

Oracle® Calendar Server

Administrator's Guide

Release 5.5

August, 2002

Part No. B10093-01

This Administrator's Guide describes deployment, administration and maintenance procedures for the Oracle Calendar server.

ORACLE®

Part No. B10093-01

Copyright © 1993, 2002, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xv
Preface	xvii
Intended Audience	xvii
Documentation Accessibility	xviii
Structure	xviii
Related Documents.....	xx
Conventions.....	xx
1 Architecture	
Overview	1
Hosts.....	1
Nodes.....	2
Clusters and master nodes	2
Daemons/services	2
Oracle Calendar Lock Manager.....	2
Oracle Calendar Engine.....	2
Oracle Calendar Synchronous Network Connection.....	3
Configuration.....	3
Oracle Calendar Corporate-Wide Services	3
Processing.....	4
Configuration.....	4
Oracle Calendar Directory Access Server	4

Calendar server architecture	4
Internal directory model.....	4
Initial client connection	5
Client requests	6
Reads and writes on the local node	6
Reads from a remote node	6
Writes to a remote node	6
Connections to nodes.....	7
Node request queue	7
External LDAP directory server model.....	7

2 Deployment

Deployment	1
Number of users	2
Acme Co. example	2
Logical divisions of users	3
Acme Co. example	3
Grouping users to create nodes.....	3
Acme Co. example	4
Product administration.....	6
Pre-installation checklist	6
User licenses	10
Installation notes	10
Distributed installs	11

3 Using Oracle Internet Directory

Benefits of using Oracle Internet Directory	1
Base DN	2
Access control	2
Directory server groups and group filters	3
Binding	4
Failover	4
Security	5
Changing the attribute used for SSO login	5

4	Setting Up Nodes	
	Creating a node	1
	Example	2
	Deleting a node	3
5	Node Networks	
	Connecting nodes	2
	NT	3
	UNIX	4
	Syntax	5
	Connections and Rules	6
	Adding a node to the network	7
	Deleting a node from the network	7
	Increasing or decreasing the number of connections between nodes	7
	Tuning connections	7
	Setting up a master node	9
	Moving a node	9
	Big-endian vs. little-endian hosts	9
	Versions of the calendar servers	10
	The node network	10
	Directory server involved	10
	Performing the move	11
	Coexistence of LDAP and non-LDAP nodes	13
	Caveat: LDIF differences between UNIX and NT	14
	Coexistence of nodes with and without Oracle9iAS Wireless	14
6	Server Administration	
	Starting and stopping the calendar server	1
	Example	2
	Example	2
	Checking server status	2
	Example	2
	Viewing current user activity	2
	Example	2

Changing the SYSOP (node) password	3
Example	3
Other administrative tasks	3

7 Users and Groups

Creating calendar users	2
Adding users to the internal calendar server directory	2
Adding users from a pre-populated directory server	2
Determine which users on the directory server have not yet been added to a calendar server node: 3	
To add users one at a time:	3
Adding calendar users to a directory server	4
Managing calendar users	4
Deleting calendar users	5
Moving calendar users	6
Managing user defaults	6
Managing groups	7
Private groups	7
Members-only groups	7
Public groups	7
Administrative groups	8
Assigning administration rights	8
Setting up e-mail notification	8
Other user configuration options	9
Global and published calendars	9

8 Alerts

Filtering e-mail alerts	2
Reminders	3
Types	3
Format	3
Controlling reminder behaviour	3
CWS operation	3
Old reminders	3
Notification	4

Types	4
Format	4
Controlling notification behaviour	4
Disabling e-mail notification	4
Limiting the number of recipients	4
Setting up wireless services	4
9 Resources, Event Calendars and Holidays	
Managing holidays	1
Creating event calendars	2
Managing and deleting event calendars	3
Creating resources	3
Adding new resources to the calendar server's internal directory	4
To add a resource:	4
Adding new resources in an LDAP directory context	4
To add a resource:	5
Managing resources	6
Deleting resources	6
Managing resource defaults	7
Assigning designates	8
10 Node Maintenance	
Maintenance procedures	1
Daily monitoring procedures.....	1
Daily maintenance procedures.....	2
Monthly maintenance procedures	2
Other maintenance procedures	3
Back up and restore	3
A Disk Space and Memory	
Database disk space requirements	1
Remote NFS storage	3
Run-time requirements	3

11	Monitoring Procedures	
	Viewing log files	1
	Interpreting log files	2
B	Sizing Guidelines	
	Memory and disk requirements.....	1
C	System Configuration	
	Adjusting kernel parameters	1
	Adjusting the HP-UX kernel parameters	4
	Adjusting the Solaris kernel parameters.....	6
	Adjusting the AIX kernel parameters	8
	Adjusting the Tru64 kernel parameters	10
	Adjusting the Windows kernel	12
	Using operating system clusters	12
	Limitations.....	12
	Additional reading	13
D	Security	
	ACE framework.....	1
	Secure connections to clients and other calendar servers.....	2
	Secure connections to clients.....	2
	Secure connections to another calendar server	3
	Configuration	4
	Extending the ACE framework	5
	Plug-ins	5
	Extending the set of plug-ins	6
	Web authentication plug-in.....	7
	How does the Web authentication plug-in work?.....	8
	Configuring your calendar server	8
	Configuring your Web calendar client	9
	Directory server security.....	11
	Other security considerations.....	11
	Dedicated Server.....	11

Password Management	11
Trust Management	12
Networking	12
Auditing.....	12
Backup and Recovery	13
Application Security	13
Calendar Administrator	13

E International Support

UTF-8	1
Configuration.....	1
Character set identification	2
Japanese server configuration	2
Setting user language preferences	3

F The Calendar Administrator

Web server configuration	1
Calendar Administrator customization	2
By parameters	2
By stylesheet	2
By templates	2
Calendar Administrator configuration	2
Directory for HTML template files	3
Directory for temporary sessions files.....	4
Directory for images	4
List of Calendar Domain Service hosts — disregarded.....	5
CGI or FCGI operation	5
SYSOP password — disregarded.....	6
Old SYSOP password — disregarded.....	6
Site name.....	7
Site logo.....	7
Domain logo.....	8
Support e-mail	8
User management	9
Resource management.....	9

Holiday management	10
Event calendar management.....	10
Admin password management	11

Glossary

Index

List of Figures

1-1	Calendar server internal architecture.....	5
1-2	Calendar server internal architecture with external directory server	8
D-1	ACE framework architecture.....	5

List of Tables

2-1	Acme Company: user base.....	2
2-2	Acme Company: geographic and administrative user divisions.....	3
2-3	Acme Company: node distribution.....	5
2-4	Installation information checklist: internal directory.....	8
2-5	Installation information checklist: external LDAP directory.....	9
4-1	Recommended node-ID ranges.....	1
5-1	Oracle Calendar server 5.4 platforms.....	10
A-1	Variable definitions.....	1
11-1	Calendar server log files.....	1
C-1	Calculating new kernel parameter values.....	2
C-2	Calculating new kernel parameter values with CWS enabled.....	2
C-3	Calendar server message and semaphore requirements.....	3
C-4	HP-UX kernel parameters.....	4
C-5	HP-UX kernel parameters (example).....	5
C-6	Solaris Kernel Parameters.....	7
C-7	Solaris kernel parameters (example).....	8
C-8	AIX kernel parameters.....	9
C-9	Tru64 kernel parameters.....	10
C-10	Tru64 kernel parameters (example).....	11
D-1	ACE configuration parameters.....	4
F-1	uniwebadm.ini configuration parameters.....	3

Send Us Your Comments

Oracle Calendar Server Administrator's Guide, Release 5.5

Part No. B10093-01

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 and resources. Among other features, it offers a dedicated database, 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, Motif), 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* documents deployment, configuration and maintenance procedures for your calendar server. This guide is directed at any administrator whose task is the installation, configuration, use and maintenance of Oracle Calendar in general and the Oracle Calendar server in particular. It is a companion volume to the *Oracle Calendar Server Reference Manual*, which provides detailed information concerning configuration parameters and server administration tools.

