Macintosh and Linux systems, and integration with wireless and synchronization applications.

The Oracle Calendar Web client back-end is easy to maintain and features a variety of automated maintenance tasks.

**Oracle Sync Server** Offers direct two-way synchronization with Oracle Calendar Server over any standard Hypertext Transfer Protocol (HTTP) connection, opening up the calendar infrastructure to any SyncML-compliant device or application with Internet access. The Oracle Sync Server architecture can also be extended to support third-party standards-based or proprietary infrastructures.

Oracle Sync Server provides a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

**Oracle Calendar Web services** Allows applications to retrieve, through common XML queries, calendaring data for display in any portal, client application, or backend server. iCal data is coded in XML, wherein iCal becomes xCal. SOAP is used to encapsulate the messages for delivery. The calendaring data Web Services SOAP is stored directly on the Calendar Server store. This is in effect the CWSL, or Calendar Web Services Language.

Developers can use the Oracle Calendar Web services toolkit, included with Oracle Calendar, to build Web services applications and create SOAP 1.1 queries. The toolkit contains functionality to search, create, modify, and delete calendar events, as well as search tasks.

### Customizing OCAS components

The configuration files of OCAS and its components are:

- **`ocas.conf`**: OCAS itself
- **`ocwc.conf`**: Oracle Calendar Web client
- **`ocws.conf`**: Oracle Calendar Web services
- **`ocst.conf`**: Oracle Sync Server
- **`ocal.conf`**: Apache directives, included from `$ORACLE_HOME/Apache/Apache/conf/oracle_apache.conf`

> **Note:** `Ocal.conf`, is used to control OCAS instances and fastcgi connections, as described in the section "General Configuration," later in this chapter. `Ocal.conf` parameters are documented in the Oracle HTTP Server Administration Guide.

The configuration files are located in the conf subdirectory of `$ORACLE_HOME/ocas/`. For information on editing their contents, see the *Oracle Calendar Reference Guide*. It is strongly recommended that for reference purposes you make a copy, in either printed or electronic format, of these files before you modify them.

## Installation Considerations

Consider the following factors when installing Oracle Calendar:

- Because OCAS uses a shared memory mechanism, an OCAS installation must be run under one specific user on a host. A single user cannot have two independent OCAS installs, as they will share the same memory. However, having multiple users on a single machine is supported, since each user's shared memory is insulated from the other users.
- For security reasons, it is best that Oracle Sync Server only be accessible through SSL (HTTPS) connections. You may also want to install Oracle Sync Server on a separate host for easier accessibility from phones. Keep in mind that GSM phones can have VPN access inside a firewall, while GPRS phones cannot.
- Standalone installations of Oracle Calendar require Apache 1.3.27 with mod-fastcgi 2.2.12. You will need to find the latter through FTP access, as it is not readily available from the Apache Web site. Standalone does not work with Apache 2.x and mod-fastcgi 2.4.x.

# Post-Installation Issues

Generally, the default settings of the Oracle Calendar application system allow you to get the system working immediately after installation. However, there are some configuration options you should consider, depending on your environment.

## General Configuration

- Run several instances of `ocas.fcgi` (number depends on setup and load). You can configure this in `ocal.conf`.
- You need to run one instance of `ochecklet.fcgi` per installation / host. This is also configured in `ocal.conf`.
- In order to use the sync server, set your KeepAlive parameter in `httpd.conf` or `apache.conf` to 300 seconds, or turn it off. This is done to correspond to the idle-timeout value of 300 seconds in `ocal.conf`.
- Make sure that the `linkdb` and `sessiondb` variables in all hosts' ocas files refer to the same path; for example, the same NFS mount.
- Make sure the file system path to the session database is accessible from all OCAS instances across all hosts. You can set the `sessionpath` in `ocas.conf`.
- Set Authentication, Compression and Encryption (ACE) values in each component's `conf` file. AUTH Web settings for all products should be configured in the `[ACE_PLUGINS_CLIENT]` section of `ocas.conf`.
- Make sure you restart OHS or Apache after any changes to the `conf` files.
- If you experience any problems, check `../logs/ocas_log` for error messages.
- To see if everything is running, open the system page at `/ocas-bin/ocas=fcgi?sub=sys`. This shows information on all components that are installed and running. If a component is not running, it will not appear in the system page.
- To connect to a component with an appropriate client, use the following URLs:
  - **Sync Server** http://<host>:<port>/ocst-bin/ocas.fcgi
  - **Web Services** http://<host>:<port>/ocws-bin/ocas.fcgi
  - **Calendar Web client** http://<host>:<port>/ocwc-bin/ocas.fcgi?sub=web

## Configuring an Apache Server

If you are using the Oracle HTTP Server (OHS), it is automatically configured to recognize the Calendar application system and the Calendar Administrator tool.

However, if you are running a standalone installation of Oracle Calendar using Apache, you should make the following changes to your Apache `httpd.conf` file so that you can use the Oracle Calendar application system:

- Include `<YOUR_ORACLE_HOME>/ocas/conf/ocal.conf` in the Apache `httpd.conf` file
- Set the system library search path to include `<YOUR_ORACLE_HOME>/lib`.
- Set the oracle_home environment variable to `<YOUR_ORACLE_HOME>`.

Keep in mind that you may have to resolve conflicting settings if you had customized your original Apache in a similar manner.

# A

# Disk Space and Memory

This appendix describes the disk space and memory requirements of the calendar server. These requirements can be broken down into the following categories:

- Database disk space requirements
- NFS storage
- Large deployment disk storage recommendation
- Memory requirements

## Database disk space requirements

### Persistent data

Local users, resources and event calendars require disk space for calendar data. Disk space requirements for these local items can be estimated at approximately 2.5 MB per user per year, depending on usage. Note that yearly disk space requirements may be higher with heavy usage of attachments or if scheduling frequent meetings on a daily basis. The disk space requirements for remote users, resources and event calendars are therefore considerably less than those for local items, although the exact figure will depend on usage. For most deployments, where users schedule meetings with other users on the same node, this data is excluded from the preceding estimation.

### Non-persistent data

Each active calendar user also has temporary data files located in the `$ORACLE_HOME/ocal/db/tmp` subdirectory, whose combined size should not exceed 2 MB total per configured user.

# NFS storage

UNIX installations of the Oracle Calendar server support linking the calendar database on a remote NFS file system. If you choose to do so, only the $ORACLE_HOME/ocal/db directory may be stored remotely. All other directories must remain local. This option is not recommended unless you use high-performance NFS equipment.

# Large deployment disk storage recommendation

Calendar data can be broken down into two categories: persistent data and non-persistent data.

Persistent data is stored in the db/nodes directory. It is recommended to use a Redundant Array of Independent Disks (RAID) 0+1 disk storage, striped on a few disks using 8K stripe-sized blocks.

Non-persistent data is stored in the db/tmp directory. For this type of data, it is recommended to use RAID 0 disk storage, striped on a few disks. However, RAID 0+1 disk storage striped on a few disks can be used for failover. Note that a higher number of writes can be expected for non-persistent data. It is therefore recommended to optimize the db/tmp directory for write access.

# Memory requirements

The following is recommended for any large deployment with a minimum of 500 configured users on a given calendar server.

### Memory requirement formula

The following general formula is used to determine the memory required on your system to offer calendar services to your user population:

```
Memory required = calendar sessions + calendar infrastructure
+ disk cache
```

Where a calendar session is defined as a connection to the calendar server. There is 750K of memory used per session, excluding Oracle Calendar Web client sessions.

The value calendar infrastructure is the memory used by the different calendar server processes such as CWS, DAS, etc. which adds up to 250K per user.

The `disk cache` is the memory required by the OS to ensure sufficient cache exists for the disk virtual memory to enhance performance. The disk cache memory needed is 250K per user.

For more information on the deployment and installation of your calendar server, see Chapter 3, "Calendar Deployment".

***Table A–1 Variable definitions***

| Variable | Definition |
|---|---|
| TOTAL_USERS | Total number of configured users hosted by the server on one or many nodes. |
| OCFO_USERS | Number of configured users using Oracle Connector for Outlook. |
| DESKTOP_USERS | Number of configured users using Oracle Calendar Desktop client. |
| WEB_USERS | Number of configured users using Oracle Calendar Web client. |
| FCGI_SESSIONS | Number of fastCGI required to serve WEB_USERS concurrently without having a large fastCGI wait time. Varies between 2.5% to 5% of configured users. |

Using the values that you determine for these variables, the following formula determines memory requirements:

```
(DESKTOP_USERS / 2) * 1 Meg +
(OCFO_USERS * 1 Meg) +
(FCGI_SESSIONS * 15 Meg) +
(TOTAL_USERS * 0.25 Meg)
```

## Assumptions

`(DESKTOP_USERS / 2) * 1 Meg`

This represents the memory used by the different processes (engines, DAS, etc.) for Oracle Calendar desktop client users. The assumption is that 50% of the configured users will be logged on at the same time.

```
OCFO_USERS * 1 Meg
```

This represents the memory used by the different processes (engines, DAS, etc.) for Oracle Connector for Outlook configured users. The assumption is that all configured users will be remain connected throughout the day.

```
FCGI_SESSIONS * 15 Meg
```

For Oracle Calendar Web client sessions, the concurrency rate as well as the memory used by the different processes (engines, DAS, etc.) will be much higher than the other clients. The relationship between Web calendar users and calendar sessions is not one-to-one, one FCGI session serves many Calendar Web client users. Depending on the load and the desired peak usage, the number of FCGI sessions needed is between 2.5% and 5% of the total number of configured calendar users.

```
TOTAL_USERS * 0.25 Meg
```

This represents the memory required by the OS to ensure sufficient cache exists for the disk virtual memory. This is not required for disks with large cache such as EMC disks.

---

**Note:** Additional memory will be needed if Oracle Calendar synchronization clients are used.

---

### Example

Let us calculate the memory requirements for an organization where there will be 2500 configured calendar users. First, the number of users for each type of calendar client must be determined:

```
1000 Oracle Calendar Web clients
 800 Outlook clients with the Oracle Connector for Outlook
 700 Oracle Calendar desktop clients
```

Based on this distribution, compute memory requirements using the defined formula.

Memory for desktop users:

```
(DESKTOP_USERS / 2) * 1 Meg = (700 / 2) = 350 Meg
```

Memory for Outlook users:

```
(OCFO_USERS * 1 Meg) = 800 Meg
```

Memory for Web client users:

```
(FCGI_SESSIONS * 15 Meg) = (5% of 1000) * 15 = 50 * 15 = 750
Meg
```

Disk cache memory for all users:

```
(TOTAL_USERS * 0.25 Meg) = 2500 * 0.25 = 625 Meg
```

Total memory requirements for all users:

`2525 Meg` needed to serve this organization's 2500 calendar users.

# B

# Adjusting Calendar Kernel Parameters

This appendix details the modifications that must be made to certain kernel parameters and operating environments in order to ensure that sufficient resources are allocated to the calendar server. It also details issues in server configuration that must be considered in order to support certain special operating environments such as Solaris clusters.



## Adjusting kernel parameters

The UNIX system parameters that need adjustment are used to control resource consumption on a user-, process-, or system-wide basis. In the case of either a user or a process parameter, the new value for the parameter should be the existing value or the calendar server requirement, whichever is larger. In the case of a system-wide parameter, the server requirement must be added to the existing value to calculate a new value.

The assumptions concerning the server `$ORACLE_HOME/ocal/misc/unison.ini` file parameters are:

- `Number of nodes = 10`
- `[ENG] maxsessions = 2500`

# Adjusting the Solaris kernel parameters

Solaris provides tunable parameters for the kernel and kernel modules. While normally you should not need to change these parameters, there are special circumstances under which it is necessary. Under Solaris, kernel parameters are modified by directly editing the `/etc/system` file with a standard text editor. All of the parameters are set using the following syntax:

```
set rlim_fd_cur=1024
set rlim_fd_max=4117
```

In addition, the message queue and semaphore parameters must include the name of the specific module to be modified. The syntax is as follows:

```
set msgsys:msginfo_msgmni=351
set semsys:seminfo_semmni=345
```

In order for the preceding changes to take effect, reboot the system.

To see the current values assigned to the kernel parameters, use the `sysdef` command. For example:

```
% sysdef
```

The maximum number of open files is unlimited under Solaris, and therefore does not need to be set.

The following forceload directives must be specified in the `/etc/system` kernel configuration file:

```
forceload: sys/semsys
forceload: sys/shmsys
```

In the “NEW setting” column, the “max” function returns the larger of the two arguments.

***Table B–1 Solaris Kernel Parameters***

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **Limits** | | | | |
| rlim_fd_cur | file descriptors soft limit | X | 1024 | max(X,1024) |
| rlim_fd_max | file descriptors hard limit | X | 4117 | max(X,4117) |

***Table B–1 Solaris Kernel Parameters***

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **IPC Messages** | | | | |
| msgsys:msginfo_msgmni | message queue identifiers | X | 2750 | X + 2750 |
| msgsys:msginfo_msgmax | max. message size | X | 4096 | max(X,4096) |
| msgsys:msginfo_msgmnb | max. bytes on queue | X | 360000 | max(X,360000) |
| msgsys:msginfo_msgtql | max. message headers | X | 2500 | max(X,2500) |
| **IPC Semaphores** | | | | |
| semsys:seminfo_semmni | max. semaphore sets system-wide | X | 42 | X + 42 |
| semsys:seminfo_semmns | max. semaphores system-wide | X | 168 | X + 168 |
| semsys:seminfo_semmnu | max. undo structures system-wide | X | 42 | X + 42 |
| semsys:seminfo_semmsl | max. semaphores per set | X | 12 | max(X,12) |
| semsys:seminfo_semopm | max. operations per semop call | X | 12 | max(X,12) |
| semsys:seminfo_semume | max. undo structures per process | X | 42 | max(X,42) |
| semsys:seminfo_semvmx | max. value of a semaphore | X | 32767 | 32767 |
| semsys:seminfo_semaem | max. adjust-on-exit value | X | 16384 | 16384 |
| **IPC Shared Memory** | | | | |
| shmsys:shminfo_shmmax | max. shared memory segment size | X | 20 000 000 | max(X,20000000) |
| shmsys:shminfo_shmmin | min. shared memory segment size | X | 1 | 1 |
| shmsys:shminfo_shmmni | max. shm identifiers system-wide | X | 18 | X + 18 |
| shmsys:shminfo_shmseg | max. shm segments per process | X | 18 | max(X,18) |