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 eleven chapters, five appendices and a glossary:

Chapter 1

This chapter contains a basic overview of the calendar server's internal architecture.

Chapter 2

This chapter contains an overview of deployment strategies for your calendar services.

Chapter 3

This chapter contains information on using the calendar server in combination with Oracle Internet Directory.

Chapter 6

This chapter contains instructions on a variety of basic administration tasks such as starting and stopping the server.

Chapter 4

This chapter contains a discussion of calendar server databases, or "nodes".

Chapter 5

This chapter contains instructions for connecting multiple nodes together into a node network.

Chapter 7

This chapter contains information on users and user management.

Chapter 9

This chapter contains information on creating and managing calendar resources, event calendars and holidays.

Chapter 8

This chapter contains an overview of the alert services offered by the calendar server through e-mail and Oracle 9iAS Wireless.

Chapter 10

This chapter contains a discussion of maintenance procedures for the calendar server, including instructions for backing up and restoring the calendar database.

Chapter 11

This chapter contains a basic look at monitoring the operation of the calendar server through log files and utilities.

Appendix A

This appendix contains some general information on sizing and scaling points to assist in planning your deployment.

Appendix B

This appendix contains additional information on sizing, including memory usage.

Appendix C

This appendix contains some specific information about tuning operating systems to increase the efficiency and effectiveness of the calendar server.

Appendix D

This appendix contains a variety of information about the security of your calendar services.

Appendix E

This appendix describes some issues concerning deploying your calendar server in an international environment.

Appendix F

This appendix contains general information regarding the Calendar Administrator, an on-line tool for administering the calendar server.

Glossary

Definitions of a number of terms used in Oracle Calendar documentation.

Related Documents

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

- *Oracle Calendar Server Reference Manual*
- *Oracle Calendar Server Release Notes*

Conventions

In this manual, Windows and NT are both used to refer to the Windows95, Windows98, 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
. . .	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.
Web GUI	Refers to a procedure executed using the Calendar Administrator, an on-line administrative tool. Consult Appendix F, "The Calendar Administrator" , for more information
/	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.

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.

Note that the Oracle Calendar server included with Oracle Collaboration Suite requires the use of Oracle Internet Directory. Therefore, for any areas of the Oracle Calendar server documentation that describe differences between "internal-directory" and "external-directory" installations, the "external-directory" information can be assumed to apply to your installation.

- [Overview](#)
- [Daemons/services](#)
- [Calendar server architecture](#)

Overview

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

Hosts

A host is the physical machine running an installation of the calendar server. Only one installation of the server may be present on any one host, but that server may contain multiple nodes.

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 on the same server. Nodes may also be connected into a node network, allowing users on separate servers to schedule meetings and events transparently with one another.

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.

Daemons/services

The calendar server contains four UNIX daemons/multi-threaded Windows NT services, with an extra daemon/service added for implementations connected to an external LDAP directory server. The calendar server daemons/services are:

- [Oracle Calendar Lock Manager](#)
- [Oracle Calendar Engine](#)
- [Oracle Calendar Synchronous Network Connection](#)
- [Oracle Calendar Corporate-Wide Services](#)
- [Oracle Calendar Directory Access Server](#) (with external directory server only)

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.

Oracle Calendar Engine

The Engine (`uniengd`) accepts and services client requests. Since many of these requests can be received at the same time, the servicing Engines rely on the Lock Manager to ensure orderly access to local nodes. 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 Synchronous Network Connection

The Synchronous Network Connection daemon/service (`unisnkd`) fulfills two roles in the server architecture. First, the `unisnkd` 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.

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

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.

Configuration

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

Oracle Calendar Corporate-Wide Services

The Corporate-Wide Services (`unicwsd`) daemon/service allows the exchange of information between users who reside on different nodes. People on one node may need to view calendar information or schedule events with users from another node. This service also provides e-mail notification through an SMTP mail server and wireless notification through Oracle's wireless services.

`unicwsd` communicates with other nodes using TCP/IP sockets and named pipe connections. These links provide fast communication and instant free-time lookups when scheduling an event. The TCP/IP sockets and named pipe connections are provided and managed by the Synchronous Network Connection daemon/service (`unisnkd`).

Processing

The Corporate-Wide Services daemon processes local requests for remote information as well as requests from other nodes for local information. Local requests are requests for information concerning users on another node, or to send mail messages to remote users. These requests are made by the users on the local node and are put in a local queue. The Corporate-Wide Services daemon/service processes the requests by reading them one after another from the queue. If a TCP/IP connection is available, the information requested by the local user can be accessed by connecting directly to the other node.

The only mail handling `unicwsd` performs is the sending of notifications when a user creates an event and the attendees need to be notified. These mail notifications are dependent on user-controlled options.

Configuration

For technical information concerning the configuration of the Corporate-Wide Services daemon/service, including a list of parameters and their default values, see the calendar server Reference Manual, Appendix B, "Server Parameters." Unless otherwise indicated, the Corporate-Wide Services must be re-started 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 (`unisnkd`) daemon/service.