The following is an actual example:

***Table B–2 Solaris kernel parameters (example)***

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **Limits** | | | | |
| rlim_fd_cur | file descriptors soft limit | 1024 | 1024 | 1024 |
| rlim_fd_max | file descriptors hard limit | 1024 | 4117 | 4117 |
| **IPC Messages** | | | | |
| msgsys:msginfo_msgmni | message queue identifiers | 50 | 2750 | 2800 |
| msgsys:msginfo_msgmax | max. message size | 2048 | 4096 | 4096 |
| msgsys:msginfo_msgmnb | max. bytes on queue | 4096 | 360000 | 360000 |
| msgsys:msginfo_msgtql | max. message headers | 40 | 2500 | 2500 |
| **IPC Semaphores** | | | | |
| semsys:seminfo_semmni | max. semaphore sets system-wide | 100 | 42 | 142 |
| semsys:seminfo_semmns | max. semaphores system-wide | 256 | 168 | 424 |
| semsys:seminfo_semmnu | max. undo structures system-wide | 4096 | 42 | 4138 |
| semsys:seminfo_semmsl | max. semaphores per set | 256 | 12 | 256 |
| semsys:seminfo_semopm | max. operations per semop call | 10 | 12 | 12 |
| semsys:seminfo_semume | max. undo structures per process | 10 | 42 | 42 |
| semsys:seminfo_semvmx | max. value of a semaphore | 32767 | 32767 | 32767 |
| semsys:seminfo_semaem | max. adjust-on-exit value | 16384 | 16384 | 16384 |
| **IPC Shared Memory** | | | | |
| shmsys:shminfo_shmmax | max. shared memory segment size | 4294967295 | 20 000 000 | 4294967295 |

*Table B–2 Solaris kernel parameters (example)*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| shmsys:shminfo_shmmin | min. shared memory segment size | 1 | 1 | 1 |
| shmsys:shminfo_shmmni | max. shm identifiers system-wide | 100 | 18 | 118 |
| shmsys:shminfo_shmseg | max. shm segments per process | 10 | 18 | 18 |

## Additional reading

Information on modifying the kernel parameters is available in the *Solaris Tunable Parameters Reference Manual* by Sun Microsystems, Inc.

For more information on modifying the kernel parameters under Solaris, refer to *Solaris Internals: Core Kernel Components* (ISBN: 0-13-022496-0)

# Adjusting the HP-UX kernel parameters

The following section describes the maximum kernel requirements, and as such should apply to most HP-UX installations. All parameters can be modified *via* SAM, a menu-based system administration manager.

The value of `semmsl` is set at 500 and is not configurable on HP-UX.

In the "NEW setting" column, the "max" function returns the larger of the two arguments.

*Table B–3 HP-UX kernel parameters*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **Open Files** | | | | |
| maxfiles | file descriptors soft limit | X | 1024 | max(X,1024) |
| maxfiles_lim † | file descriptors hard limit | X | 3861 | max(X,3861) |
| nfile | max. file descriptors system-wide | X | 100000 | X + 100000 |
| **Process Management** | | | | |

*Table B–3 HP-UX kernel parameters*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| max_thread_proc | max. threads per process | X | 210 | max(X,210) |
| nkthread | max. kernel thread system-wide | X | 2850 | X + 2850 |
| maxuprc | max. user processes | X | 99 | max(X,99) |
| nproc | max. process system-wide | X | 99 | X + 99 |
| **IPC Messages** | | | | |
| mesg | enable / disable IPC messages | X | 1 | 1 |
| msgmap | message free-space map size | X | 2502 | msgtql + 2 |
| msgmax | max. message size | X | 4096 | max(X,4096) |
| msgmnb | max. bytes in message queue | X | 65535 | max(X,65535) |
| msgmni | max. msg queues system-wide | X | 2750 | X + 2750 |
| msgseg | max. msg segments system-wide | X | 2500 | max(X,2500) |
| msgssz | message segment size | X | 159 | max(X,159) |
| msgtql | max. messages system-wide | X | 2500 | X + 2500 |
| **IPC Semaphores** | | | | |
| sema | enable / disable semaphores | X | 1 | 1 |
| semaem | sem value-change limit | X | 16384 | max(X,16384) |
| semmap | size of free-sem resource map | X | 44 | max(X,44) |
| semmni | max. sem sets system-wide | X | 42 | X + 42 |
| semmns | max. user sem system-wide | X | 168 | X + 168 |
| semmnu | max. undo per semaphore | X | 42 | max(X,42) |
| semume | max. sem undo per process | X | 42 | max(X,42) |
| semvmx | max. value of a semaphore | X | 32767 | max(X,32767) |
| **IPC Shared Memory** | | | | |
| shmem | enable / disable shared memory | X | 1 | 1 |
| shmmax | max. shmem segment size | X | 20000000 | max(X,20000000) |

*Table B–3 HP-UX kernel parameters*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| shmmni | max. segments system-wide | X | 18 | X + 18 |
| shmseg | max. segments per process | X | 18 | max(X,18) |

† Refer to "Setting maxfiles / maxfiles_lim above 2048 on HP-UX 11.0"

## Setting maxfiles/maxfiles_lim above 2048 on HP-UX 11.0

This section describes how to set the `maxfiles` and / or `maxfiles_lim` kernel parameters(s) to greater than 2048 when running HP-UX 11.0.

Although the `/usr/conf/master.d/core-hpux` file contains the following lines:

```
*range maxfiles<=2048

*range maxfiles_lim<=2048
```

the kernel can still be compiled manually with values for `maxfiles` and `maxfiles_lim` larger than 2048. Note that 60000 is the upper bound for these parameters in HP-UX 10.20 and above.

When trying to set either `maxfiles` or `maxfiles_lim` to larger than 2048, for example 4000, in System Administration Manager (SAM) the following error occurs:

The value specified for tunable parameter "maxfiles", "4000", evaluates to "4000" which is more than the maximum allowed value of "2048". SAM did not have this problem in HP-UX 10.20.

At HP-UX 11.0 SAM relies on the `/usr/conf/master.d/*` files to set the upper and lower values for kernel parameters. As already discussed the `/usr/conf/master.d/core-hpux` file contains the following lines:

```
*range maxfiles<=2048

*range maxfiles_lim<=2048
```

To allow SAM to support setting either `maxfiles` or `maxfiles_lim` to larger than 2048, the preceding lines in the `/usr/conf/master.d/core-hpux` file must be changed to:

```
*range maxfiles<=60000
```

```
*range maxfiles_lim<=60000
```

After making the change, SAM may still have stored the old values within the `/var/sam/boot.config` file. The `boot.config` file can be move out of the way to cause SAM to recognize the changes made to the `core-hpux` file. After moving the `boot.config` file, restarting SAM should cause this file to be rebuilt using the new values. `/usr/sam/lbin/getkinfo -b` can also be used to recreate the `boot.config` file.

Similarly, if `maxfiles` or `maxfiles_lim` have been set manually to values larger than 2048, the preceding steps can be used to prevent SAM from erroring on the parameter's values when entering the **Configurable Parameters** menu item within SAM. Without making the preceding changes, SAM will typically reset the "Pending Value" for these parameters back to 2048 anytime SAM is used.

The following is an actual example:

***Table B–4 HP-UX kernel parameters (example)***

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **Open Files** | | | | |
| maxfiles | file descriptors soft limit | 60 | 1024 | 1024 |
| maxfiles_lim | file descriptors hard limit | 1024 | 3861 | 3861 |
| nfile | max. file descriptors system-wide | 63488 | 100000 | 163488 |
| **Process Management** | | | | |
| max_thread_ proc | max. threads per process | 256 | 210 | 256 |
| nkthread | max. kernel thread system-wide | 7184 | 2850 | 10034 |
| maxuprc | max. user processes | 3686 | 99 | 3686 |
| nproc | max. process system-wide | 4096 | 99 | 4195 |
| **IPC Messages** | | | | |
| mesg | enable / disable IPC messages | 1 | 1 | 1 |
| msgmap | message free-space map size | 4098 | 2502 | 6598 |
| msgmax | max. message size | 8192 | 4096 | 8192 |
| msgmnb | max. bytes in message queue | 16384 | 65535 | 65535 |
| msgmni | max. msg queues system-wide | 4096 | 2750 | 6846 |

*Table B–4 HP-UX kernel parameters (example)*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| msgseg | max. msg segments system-wide | 32767 | 2500 | 32767 |
| msgssz | message segment size | 8 | 159 | 159 |
| msgtql | max. messages system-wide | 4096 | 2500 | 6596 |
| **IPC Semaphores** | | | | |
| sema | enable / disable semaphores | 1 | 1 | 1 |
| semaem | sem value-change limit | 16384 | 16384 | 16384 |
| semmap | size of free-sem resource map | 4098 | 44 | 4098 |
| semmni | max. sem sets system-wide | 4096 | 42 | 4138 |
| semmns | max. user sem system-wide | 8192 | 168 | 8360 |
| semmnu | max. undo per semaphore | 4092 | 42 | 4092 |
| semume | max. sem undo per process | 10 | 42 | 42 |
| semvmx | max. value of a semaphore | 32767 | 32767 | 32767 |
| **IPC Shared Memory** | | | | |
| shmem | enable / disable shared memory | 1 | 1 | 1 |
| shmmax | max. shmem segment size | 4294967295 | 20000000 | 4294967295 |
| shmmni | max. segments system-wide | 512 | 18 | 530 |
| shmseg | max. segments per process | 32 | 18 | 32 |

## Additional reading

For more information on modifying the kernel parameters under HP-UX, refer to http://docs.hp.com/hpux/onlinedocs/939/KCParms/KCparams.OverviewAll.html

# Adjusting the Linux kernel parameters

Linux platforms require a change to the system kernel parameters in order to support the Oracle Calendar server's default configuration. Use the following table to determine the calendar server requirements for each parameter, where the value of X for each parameter is equal to its previous setting.

Note that the `kernel.sem` parameter is multi-valued, accepting a value of the format: `'semmsl semmns semopm semmni'`.

To increase the file descriptor's soft and hard limits, edit `/etc/security/limits.conf` and add the following lines:

```
*soft nofile 1024
*hard nofile 65535
```

Note that you may need to log out and back in again before the changes take effect.

In the "NEW setting" column, the "max" function returns the larger of the two arguments.

***Table B–5 Kernel tuning requirements for Linux***

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **/etc/security/limits.conf** | | | | |
| * soft nofile | file descriptors soft limit | X | 1024 | max(X,1024) |
| * hard nofile | file descriptors hard limit | X | 198 | max(X,198) |
| **/etc/sysctl.conf** | | | | |
| **Files** | | | | |
| fs.file-max | max. file descriptors system-wide | X | 262144 | X + 262144 |
| **IPC Messages** | | | | |
| kernel.msgmni | message queue identifiers | X | 2750 | X + 2750 |
| kernel.msgmax | max. message size | X | 4096 | max(X,4096) |
| kernel.msgmnb | max. bytes on queue | X | 65535 | max(X,65535) |
| **IPC Semaphores** | | | | |
| kernel.sem (1: semmsl) | max. semaphores per set | X | 12 | max(X,12) |
| kernel.sem (2: semmns) | max. semaphores system-wide | X | 42 | X + 42 |

*Table B–5 Kernel tuning requirements for Linux*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| kernel.sem (3: semopm) | max. operations per semop call | X | 12 | max(X,12) |
| kernel.sem (4: semmni) | max. semaphore sets system-wide | X | 42 | X + 42 |
| **IPC Shared Memory** | | | | |
| kernel.shmmax | max. shared memory segment size | X | 20000000 | max(X,20000000) |
| kernel.shmmni | max. shm identifiers system-wide | X | 18 | X + 18 |
| kernel.shmall | total shm pages available system-wide | X | 2747 | X + 2747 |

The following is an actual example:

*Table B–6 Kernel tuning requirements for Linux (example)*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **/etc/security/limits.conf** | | | | |
| * soft nofile | file descriptors soft limit | 1024 | 1024 | 1024 |
| * hard nofile | file descriptors hard limit | 1024 | 198 | 1024 |
| **/etc/sysctl.conf** | | | | |
| **Files** | | | | |
| fs.file-max | max. file descriptors system-wide | 65535 | 262144 | 327679 |
| **IPC Messages** | | | | |
| kernel.msgmni | message queue identifiers | 128 | 2750 | 2878 |
| kernel.msgmax | max. message size | 8192 | 4096 | 8192 |
| kernel.msgmnb | max. bytes on queue | 16384 | 65535 | 65535 |

*Table B–6 Kernel tuning requirements for Linux (example)*

| Kernel Parameter | Parameter Description | CURRENT setting | Calendar server requirement | NEW setting |
|---|---|---|---|---|
| **IPC Semaphores** | | | | |
| kernel.sem (1: semmsl) | max. semaphores per set | 100 | 12 | 100 |
| kernel.sem (2: semmns) | max. semaphores system-wide | 256 | 42 | 298 |
| kernel.sem (3: semopm) | max. operations per semop call | 100 | 12 | 100 |
| kernel.sem (4: semmni) | max. semaphore sets system-wide | 100 | 42 | 142 |
| **IPC Shared Memory** | | | | |
| kernel.shmmax | max. shared memory segment size | 2147483648 | 20000000 | 2147483648 |
| kernel.shmmni | max. shm identifiers system-wide | 100 | 18 | 118 |
| kernel.shmall | total shm pages available system-wide | 3276800 | 2747 | 3279547 |

# Using operating system clusters

The term “cluster” does not refer to the same concept as a calendar server cluster — a calendar server cluster is a node network in which one node is designated a “master node” for the purposes of client sign-in, automated registration, etc., while an operating system cluster is considered to be a system in which two or more machines can be used to manage the same data, providing failover capabilities.