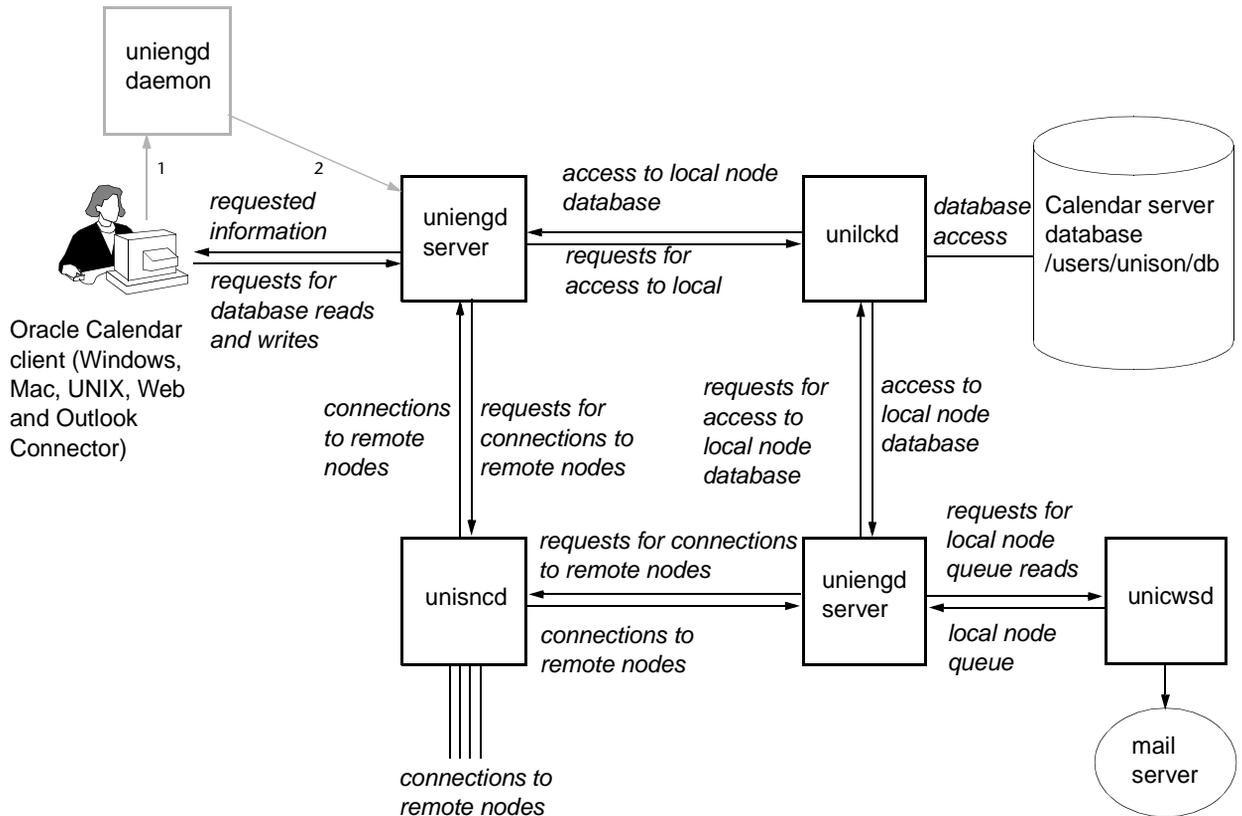
Calendar server architecture

Two configurations are presented here. One uses the calendar server's internal directory, and the other uses an external LDAP directory server.

Internal directory model

In this configuration, the calendar server uses its internal directory instead of connecting to an LDAP directory server, and does not use the `unidasd` daemon/service. The node network may or may not include other calendar servers. For simplicity, a single client session is represented.

Figure 1-1 Calendar server internal architecture



Initial client connection

The client first connects to the **uniengd daemon/service** (1). The **uniengd daemon/service** then spawns a **uniengd server** for that client session (2). All subsequent requests from the same client are sent to this dedicated **uniengd server**. 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 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` server 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` server requests access to the node database (`/users/unison/db/<node name>`) from the `unilckd`. When the `unilckd` grants access, the `uniengd` 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` server requests a connection to the remote node from the `unisncd`. When the `unisncd` hands it a connection to that node, the client's `uniengd` sends the read request to the `uniengd` server at the other end of the connection (see "[Connections to nodes](#)" below). 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` server. The requesting `uniengd` server 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` server places a request in the queue of the local node (see "[Node request queue](#)" below for a discussion of the node queue). Another `uniengd` server associated with this local node services these requests in the same manner as reads on a remote node.

Connections to nodes

`unisnkd` establishes persistent connections to other nodes on startup, using information in the `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` spawns a `uniengd` server to handle requests to the specified node. A `uniengd` server sits on the end of each connection to a remote node. This diagram does not show the `uniengd` server at the end of each connection, nor does it show that each of these `uniengd` servers obtains access to their node database from `unilckd`.

Node request queue

Although not shown on this diagram, each node has a request queue associated with it. This request queue is maintained within the node database and used by all `uniengd` servers associated with the node. These request queues contain requests for SMS 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` server for each node on the local machine. The `uniengd daemon/service` then spawns one `uniengd` server per node to service that node's request queue. The diagram shows one of these `uniengd` servers.

`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 Oracle9iAS Wireless. If the request is for a write to a remote node, it hands the request to the `uniengd` server servicing the request queue of the node. `unicwsd` also deletes redundant requests in order to optimize performance.

External LDAP directory server model

This configuration is the same as the previous one except that all calendar servers now use an LDAP directory server for user attributes. All nodes in a node network that use a directory server must use the same one.

The notable addition here is the `unidasd` server, which handles all interactions with the LDAP directory server.

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.

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 2–1 *Acme Company: user base*

User category	Estimates
Configured users	16,000
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 2–2 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

With the logical divisions among your user base clearly delineated, you are now ready to group your users into nodes, calendar databases containing agendas and information for users and resources. 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 desktop calendar clients (Windows, Mac, Motif) and Oracle Outlook Connector, 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"](#), and [Appendix B, "Sizing Guidelines"](#), 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 the calendar server Reference Manual, Appendix C, "Utilities."

- **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, designates and members-only groups, are specific to each node. The tasks associated with these features, such as adding holidays or assigning designates, 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 above 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 the calendar server Reference Manual, Appendix B, "Server Parameters").
- The final two nodes will be located on one server in Seattle. All Seattle users will be on one node, and all Vancouver users will be on the other node. Although there is no performance advantage in splitting the users between two nodes on a single server, 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 2–3 Acme Company: node distribution

Node	Server Location	Server Number	User Base
Node 1	Los Angeles	1	LA: Engineering Division (7,000)
Node 2	Los Angeles	2	LA: Administration Division (5,000)
Node 3	New York	3	NY: Marketing (600) and Administration (400) Divisions Chicago: Marketing Division (500)
Node 4	Seattle	4	Seattle: Engineering (1,500) and Marketing (500) Divisions
Node 5	Seattle	4	Vancouver: Marketing Division (500)

See [Appendix B, "Sizing Guidelines"](#), 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 and resources
- resource administration (assigning designates)
- holiday administration
- front-line support
- client training

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 `/users/unison/misc/unison.ini` file. For more information on these parameters, see the calendar server Reference Manual, Appendix B, "Server Parameters."

- **Kernel parameters**

Operating system kernel parameters must be evaluated, and if necessary tuned, for each installation. Refer to the calendar server Reference Manual, Appendix C, "Utilities," 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 `[LCK] lck_users` 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.

- **Mail notification**

The [LIMITS] mail parameter enables or disables e-mail notification for all event or task creation. The default value enables this feature.

- **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] resourcerelatedn parameter for details.

- **Attachments**

The [LIMITS] allowattachments parameter enables or disables the attachment of files to events or tasks created by clients. The default value disables this feature. 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, either individually or in batch mode using a default user profile. See "[Assigning administration rights](#)" in [Chapter 7, "Users and Groups"](#), for a discussion of the differences between the group types and the methods used to change default administration rights.

- **SYSOP (node) password**

When you install the calendar server in internal directory mode, a SYSOP (node) password will not be assigned to the node created during the installation process. Sign in to the node to set a password before adding users. See "[Changing the SYSOP \(node\) password](#)" in [Chapter 6, "Server Administration"](#), for instructions on this process.