In these environments, it is important to differentiate between the *physical host name*, which is the actual host name of a given machine, and the *cluster host name*, which is the host name of the cluster containing that machine. If you intend to use your calendar server in a cluster environment, you should set all parameters in the `$ORACLE_HOME/ocal/misc/unison.ini` file that require the host name of the local host to the *cluster* host name. In addition, you must add the `[ENG] calendarhostname` parameter to `unison.ini`, and set its value to the cluster host name. Finally, if using an external LDAP directory server, you must ensure that the `[YOURHOSTNAME, unidas]` section specifies the cluster host name in place of `YOURHOSTNAME`.

If using a node network, ensure also that your `$ORACLE_HOME/ocal/misc/nodes.ini` file uses only cluster host names instead of physical host names, and ensure that all clients are using the cluster host name to sign in to the calendar server.

---

**Note:** When a machine containing a master node switches over to another machine in the cluster, Oracle Calendar Web clients can have difficulty signing in, since master nodes currently identify themselves to clients using physical host names.

---

# C

# Security

Security is a primary concern for any application used to manage sensitive, personal information. A number of options are available to an administrator seeking to enhance or customize the security of a calendar server installation. In addition to increasing the security of the operating environment and implementing good maintenance and monitoring practices, calendar server administrators have access to a configurable, extensible Authentication, Compression and Encryption (ACE) framework.

This appendix describes the structure and configuration of the authentication, compression and encryption methods. Additional security considerations for installations using a directory server are detailed, as well as a number of other measures that may be employed to further protect calendar data.



## ACE framework

The ACE framework was developed to allow administrators to ensure the security and integrity of all data passing between calendar servers, and between server and client.

Data passes between the server and clients, and between multiple servers if nodes are distributed across more than one host. If an external directory is used, data also passes between the calendar server and the LDAP directory server. The ACE framework applies only to communication between the calendar server(s) and clients. See "Directory server security" later in this chapter for a separate discussion

of the security options for data passing between calendar servers and their supporting directory servers.

Secure connections may involve the use of compression (to reduce the network bandwidth required for communications) and/or encryption (to enhance the security of network communications). Both compression and encryption increase the amount of CPU time required to prepare the communication for transmission. The impact on performance varies with the methods. In general, the better the compression or the more secure the encryption, the greater the impact on performance.

## Secure connections to clients and other calendar servers

Secure connections to calendar clients and other calendar servers are controlled by a configurable set of authentication, compression and encryption methods. These methods are determined at the time the connection is requested. The ACE methods are both configurable and extensible. See "Configuration" for the relevant configuration parameters, and "Extending the ACE framework" for details on extending the available set of methods later in this chapter.

### Secure connections to clients

---

**Note:** Only desktop calendar clients 5.0 and higher, Oracle CorporateSync 3.0 and higher, Web clients 2.0 and higher and Oracle Outlook Connector support the ACE framework. Other clients including Oracle CorporateSync for Mac 2.1.x require the use of the `cs-basic` authentication method. If you plan to use Oracle CorporateSync 2.1.x for the Mac, you must add `cs-basic` to the list of supported authentication mechanisms specified by the `[AUTHENTICATION] supported` parameter.

---

The calendar server negotiates with a client as follows.

1. The client starts up and connects to the server.
2. The client queries the server for the supported and default authentication, compression, and encryption methods.
3. The server returns a list of the supported and default authentication, compression and encryption methods.

4. If the client cannot support one of the default methods, the server and client negotiate using the list of supported methods sent to the client in step 3 to agree on a method that both support. Note that one of the supported methods for both compression and encryption is “none”, making both compression and encryption optional.
5. The server authenticates the user using the negotiated authentication method.
6. The client and server communicate using the agreed upon methods for the duration of the user session.

---

**Note:** If the client and server cannot agree on authentication, compression and encryption methods, the negotiation fails and the server does not accept requests from the client.

---

## Secure connections to another calendar server

The server negotiates with another calendar server as follows.

1. Server A receives a request from Server B.
2. Server A sends Server B a list of the supported and default authentication, compression, and encryption methods.
3. If Server B cannot support one of the default methods, Server A and Server B negotiate using the lists of supported methods sent in step 2 to agree on a method that both support. Note that one of the supported methods for both compression and encryption is “none”, making both compression and encryption optional.
4. Server A authenticates Server B using the negotiated authentication method.
5. Servers A and B communicate using the agreed upon methods for the duration of the connection.

   Recall that communication between two calendar servers is through the `uniengd`. In this case, the `uniengd` on Server B asks the `unisncd` on Server B for a connection to a `uniengd` on Server A. The methods are in effect until the requesting `uniengd` on Server B returns the connection to the `unisncd`.

**Note:** If the two servers cannot agree on authentication, compression and encryption methods, the negotiation fails and Server A does not accept requests from Server B.

## Configuration

To enable the ACE framework and ensure secure server-to-client or server-to-server connections in a node network, set the `[ACE] frameworkenable` parameter in the `$ORACLE_HOME/ocal/misc/unison.ini` file to `TRUE`.

The following table lists the parameters used to configure the authentication, compression and encryption methods used for communication within a calendar network (server to client, server to server). See "Extending the ACE framework" later in this chapter for information on extending the sets of supported methods. Consult Appendix C, "Server Parameters" of the *Oracle Calendar Reference Manual*, for details on these parameters.

***Table C–1 ACE configuration parameters***

| Section | Parameter | Description |
|---|---|---|
| `[ACE]` | `frameworkenable` | Enable the ACE framework |
| | `minbufsizetocompress` | Minimum buffer size for compression |
| | `slibcachecount` | Maximum number of shared libraries per type |
| | `workbufsize` | Buffer size for compression and encryption |
| `[ACE_PLUGINS]` | `sasl_KERBEROS_V4_useridneeded` or `sasl_GSSAPI_useridneeded` | SASL — userID needed |
| | `sasl_KERBEROS_V4_mac_realm` | SASL — Kerberos realm for Mac clients |
| | `sasl_KERBEROS_V4_srvtab` | SASL — Path to Kerberos "srvtab" file |
| `[ACE_PLUGINS_CLIENT]` | `web_attribute_name` | Web authentication - user attribute name |

***Table C–1 ACE configuration parameters***

| Section | Parameter | Description |
|---|---|---|
| | web_attribute_type | Web authentication - user attribute type |
| | web_attribute_valuemax | Web authentication - maximum size of user attribute name |
| | web_cacheexpiresec | Web authentication time-out |
| | web_cachesize | Web authentication - cache size |
| | web_CAL_sharedkey | Web authentication - Web:CAL shared key |
| | web_custom_script | Web authentication - custom user-ID to attribute mapping script |
| | web_tmppath | Web authentication - path for custom script temporary files |
| [ACE_PLUGINS_SERVER] | web_CAL_sharedkey | Web authentication — shared key |
| [AUTHENTICATION] | admindefault | Default authentication method for administrators |
| | default | Default authentication method for clients |
| | keepresourcepwdincaldb | Location of resource passwords for authentication |
| | servicedefault | Default authentication method for other servers |
| | supported | Supported authentication methods for clients |
| [COMPRESSION] | admindefault | Default compression method for administrators |
| | default | Default compression method for clients |
| | servicedefault | Default compression method for other servers |