- **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; 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, or they may be set up to allow conflicts to

occur. The default value for the [ENG] `allowresourceconflict` parameter prohibits double-bookings.

The following table presents a list of the major items to consider before installing the calendar server in internal directory mode:

Table 2–4 Installation information checklist: internal directory

Item	Accepted Values	Mandatory or Optional	Default Value
Node-ID	Recommended Node-ID ranges: 1-5: Evaluation 6-100: Test 101-9999: Permanent 10000-59999: Future use 60000+: Reserved (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 the calendar server Reference Manual, Appendix D, "Time Zone Table."	Mandatory	N/A
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

The following table presents a list of the major items to consider before installing the calendar server to connect to an external LDAP directory.

Table 2–5 Installation information checklist: external LDAP directory

Item	Accepted Values	Mandatory or Optional	Default Value
Node-ID	Recommended Node-ID ranges: 1-5: Evaluation 6-100: Test 101-9999: Permanent 10000-59999: Future use 60000+: Reserved (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
Node (SYSOP) Password	Up to 15 alphanumeric characters in length	Mandatory	N/A
Time Zone	See the calendar server Reference Manual, Appendix D, "Time Zone Table."	Mandatory	N/A
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

Table 2–5 Installation information checklist: external LDAP directory

Item	Accepted Values	Mandatory or Optional	Default Value
Mail Host	Any host	Mandatory if mail notification enabled	local host
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, <code>unison.ini</code> value; otherwise N/A
Calendar server Administrators' Group DN	A new group created under the base DN	Mandatory	If present, <code>unison.ini</code> 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 listed in the Oracle Collaboration Suite *Installation Guide*.

Distributed installs

Note that each time you install Oracle Calendar, the server, Oracle Calendar Administrator, Oracle Calendar Web calendar client and Oracle Calendar API are all installed on the chosen host. 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.

- 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 Oracle Collaboration Suite *Administrator's Guide* for details.
- 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 5, "Node Networks"](#), for details.
- Consult the Oracle Calendar Web client *Administrator's Guide* for other issues that may be involved, such as ensuring that the client points to the master node of your calendar server installation.

Using Oracle Internet Directory

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 Oracle Calendar server, and a discussion of some more specific configuration issues.

- [Benefits of using Oracle Internet Directory](#)
- [Access control](#)
- [Directory server groups and group filters](#)
- [Binding](#)
- [Failover](#)
- [Security](#)
- [Changing the attribute used for SSO login](#)

Benefits of using Oracle Internet Directory

Oracle Internet Directory is a database application that stores information about users, and makes 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 instead of many. 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 such as `/users/unison/misc/unison.ini` specifies the location of a particular item in a file system.

Directory servers have schemas that define the information they store. These schemas consist 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 an 'e-mail 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 will be preset with all the objects and attributes needed by the calendar server.

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.

Access control

Using calendar clients and server tools, both calendar users and administrators have the ability to modify entries in the LDAP directory. The level of access they have depends upon access the control restrictions set on your directory server. By default, calendar users can modify any of their own attributes, and calendar administrators (SYSOPs) have full control over all LDAP attributes under the calendar base DN.

Calendar users are created by adding an object class (ctCalUser) and a number of calendar attributes (identified by the "ctcal" prefix) to existing users. They bind to the directory server as themselves, using the self entry modification access control profile (ldap:///self), which on most directory servers grants full access to modify all attributes of that user. This includes both calendar-specific attributes and other user attributes such as e-mail address and mobile phone number.

Calendar administrators are added to the directory server as members of the group specified by the `unison.ini` parameter `[LDAP] admingroup`. When you install

the calendar server, a default access control profile is created for members of this group which allows access to all LDAP attributes 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 administrator and user 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; for detailed information on your calendar server's minimum requirements for user and administrator access controls contact Oracle support.

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.

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 (i.e. users with the `ctCalUser` object class), if a given LDAP group has only non-calendar users as members, that group will be listed as empty, with no members at all.

To prevent calendar clients from listing such groups, you can configure the search filter the calendar server uses when looking for groups on the directory server. To do this, use the `unison.ini` parameter `[LDAP] groupfilter`. By default, the value of this parameter retrieves all groups. You can restrict it by applying a new object class to all groups you want to be listed, and adding this object class to the `groupfilter` parameter.

For example, consider the following two groups:

group 1:

```
dn: cn=Marketing Group, o=acme
objectclass: top
objectclass: groupOfUniqueNames
cn: Marketing Group
description: Marketing mailing-list group
uniqueMember: cn=Beverly Cleary,o=acme
uniqueMember: cn=Friedrich Schiller,o=acme
uniqueMember: cn=Philip MacDonald,o=acme
```

group 2:

```
dn: cn=Finance Group, o=acme
objectclass: top
objectclass: groupOfUniqueNames
cn: Finance Group
description: Finance mailing-list group
uniqueMember: cn=Ambrose Bierce,o=acme
uniqueMember: cn=Willa Cather,o=acme
uniqueMember: cn=Jamaica Kincaid,o=acme
```

If the Finance Group were composed only of non-calendar users, you could prevent it from being listed in calendar clients by adding an object class such as:

```
objectclass: CalendarGroup
```

to the Marketing Group definition, and changing the value of the [LDAP] `groupfilter` parameter to:

```
"(&(objectclass=groupOfUniqueNames)(objectClass=CalendarGroup)(uniqueMember=*))"
```

Binding

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 `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 item 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 the calendar server Reference Manual, Appendix C, “Utilities.”

Failover

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. Consult the documentation on the [LDAP] `host` parameter in the calendar server Reference Manual, Appendix B, "Server Parameters," for full details.

Note: After switching to a secondary directory server, the calendar server will not attempt to reconnect to the primary directory server in the event that it becomes available again. To reconnect to the primary directory server, stop and restart your calendar server, or, at a minimum, stop and restart the `unidasd` daemon using the `unistart` and `unistop` utilities with the `-das` option, available on UNIX platforms only.

Security

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.

Changing the attribute used for SSO login

If you change the attribute your Oracle Internet Directory uses to authenticate SSO logins at any time 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 `/users/unison/misc/unison.ini` file on each calendar server host, 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.

Setting Up Nodes

A node is a database containing agendas and information for users and resources. 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 is commonly encountered when a group of users requires a different time zone, or when there is a logical division that the administrator wants to maintain within a group of users in the same time zone.

This chapter will cover the following topics:

- [Creating a node](#)
- [Deleting a node](#)

Creating a 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 make it easier to tell nodes apart. A SYSOP (node administrator) password restricts access to the calendar account used for all node management tasks. Each node has a default time zone.