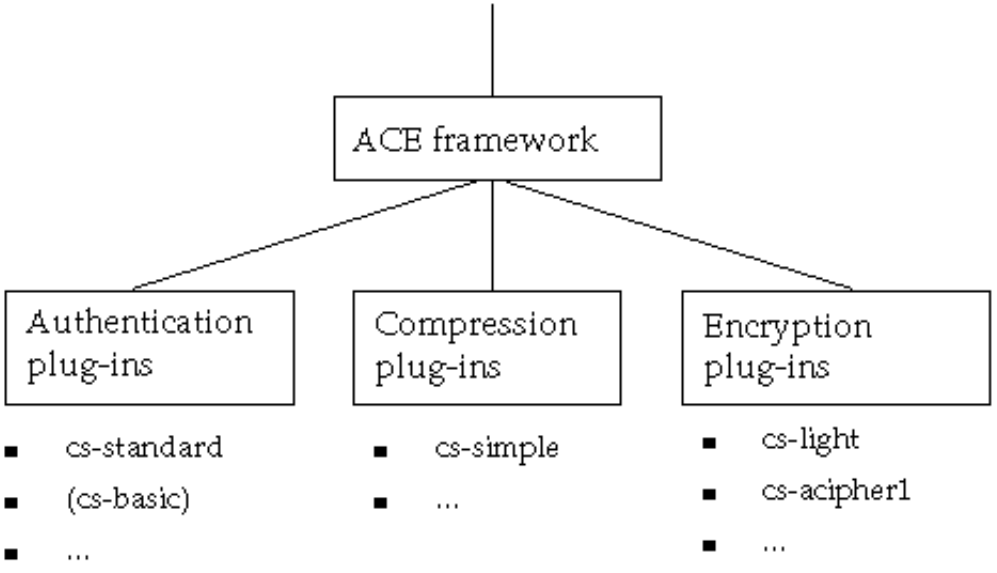
*Table C–1 ACE configuration parameters*

| Section | Parameter | Description |
|---|---|---|
| [ENCRYPTION] | admindefault | Default encryption method for administrators |
| | default | Default encryption method for clients |
| | needsauthenticate | Encryption methods requiring prior authentication |
| | servicedefault | Default encryption method for other servers |
| | supported | Supported encryption methods |

## Extending the ACE framework

The ACE framework provides an interface to an extensible set of authentication, compression, and encryption plug-ins. This section describes the use of these plug-ins, and details the mechanism for extending the set of plug-ins available in the server.

***Figure C–1 ACE framework architecture***




## Plug-ins

Each plug-in is a shared library under UNIX, or a DLL under NT. The name of the plug-in contains a substring that indicates the type of the plug-in, as shown in the following table.

| Substring | Type of plug-in | Examples |
|---|---|---|
| `aut_` | Authentication | `aut_cs-standard.dll` (NT)<br>`libaut_cs-standard.sl` (HP-UX)<br>`libaut_cs-standard.so` (SunOS) |
| `cmp_` | Compression | `cmp_cs-simple.dll` (NT)<br>`libcmp_cs-simple.sl` (HP-UX)<br>`libcmp_cs-simple.so` (SunOS) |

| Substring | Type of plug-in | Examples |
|---|---|---|
| enc_ | Encryption | enc_cs-acipher1.dll (NT)<br>libenc_cs-acipher1.sl (HP-UX)<br>libenc_cs-acipher1.so (SunOS) |

**Note:** The cs-basic authentication plug-in is not a true plug-in in that it is not a shared library. It pre-dates the ACE framework as the authentication method built into the server. The cs-basic authentication plug-in must be enabled manually if older clients are to be used, including Oracle CorporateSync 2.1.x for Mac.

## Extending the set of plug-ins

To extend the set of plug-ins available through the ACE framework, first install the plug-in on your system, and then integrate it into the server. The installation of plug-ins is the responsibility of the system administrator. Consult the appropriate documentation for details. To integrate the plug-in into the calendar server, add the appropriate keywords to one or more places in the unison.ini file.

For all methods except those that support sub-mechanisms, derive the keyword from the name of the plug-in in the following manner. Remove the substring “aut_”, “cmp_”, or “enc_” and all characters that precede it. Remove the filename extension and the period that precedes it. The remaining string is the keyword to add to the unison.ini file.

| Plug-in Name | Keyword |
|---|---|
| aut_cs-standard.dll | cs-standard |
| libcmp_cs-simple.sl | cs-simple |
| libenc_cs-acipher1.so | cs-acipher1 |

In the case of plug-ins which support sub-mechanisms, the keyword has the format <plug-in_name>:<sub-mechanism_name>. Derive <plug-in_name> as described earlier in this chapter. Consult the plug-in documentation to determine <sub-mechanism_name>. For example:

| Plug-in Name | Keyword |
|---|---|
| libaut_sasl.so | sasl:GSSAPI |

Note that sasl:GSSAPI is not certified with this release of the Oracle Calendar server.

Once you have determined the keyword, add it to the appropriate list of supported methods in the unison.ini file. Add authentication methods to the [AUTHENTICATION] supported parameter, compression methods to the [COMPRESSION] supported parameter, and encryption methods to the [ENCRYPTION] supported parameter. If you want the new method to be the default, also set the appropriate default and / or servicedefault parameters. For more details on these parameters, consult Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual*.

## Web authentication plug-in

The Web authentication plug-in configures the calendar server to trust Web server authentication methods, allowing users to view their calendars without having to sign in explicitly to the calendar server.

This plug-in is supported by Oracle Calendar Web clients version 3.0 or greater and Oracle calendar servers of version 5.2 and 5.3. Future versions of the calendar server will feature enhanced Web authentication capabilities through an updated version of this plug-in. The aut_web module must be configured correctly on both the client side and server side.

Note that the Oracle Collaboration Suite installation procedure configures your calendar Web client and server to use this Web authentication plug-in by default, in order to provide support for Single Sign-On.

### Configuring your calendar server

1. Edit the users $ORACLE_HOME/ocal/misc/unison.ini file.
2. Ensure that the [ACE] frameworkenable parameter is set to TRUE.
3. Add the Web authentication plug-in, represented by the string web:OTMT, to the list of supported authentication methods given in the [AUTHENTICATION] supported parameter. For example:

```
[AUTHENTICATION]
supported = {cs-standard, web:OTMT}
```

## Configuring your Web calendar client

1. Open the `$ORACLE_HOME/ocas/conf/ocwc.conf` file for editing.
2. Enable the Web authentication plug-in by changing the value of the `[ACE] Authentication` parameter to `web:OTMT`. Set this parameter as follows:

   ```
   [ACE]
   Authentication = web:OTMT
   ```

3. Modify the parameter, `[ACE_PLUGINS_CLIENT] web_attribute_type`, to specify the calendar user attribute that will be used to match the value of the Web server environment variable. Currently, this parameter may take either of two values:

   - `userid`, which forces the calendar server to identify users by comparing the value of the Web server environment variable with calendar server UIDs (default value), or
   - `custom`, which processes the value of the Web server environment variable through a separate script before comparing it with calendar server UIDs.

   For example:

   ```
   web_attribute_type = userid
   ```

   If you choose to use the `custom` option for the `[ACE_PLUGINS_CLIENT] web_attribute_type` parameter, then you must add another parameter, `[ACE_PLUGINS_CLIENT] web_custom_script`, to specify the script to be used. For example:

   ```
   web_custom_script = /usr/local/apache/ctw-bin/lexacal/custom.sh
   ```

   Sample script: This sample shell script (UNIX only) takes e-mail addresses stored in the Web server environment variable `SSL_CLIENT_S_DN_Email`, strips off the `@` character and all characters following, and returns the remaining characters to be matched against users' calendar server UIDs:

   ```
   #!/bin/sh
   echo userid
   echo $SSL_CLIENT_S_DN_Email | sed s/\@.*//g
   ```

   To use this script, save it as an executable on your Web server, and supply the path and file name using the `web_custom_script` parameter. Use this script as a template for designing your own custom scripts. Output must be two lines: the attribute type on the first line and the value on the second.

**Note:** If more than one Web client application is running on the same UNIX machine using the custom option, each Web client must run under a different user name.

4. Modify the parameter, `[ACE_PLUGINS_CLIENT] web_attribute_name`, to specify the Web server environment variable to use for identifying calendar users. For example:

   ```
   web_attribute_name = REMOTE_USER
   ```

   Any environment variable may be used. When the `web_attribute_type` parameter is set to `userid`, the value of the environment variable specified by the `web_attribute_name` parameter must be equivalent to the calendar server UID. When the `web_attribute_type` parameter is set to custom, the specified script must convert the value of the environment variable specified by the `web_attribute_name` parameter into a valid calendar server UID.

# Directory server security

Secure Sockets Layer (SSL) encryption is used by default for all connections to Oracle Internet Directory to protect the data that flows between the calendar server and the directory server, and prevent passwords from being sent across the wire in clear text.

# Other security considerations

The following safeguards can be used to enhance the security of calendar data.

## Dedicated Server

We recommend that the calendar server, if financial resources permit, be placed on a dedicated computer. Additionally, turn off any TCP and UDP services on the host which are not critical to the calendar server (e.g., ftp, NFS server and client, X server, etc.).

## Password Management

Users and administrators should take advantage of the other directory administration tools provided for password management. For calendar server SYSOP passwords, the following are policy / procedure recommendations:

- Passwords should never be empty (or blank). This is especially important for the SYSOP or node administrator password.
- Passwords should never be words, names, or personal information which would be easy for others to guess.
- Passwords should be at least 8 characters long, and contain a combination of letters and numbers.
- Avoid using the same password to access the calendar server and other mission-critical systems (although this may not be possible if these applications all use the same directory server).

The calendar server 9.0.4 supports passwords of up to 63 characters. Users who set their passwords to more than 15 characters may not be able to sign-in using calendar clients older than 9.0.4 which only support password lengths of 15 characters.

Other configuration parameters to consider are `ssignin` and `ssigninrestrictions` in the `[LIMITS]` section which control whether a user can use the desktop clients' automatic sign-in feature. Parameter `invalidlogin_enable` in the `[ENG]` section can be used to enable the invalid sign-in counting mechanism. The `[ENG] allowpasswordchange_user` parameter can be used to stop users from changing their password. For more details on these parameters, consult Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual*.

## Trust Management

Even if the server is dedicated to the calendaring application, there are still additional security safeguards to consider.

If you have security servers within your organization, consider sending audit trail information from the calendar server to your central security server. Turn on auditing for the server and conduct spot audits of the commands issued by the calendar user. The server protects a great deal of aggregate data; ensure that your backups are protected from theft. Consider separate ownership of the root / administrator (auditing account) and the calendar (server management) accounts. This would allow root / administrator to detect potential abuses by the calendar owner.

## Networking

It is more secure to run mission-critical applications within firewall-protected intranets. Make sure that the dial-up connections to your intranet are protected. This can be improved by using one-time password technology (e.g. SecurID). As with many TCP/IP protocols, promiscuous listening (where the attacker monitors network traffic) is a threat in any broadcast network. A number of steps can be performed to reduce the risk of this threat:

- Physically protect hubs and routers. Use switched hubs when possible, especially on the server itself. Some hubs will block unauthorized, or unregistered MAC (or ethernet addresses) on the LAN.
- Consider router filtering between untrusted internal networks.
- Use commercial firewalls to allow more complex TCP/IP filtering rules.

## Auditing

The server generates a number of useful audit trails. It is important to become familiar with these audit trails, and to check them regularly. Many commands will create log files on error conditions. Routinely check for the existence of new log files, and review their contents. Monitor the `$ORACLE_HOME/ocal/log/act.log` for login attempt abuses. You can detect login attempt abuses from the originating IP addresses. After the application is initially installed, record the file dates, file sizes, and checksums of all the binaries (the `unicksum` utility generates a checksum for a file). Periodically check that none of the binaries have been edited by comparing the current file dates, file sizes, and checksums with those recorded. Review <`temp`> directories for any suspicious files as they can be used as work areas.

## Backup and Recovery

Calendar data is very important and should be backed up regularly. See Chapter 15, "Node Maintenance" for more information.

## Application Security

The server supports a very rich set of user-controlled access privileges (or rights). It is important to train end-users on how these capabilities can be managed, so that the users' information is protected from unauthorized access.

Try to limit assigning designate rights. You should only give designate rights to trusted individuals.

The default designate rights should be no designate rights. Set the viewing rights to no privileges, and add privileges as needed.

There are a number of overall limits, set by the server administrator, that can be set for all users.

Disabling attachments (`[LIMITS] allowattachments`) can prevent users from propagating proprietary information improperly. Setting maximum attachment size (`[LIMITS] maxattachmentsize`) can help prevent denial of service errors caused by very large files that cause a server to run out of disk space. For more details on these parameters and others, consult Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual*.

## Calendar Administrator

It is recommended that you always use Secure Sockets Layer (SSL) encryption to access the Calendar Administrator, in order to protect sensitive information. Always use the `https://` URL prefix instead of `http://` to ensure secure access.

## Oracle Web Conferencing server

The Oracle HTTP server (OHS) allows clients and servers to authenticate over SSL using X.509v3 certificates. The certificates are stored in an Oracle Wallet, a container in which certificates and trusted certificates are stored and managed.

When communicating with the Web Conferencing server, the calendar server uses a secure HTTP connection (HTTPS) to OHS. To establish an HTTPS connection, a wallet is used with a default certificate which ensures that the information passed between the two servers is encrypted. The location of the wallet is defined by the `[CONFERENCING] walletfile` parameter in the `unison.ini` configuration file. The password of the wallet is defined by the `[CONFERENCING] walletpassword` parameter.