- **Node-ID:** Oracle's recommended node-ID ranges are:

Table 4–1 Recommended node-ID ranges

Node-ID	Purpose
1-5	Evaluation Node-IDs
6-100	Test Node-IDs
101-9999	Permanent Node-IDs
10000-59999	Future Use

Table 4–1 Recommended node-ID ranges

Node-ID	Purpose
60000+	Reserved

When setting up a node, it is important to bear in mind 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 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 the calendar server Reference Manual, Appendix D, "Time Zone Table," for a complete list of countries with their corresponding time zone notation.
- **SYSOP Password:** The password for the SYSOP, or node administrator, can be any string of up to 15 alphanumeric characters in length.
- **Directory Manager Password:** The password for the SuperUserDN, required only for installations using an external directory server.

To create a node:

1. Use the `unistop` utility introduced in [Chapter 6, "Server Administration"](#), to bring down the calendar server. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities." 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, see the calendar server Reference Manual, Appendix C, "Utilities."

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

```
% uniaddnode -n 144 -a publications
Please enter Sysop password:
Please enter Sysop password again:
unidsndini: working, please wait ...
```

```
uniaddnode: unidsndini done
uniaddnode: Database initialization done
```

An entry similar to the following would now exist in the `/users/unison/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 6, "Server Administration"](#), to restart the calendar server. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Deleting a 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 ["Back up and restore"](#) in [Chapter 10, "Node Maintenance"](#), for more information.
2. Remove the node from the node network (if it is part of one) by editing the `/users/unison/misc/nodes.ini` file and applying the change. Understand the contents of [Chapter 5, "Node Networks"](#), before attempting to do this.
3. If you are running a directory server, delete all users and resources on the node (using `uniuser -ex` and `unires -ex` respectively) from the directory server.
4. Shut down the calendar server.
5. Delete the entire `/users/unison/db/nodes/<Nx>` directory, where `<Nx>` is the value of the `name` parameter in the appropriate node section of the `/users/unison/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 . See the calendar server Reference Manual, Appendix B, "Server Parameters," for details on `unison.ini` parameters.

6. Delete the corresponding node section in the `/users/unison/misc/unison.ini` file. For example, if you are deleting the node with node-ID 144, delete the [144] section of `unison.ini`.

Steps 7-11 are necessary only for calendar server installations using an external LDAP directory server.

7. Use `ldapmodify` to delete all references to reserved calendar users for the node scheduled for deletion.
8. Generate an LDIF file from the contents of your LDAP directory server.
9. Edit the LDIF file to delete all references to reserved calendar users for the node scheduled for deletion.
10. Import the modified LDIF file back into your directory server.
11. Restart the calendar server.

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 the node network.** Two nodes with the same node-ID cannot be connected in a network.

This chapter details the following node network management:

- [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 (`/users/unison/misc/nodes.ini`), managed using the `uninode` utility or the Admin GUI. 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.

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 (see Appendices A, B and C)
- clients used are not Web-based (i.e. Windows, Macintosh or Motif clients, or Oracle Outlook Connector)
- 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
- user/resource retrievals are from the internal cache (the `[ENG] dac_itemget` parameter is set to `FALSE` in the `/users/unison/misc/unison.ini` file)

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. Contact Oracle support for help tuning the number of connections.

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

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 `/users/unison/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 "/users/unison/misc/nodes.ini" file
initialization succeeded
```

3. The newly created `/users/unison/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"](#) on page 5-5 and ["Connections and Rules"](#) on page 5-6 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](#)" on page 5-5 and "[Connections and Rules](#)" on page 5-6 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 `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 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 above 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 line above.

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 the calendar server Reference Manual, Appendix B, "Server Parameters," for an explanation of each of these.

Contact Oracle support for assistance in determining appropriate values for these parameters.

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 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 or alias 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](#)
- [Directory server involved](#)

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 utility. Contact Oracle support for more information

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

Table 5–1 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 a database conversion utility. Contact Oracle support for more information on the `unidbconv` 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 Admin GUI, removing a node from a node network using either of these tools results in data loss. Namely, any event 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](#)" below for the exact procedure to follow in each case.

Also recall that if you are running a directory server, all nodes in a node network must point at the same directory server.

Directory server involved

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 an internal directory or both using a directory server

The first procedure describes moving a node within a node network, and the second describes moving a node between node networks. 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 `/users/unison/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 `/users/unison/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 `/users/unison/misc/unison.ini` file on the old host and add it to the `/users/unison/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 `/users/unison/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 `/users/unison/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 `/users/unison/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 `/users/unison/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 `/users/unison/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 `/users/unison/db/nodes/<Nx>` directory now exists on the new host.
6. Remove the section configuring the node from the `/users/unison/misc/unison.ini` file on the old host and add it to the `/users/unison/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 `/users/unison/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. If this is not your situation, contact Oracle support for assistance.

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 "[Back up and restore](#)" in [Chapter 10, "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. It is strongly recommended that you consult Oracle support before changing this parameter.

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 Oracle9iAS 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 is set up with Oracle9iAS 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 Oracle9iAS 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.

Server Administration

This chapter describes how to manage your calendar server. All administrative tasks and procedures are executed using command-line utilities, the Admin GUI, the Web GUI, or operating system tools.

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. 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 procedures:

- [Starting and stopping the calendar server](#)
- [Checking server status](#)
- [Viewing current user activity](#)
- [Changing the SYSOP \(node\) password](#)
- [Other administrative tasks](#)

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 (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, see the calendar server Reference Manual, Appendix C, "Utilities."

Example To start the calendar server (bringing up four daemons if using an internal directory and five 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 -y
```

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, see the calendar server Reference Manual, Appendix C, "Utilities."

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
unison 26512    1 17:18:14 0:00  unilckd  daemon
unison 26517    1 17:18:16 0:00  uniengd  daemon
unison 26591    1 17:18:22 0:00  unidasd  daemon
unison 26669    1 17:18:24 0:00  unisncd  daemon
unison 26678    1 17:18:26 0:00  unicwsd  daemon
```

```
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 the calendar server Reference Manual, Appendix C, "Utilities."

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

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, see the calendar server Reference Manual, Appendix C, "Utilities."

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: A node created during the installation of the internal-directory version of the calendar server will not have a SYSOP password. The password must be set once the node is created. However, a SYSOP password must be set for a node created as part of the calendar server installation when using a directory server.

Other administrative tasks

Use command-line utilities for a variety of other less frequent tasks. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Some of the utilities that you may want to be familiar with once the calendar server is up and running are:

- `UNIREQDUMP` — to view and delete requests in the Corporate-Wide Services (CWS) queue.
- `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.
- **UNICLR_IPC (UNIX only)** — to clear Inter-Process Communication (IPC) resources consumed by the calendar server.

See the calendar server Reference Manual, Appendix C, "Utilities," for a description of the function and syntax of all utilities included with the calendar server.

Users and Groups

This chapter describes the different tasks involved in managing users and groups within a calendar server node. Whether user and group 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.

Note that user passwords can not be set or changed using calendar server administration tools. Take advantage of the other directory management tools provided to you for password management.

The following topics are dealt with in this chapter:

- [Creating calendar users](#)
 - Internal directory context
 - LDAP directory context
- [Managing calendar users](#)
- [Deleting calendar users](#)
- [Managing user defaults](#)
- [Managing groups](#)
- [Assigning administration rights](#)
- [Setting up e-mail notification](#)
- [Other user configuration options](#)

Creating calendar users

Each person who plans to use calendar services must have a profile on the server. Once a user's profile has been created and added to a node, that person can then use a calendar client (Windows, Macintosh, Motif, Outlook, Web or wireless) to connect to the server and manage his/her personal agenda.