It is possible to replace the default certificates in the wallet with ones from another certificate authority. For more information on how to do this, consult the *Oracle9i Application Server Security Guide.*

The parameter `[CONFERENCING] url` defines the URL used by the calendar server to access the Web Conferencing server. By default Secure Sockets Layer (SSL) encryption is used and therefore the "`https://`" URL prefix is used rather than the "`http://`" to ensure secure access. The password and ID used to authenticate to the Web Conferencing server are defined by the parameters `[CONFERENCING] siteid` and `siteauthkey`.

For more information on the Web Conferencing parameters, consult Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual.*

# D

# International Support

This appendix contains information relating to international installations of the calendar server.

- UTF-8
- Character set identification
- Japanese server configuration
- Setting user language preferences

## UTF-8

The server uses UTF-8, an 8 bit encoding of 16 bit UNICODE, to achieve an international character representation. Data passed to the server must be converted from the source character set to UTF-8. When the data is read from the server, it once again passes through a conversion from UTF-8 to the character set defined on the display device. This functionality is crucial in heterogeneous environments where data may be entered in one character set and retrieved in another.

### Configuration

The `utf8_autoconvert` parameter in the `$ORACLE_HOME/ocal/misc/unison.ini` file controls the conversion and storage of calendar data in the UTF-8 format. By default, this feature is enabled. See Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual*, for information on the `utf8_autoconvert` parameter.

# Character set identification

In order to successfully complete the conversion to or from UTF-8, the server must know the source or destination character set. The character set used by the directory server is defined by the parameter `[LDAP] charset` in the `$ORACLE_HOME/ocal/misc/unison.ini` file (see Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual* for information on this parameter). Client character sets are identified explicitly to the server by the client application itself.

# Japanese server configuration

If the calendar server is installed on a Japanese operating system, the following parameters in the `$ORACLE_HOME/ocal/misc/unison.ini` file should be set as shown in the following example. In the `[ENG]` section, parameter `localcharset` defines the character set to use for data in log files. In the `[CWS]` section, parameter `mimecontentcharset` determines the default character set to use to encode the content and subject portion of all MIME mail messages sent by the CWS daemon/service. Parameter `mailhdrtoname` determines whether or not to include names along with addresses in the "To:" field of the mail header. See Appendix C, "Calendar Server Parameters" of the *Oracle Calendar Reference Manual* for more detailed explanations of the function of each parameter. Example:

```
[ENG]
localcharset= MSCP932
[CWS]
mimecontentcharset=ISO2022-JP
mailhdrtoname=FALSE
```

If older Japanese clients are used (versions prior to 9.0.4), the definition of the client character set can be set manually by inserting the following parameters in the `[LOCALE]` section of the `unison.ini` file: the parameter `charsetwindows` which indicates the character set used by the older Japanese Windows client, and the parameter `charsetCGI` which indicates the character set used by the older Japanese Web client. Example:

```
[LOCALE]
charsetwindows=MSCP932
charsetCGI=Shift_JIS
```

# Setting user language preferences

The calendar server offers users e-mail and Alert reminders in the language of their choice.

### Cmd line

Set a user's language through the `uniuser` utility with the "`language`" key. Consult Appendix E, "Utilities" of the *Oracle Calendar Reference Manual* for full information on use and syntax. The following example sets Herman Hesse's language to German:

```
uniuser -mod "S=Hesse/G=Herman" -m "language=de" -n 14
```

Alternately, you may use the default user profile file (`user.ini`) to set the `[GEN] language` parameter to specify a default language to apply to all new users to be created. Consult Chapter 8, "Calendar Users" of this guide and Appendix A, "User & Resource Parameters" of the *Oracle Calendar Reference Manual* for details on available languages and parameter values.

# Glossary

### ACE

Authentication, Compression and Encryption framework for the calendar server.

### agenda or calendar

The scheduling calendar of a user or resource.

### base DN

The location in the directory server tree under which all calendar users and resources are located.

### BindDN

Distinguished name used to authenticate to the Directory Server when performing an operation. See your Directory Server documentation for more information.

### Cluster

A node network in which one node, designated the “master node”, is responsible for on-line registration (via the web client) and network management.

### Calendar Administrator

A web-based tool for on-line calendar administration.

### Calendar daemons/ services

Six UNIX daemons or multi-threaded Windows NT services:

- Oracle Calendar Lock Manager (`unilckd`)
- Oracle Calendar Engine (`uniengd`)
- Oracle Calendar Synchronous Network Connections (`unisncd`)

- Oracle Calendar Corporate-Wide Services (`unicwsd`)
- Oracle Calendar Server Manager (`unicsmd`)
- Oracle Calendar Directory Access Server (`unidasd`) - external directory only

### Designate

A user who has been given the right to modify the calendar of another user or resource.

### Directory service

A database application designed to manage descriptive, attribute-based information about people and resources within an organization.

### Distinguished Name

String representation of an entry's name and location in an LDAP directory.

### Host

A computer running an installation of the calendar server or directory server.

### LDAP

Lightweight Directory Access Protocol, a directory service protocol designed to run over TCP/IP and across multiple platforms.

### LDIF

LDAP Data Interchange Format. Format used to represent Directory Server entries in text form.

### Local node

A calendar server node located on the host you are presently signed in to.

### Local resource

A resource whose calendar is maintained on the node you are presently signed in to.

### Local user

A user whose calendar is maintained on the node you are presently signed in to.

### Master node

A specially designated node in a cluster that finds user accounts on other nodes.

### Node

A calendar database containing all user and resource information and calendars.

### Node alias

A descriptive title that can be used in addition to the Node-ID to differentiate between multiple nodes.

### Node-ID

A unique identification number assigned when a node is created. Node-IDs must be unique across the enterprise.

### Node network

A series of two or more connected nodes.

### Object class

Defines an entry type in an LDAP directory by defining which attributes are contained in the entry.

### Oracle Internet Directory

A database application that stores information about users, and makes this information available to various other services, such as e-mail or calendar applications.

### Remote node

A calendar server node that is part of a node network but is not located on the node you are signed in to.

### Remote resource

A resource whose calendar is maintained on a another node.

### Remote user

A user whose calendar is maintained on a another node.

### Resource

An inanimate object, such as a conference room or a piece of equipment, that has its own calendar. When creating an event in their calendar, users can invite resources in the same way that they invite other users. Resources are managed by users who act as designates. Resources can also be used to create a calendar for tracking

related enterprise-wide information, such as company holidays or employees' travel schedules.

### Resource Relative DN

A location in the LDAP directory relative to the calendar server base DN in which all calendar server resources are stored.

### Schema definition

Describes the types of information that can be stored as entries in the LDAP directory.

### SYSOP

The administrator of a calendar server node.

### User

A person who uses a calendar client to connect to a node and manage his/her personal calendar.

### Web GUI

See Calendar Administrator

# Index

## A



## B



## C

## D



## E



## F



## G



## H



## I

## J



## K



## L



## M



## N

## O



## P



## R



## S

## T



## U



## W