Adding users to the internal calendar server directory

Web GUI

Use the Calendar Administrator to add users to a node. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

Use the `uniuser` utility to add users to the calendar server's internal directory. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Example

```
uniuser -add "S=Addison/G=Thomas/I=W/O=acme" -n 786 -p <SYSOP password>
uniuser: added "Addison,Thomas,W"
```

Adding users from a pre-populated directory server

An administrator is presented with two possible scenarios when adding calendar services to an installation that already uses an LDAP directory service. In the most common situation, the database of users will already exist on the directory server. This data is then used to create the users on the newly created calendar server nodes.

Web GUI

Creating calendar accounts for existing directory users can be done quickly and simply through the Calendar Administrator. Sign in to the node on which you want to create your users, select **Users** from the main screen, and **Create calendar accounts** on the user management page.

You will then be able to search the directory server, using your choice of search filters, for any users who do not already have calendar accounts. Select them in the list of search results and click **Create**.

Cmd line

Adding calendar users from an existing directory server 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 the calendar server Reference Manual, Appendix C, "Utilities."

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 the calendar server Reference Manual, Appendix C, "Utilities."

```
% 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`. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

```
% uniuser -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 `users`.

```
% unidssearch -c 5 > users
```

2. The file created may then be modified, filtered or added to as required and according to a set format and syntax. For example, if you are using Control

Data's Global Directory Server, you would be advised to add an attribute and value for the user's given name, as this is used by the calendar server and is not included in the directory schema. Additions are made in X.400 format. For a complete description of the X.400 keys, fields and syntax, see the `uniuser` documentation in the calendar server Reference Manual, Appendix C, "Utilities."

```
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 "users" file to the specified node.

```
% uniuser -ex users -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 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

Web GUI

Use the Calendar Administrator to view and modify a user's calendar attributes easily. To modify attributes that are not calendar-specific, use your Oracle Internet Directory administration tools. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

You may view and modify a user's calendar attributes using the `uniuser` utility. To modify attributes that are not calendar-specific, use your Oracle Internet Directory administration tools. For full information on the use and syntax of `uniuser`, see the calendar server Reference Manual, Appendix C, "Utilities."

Deleting calendar users

Web GUI

Use the Calendar Administrator to delete users easily from the server. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

Remove the user(s) from the calendar server node using the `uniuser -del` (single deletion) or `uniuser -ex` (multiple deletions) commands. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

For installations using an external directory, delete the same user(s) from the directory server or run `unidsdiff` to synchronize the information on the calendar server node with that kept in the directory server.

Warning: When a user is deleted from a node, the user's records and directory entry are removed from the local node. This means that all data owned by the user, including any events or groups, will be deleted. The user will no longer appear in others' agendas, nor will any events owned by this user remain. Any remote directory listings and remote copies of events owned by the user will also be removed. To preserve the user's agenda in a file prior to deletion from the node, you may use the `unicpoutu` utility. Use the `unicpinu` utility to copy this information back into a calendar server node.

If you do not want to run the risk of deleting valid events from the agendas of other users, it is a good practice to not delete, but to rename, the user until all of the events have passed. For example, if a manager who controlled group scheduling leaves the company, you might delete all personal information from his/her user

profile, change the password, and enter "manager" and "sales" for last and first name respectively. All events and groups owned by this user would therefore remain in the agendas of other users. Alternatively, you may wish to change all of the personal information in the manager's user profile to that of a new employee who assumes the same function and therefore takes over the management of the created events and groups.

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. For full information on use and syntax, including a variety of crucial warnings and considerations, see the calendar server Reference Manual, Appendix C, "Utilities."

```
% unimvuser -u "ID=56" -host1 scribe -host2 hoth -n1 15 -n2 2005
```

Always use the most recent version of `unimvuser` in your node network.

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 `/users/unison/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 the calendar server Reference Manual, Appendix A, "User and Resource Parameters."
- 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 `/users/unison/misc/user.ini` file.

Applying a default user profile:

- The profile is applied during user creation (using the `uniuser` utility, the Admin GUI or the Web GUI).
- The default user profile is outlined under the section heading `[GEN]` in the `/users/unison/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. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Managing groups

Four different types of groups allow users to schedule entries, tasks, events and notes efficiently with other users and resources:

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
- can include members on remote nodes, but those members may not use 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 rights to do so by the calendar server administrator

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 rights to do so by the calendar server administrator

Assigning administration rights

You may grant to or revoke from individual users the right to administer groups and holidays. A default administrative rights profile is assigned to each new user according to the parameters set in the `/users/unison/misc/user.ini` file. You may wish to initially assign no rights to administer holidays and groups (the current default setting in the `user.ini` file), and then selectively grant these rights. Alternatively, you may wish to define a default profile and then use it as a template to add all users requiring the same administrative privileges.

Cmd line

Users' administrative rights can be set and modified from the command line using the `uniadmrights` utility. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Setting up e-mail notification

The calendar server stores users' e-mail addresses, allowing users to notify each other of created, modified or deleted entries.

To set up e-mail notification:

1. Choose an X.400 address field in which to store e-mail addresses. For example: "OU1", "O", "A", etc. Be aware, however, that these fields have varying limitations on the number of available characters.
2. Set the value of the `[ENG] usermailmap` paramter in `unison.ini` to the key of the X.400 field you chose. The default value is "O".

When you add users, you may now specify their e-mail addresses.

Cmd line

Use the EMAIL key/value pair.

When adding a user, the value you specify for the EMAIL key will be stored in the field specified by [ENG] usermailmap. 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 the calendar server Reference Manual, Appendix C, "Utilities." Please note that the field specified by [ENG] usermailmap is disabled, and can only be set through the EMAIL key. For example, if usermailmap were set to FAX:

```
% uniuser -add "S=Kafka/G=Franz/FAX=fkafka@mail.org" -n 23
uniuser: modification of "Fax phone number" has been disabled
uniuser: ignoring "FAX=fkafka@mail.org"
uniuser: added "Kafka, Franz"
```

The EMAIL key-value pair may be used wherever the <user> argument is specified, but no -format parameter exists for it.

Other user configuration options

Global and published calendars

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 set to 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 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. In addition, this attribute allows the setting EVENTCALENDAR, which is equivalent to a published calendar, but in Web clients causes the current account to appear in the list of published event calendars rather than the list of published user calendars. The default setting for this attribute is PUBLISHED.

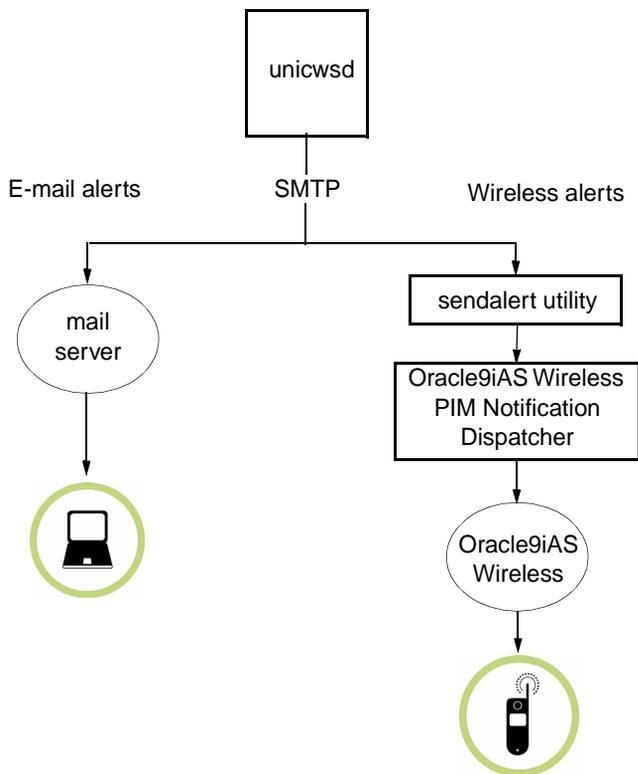
You can set the Published Type and Global Read Access attributes using the Calendar Administrator or the `uniuser` utility.

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 Oracle9iAS Wireless, which provides reminders and notifications through a number of channels including voice mail, FAX and Short Message Service (SMS).

- [Reminders](#)
- [Notification](#)
- [Setting up wireless services](#)

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 Oracle9iAS 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 Oracle9iAS Wireless PIM Notification Dispatcher.



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 Oracle9iAS Wireless.

Format

The format of the reminder delivered by the calendar server is determined by template files installed in `/users/unison/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 all must be stored in the UTF-8 character set.

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 `unison.ini [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.

Notification

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 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.

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

You can disable e-mail notification entirely in all 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 `/users/unison/misc/unison.ini` file on each host.
3. Set the following parameter values:

```
[NOTIFY]
sms = TRUE
```

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

In place of the `<hostname>` and `<portnumber>` variables in the `smsnotifyprogramparam` parameter example given above, supply the host name and port number of an instance of Oracle9iAS Wireless PIM Notification Dispatcher.

If you do not know the host name and port number of an instance of Oracle9iAS Wireless PIM Notification Dispatcher, you can find out using the Wireless system management area of Oracle Enterprise Manager. For more details, consult the Oracle9iAS Wireless documentation.

4. Restart all calendar servers stopped in step 1.

Resources, Event Calendars and Holidays

This chapter describes the various tasks involved in creating and managing organizational resources, event calendars and holidays. 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, although with some important differences in an LDAP context.

The following topics are dealt with in this chapter:

- [Managing holidays](#)
- [Creating event calendars](#)
- [Managing and deleting event calendars](#)
- [Creating resources](#)
 - Internal directory context
 - LDAP Directory context
- [Managing resources](#)
- [Deleting resources](#)
- [Managing resource defaults](#)
- [Assigning designates](#)

Managing holidays

The first step in holiday management is to assign the appropriate administration rights to a user. The SYSOP, by default, does not have holiday management rights.

Web GUI

You can add holidays to a node using the Calendar Administrator as long as any user on the selected node has the right to administer holidays. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

Use the `uniadmrights` utility to assign holiday administration rights. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

For example:

```
% uniadmrights -e "S=Sitchin/G=Zechariah" -add -hday -n 165
```

Once you have assigned holiday administration rights to a user, use a desktop client to sign in as that user, and select [Directory | Manage Holidays...](#)

Creating event calendars

An event calendar is an administrative user account which exists for the purpose of informing your user base about upcoming happenings that may be of interest to them. Essentially, an event calendar is simply a user account that has the `PUBLISHEDTYPE` key set to `EVENTCALENDAR`. Other users have read-only access to the events you create in your event calendars. Web calendar users, however, may copy events and appointments from the event calendar into their own calendars. Only administrators may create event calendars, and only administrators may modify events in an event calendar.

Web GUI

Use the Calendar Administrator to create event calendars easily. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

Use the `uniuser` utility introduced in [Chapter 7, "Users and Groups"](#), to create event calendars. Remember to set the value of the `PUBLISHEDTYPE` key to `EVENTCALENDAR` to make the user account an event calendar.

For example:

```
% uniuser -add "S=Season/G=Hockey/PUBLISHEDTYPE=EVENTCALENDAR" -n 156
```

Set the `GLOBALREADACCESS` key to `TRUE` if you want any Internet user to have the ability to view this event calendar, just as you would with a user account. For full information on the use and syntax of `uniuser`, see the calendar server Reference Manual, Appendix C, "Utilities."

Managing and deleting event calendars

Web GUI

Use the Calendar Administrator to manage, populate and delete event calendars easily. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

Use the `uniuser` utility introduced in [Chapter 7, "Users and Groups"](#), to modify and delete event calendars. For full information on the use and syntax of `uniuser`, see the calendar server Reference Manual, Appendix C, "Utilities."

To add new events to an event calendar, you may also sign in as that event calendar using your calendar client of choice and add the desired events or appointments. 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.

Creating 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 invite resources in the same way that they invite other users. Resources are managed by local users who act as designates. Resources can either be set up to permit more than one reservation at a time, or on a first come first served basis to prevent double-bookings.

Resources 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.

Adding new resources to the calendar server's internal directory

Web GUI

You can add resources to a node using the Calendar Administrator. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

To add a resource: Use `unires -add` to add a single resource. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

```
% unires -add "R=conference room/N=4" -n 786 -p <SYSOP password>
unires: added "conference room"
```

To add several resources:

1. Create a file of the resources that you wish to add to a node. The information for each resource must be entered following the format and syntax documented in the `unires` documentation in the calendar server Reference Manual, Appendix C, "Utilities."

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

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

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

Adding new resources in an LDAP directory context

You may specify a Resource Relative DN for your installation. 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.

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. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

To add a resource: Use `unires -add` to add a single resource. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities." This example specifies a full DN for the new resource.

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

```
% unires -add "R=Room614" -n 134
Enter SysOp password:
unires: 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 `unires` documentation in the calendar server Reference Manual, Appendix C, "Utilities." 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.

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

```
unires: added "cn=projector3, o=Acme, c=US"  
unires: added "cn=lab5, o=Acme, c=US"
```

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

Managing resources

Web GUI

You can manage resources easily using the Calendar Administrator. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

View and modify resource attributes using the `unires` utility. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Admin GUI

To edit resource information:

1. Select **Resource | Properties** to open the Search dialogue box.
2. Click **Search** without entering any information for a complete list of resources on the node network. To restrict your search, enter a valid Resource Number or the first letter(s) of a resource name and click **Search**.
3. Select a resource name from the list box and click **Properties**, or double-click a resource. Although the list box may contain resources from connected nodes, you can view and modify properties only for resources located on the node that you are signed into.
4. Click **OK** when you have finished editing the record.

Deleting resources

When a resource is deleted from a node, the resource's records and directory entry are removed from the local node. This means that the resource will no longer appear as invited in user's agendas, nor will it appear in any directory listings. If required, the resource's agenda can be copied out using the `unicpoutr` utility, and subsequently copied back in to any calendar server node using the `unicpinr`

utility. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Web GUI

You can delete resources easily using the Calendar Administrator. For full information on the Calendar Administrator, see [Appendix F, "The Calendar Administrator"](#).

Cmd line

Remove the resource(s) from the calendar server node using the `unires -del` (single deletion) or `unires -ex` (multiple deletions) commands. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Admin GUI

To delete a calendar resource:

1. Select a resource, following the procedure outlined under "[To edit resource information:](#)" above.
2. Select the desired resource name and click **Delete**.

Managing resource defaults

You may wish to 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 `/users/unison/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 the calendar server Reference Manual, Appendix A, "User and Resource Parameters."
- 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 `/users/unison/misc/resource.ini` file.

Applying a default resource profile:

- The resource profile is applied during resource creation (using either the `unires` utility or the Admin GUI).
- The default resource profile is outlined under the section heading `[GEN]` in the `/users/unison/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 `unires` command. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."
- The default profile of one or more resources can be modified using the `unires -ex` command. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."

Assigning designates

Designate and other access rights are set according to the default resource profile at the time the resource is added to the node. They can also be set later by using a calendar client to log in as the resource and modify its access rights.

Node Maintenance

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

This chapter outlines the following tasks:

- [Maintenance procedures](#)
- [Back up and restore](#)

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.
The `unistatus` utility displays the current status of the calendar server. For full information on use and syntax, see the calendar server Reference Manual, Appendix C, "Utilities."
- Check that ample space is left in the `/users/unison` 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 `/users/unison/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 `/users/unison/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 (contact your support provider for the most current version), 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.

Daily maintenance procedures

A nightly backup of the calendar database (`/users/unison/db`) and configuration files (`/users/unison/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 "[Back up and restore](#)" on page 10-3.

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, see the calendar server Reference Manual, Appendix C, "Utilities."
- 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, see the calendar server Reference Manual, Appendix C, "Utilities."

Other maintenance procedures

- Verify the consistency of the server database(s) using the `unidbfix` utility. Contact your support provider for the most current version of this comprehensive database maintenance and repair 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.
- 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, see the calendar server Reference Manual, Appendix C, "Utilities." You should perform this synchronization procedure every 2-4 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 Admin GUI.
- 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 `/users/unison/log/stats.log` file. After analysis, set `[ENG] stats = FALSE` to disable logging, as the file grows quickly.

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, important in the context of installations requiring 24-hour access to their calendar data. An on-line backup can **not** 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:

- clients connected when the backup is initiated are permitted to log off
- clients not connected when the backup is initiated are not permitted to log on
- logged-on users 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

`unidbrestore` is the complementary utility used for database restoration. For full information on the use and syntax of the `unidbbackup` and `unidbrestore` utilities, see the calendar server Reference Manual, Appendix C, "Utilities."

To back up a calendar host:

- Execute the `unidbbackup` utility through the command line or using the Database Backup dialog box of the Admin GUI. 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 in turn.

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 `/users/unison` 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 the calendar server Reference Manual, Appendix C, "Utilities."

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.

Monitoring Procedures

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

- [Viewing log files](#)
- [Interpreting log files](#)

Viewing log files

To view a log file, go to the `/users/unison/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 11-1 *Calendar server log files*

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

Table 11–1 Calendar server log files

Filename	Description
<code>dbi.log</code>	For node (database) initialization.
<code>dbv.log</code>	Database operation file. Created only if there is a problem.
<code>dsstats.log</code>	For directory server (LDAP) calls.
<code>eng.log</code>	For the Engine.
<code>lck.log</code>	For the Lock Manager.
<code>script.log</code>	For all UNIX utilities.
<code>snc.log</code>	For the Synchronous Network Connections.
<code>stats.log</code>	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 <code>[ENG] stats</code> parameter in <code>unison.ini</code> to <code>TRUE</code> . The size of the <code>stats.log</code> file should be closely monitored since it can increase quickly.
<code><utility>.log</code>	For various utilities that create and update self-named log files when they are run.

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, contact your technical support representative for further assistance.

Disk Space and Memory

This appendix describes the memory requirements of the calendar server. These requirements can be broken down into two categories: database disk space requirements and run-time requirements involving virtual memory and swap space.

- [Database disk space requirements](#)
- [Remote NFS storage](#)
- [Run-time requirements](#)

Database disk space requirements

Server performance is influenced by disk configuration. To enhance the performance of the server, it is recommended that the application be allocated its own disk (`/users/unison`). Further enhancements can be obtained by devoting one or more disks for permanent user data (`/users/unison/db/nodes`) and temporary user data (`/users/unison/db/tmp`).

Each directory entry, whether for a local or remote item, requires 2 Kb of disk space. Local items require additional disk space for calendar data. The disk space requirements of a remote item are therefore considerably less than those of a local item, although the exact figure will depend on the usage of Corporate-Wide Services. Additional yearly disk space requirements of a local item can be estimated by determining the following values:

Table A-1 Variable definitions

Variable	Definition
DAYS_YEAR	number of work days in a year

Table A-1 Variable definitions

Variable	Definition
ITEMS_EVENT	average number of items invited to an event
EVENTS_ITEM_DAY	average number of events attended by an item daily
GROUPS_ITEM	average number of groups created by an item
ITEMS_GROUP	average number of items in a group
SECURITIES_ITEM	average number of security privileges granted by an item
ATTACHED_STREAMS	average size of the comments and objects attached to an event
TASK_STREAMS	average size of the comments attached to a task
TASKS_ITEM_DAY	average number of tasks created daily by an item

Using the values that you determine for these variables, the following formula determines disk space, in bytes, attributed to a local item on a yearly basis:

$$\begin{aligned}
 \text{disk space} = & 1631 + \\
 & (\text{SECURITIES_ITEMS} * 123) + \\
 & (\text{GROUPS_ITEM} * (335 + (\text{ITEMS_GROUP} * 63))) + \\
 & (((\text{ITEMS_EVENT} * 128) + 768 + \text{ATTACHED_STREAMS}) / \text{ITEMS_EVENT}) * \\
 & \text{DAYS_YEAR} * \text{EVENTS_ITEM_DAY}) + \\
 & (\text{DAYS_YEAR} * \text{TASKS_ITEM_DAY} * (1155 + \text{TASK_STREAMS}))
 \end{aligned}$$

Consider an example where the following values are set:

Variable	Definition
DAYS_YEAR	260
ITEMS_EVENT	7
EVENTS_ITEM_DAY	8
GROUPS_ITEM	50
ITEMS_GROUP	15
SECURITIES_ITEM	100

ATTACHED_STREAMS	512
TASKS_ITEM_DAY	3
TASK_STREAMS	256

$\text{disk-space} = 1631 +$
 $(100 * 123) +$
 $(50 * (335 + (15 * 63))) +$
 $(260 * 8 * ((768 + 512 + (7 * 128)) / 7) +$
 $(260 * 3 * (1155 + 256)))$

2,080,000 or 2 megabytes of disk space per user per year

One final note concerning disk-space: each active calendar user has temporary data files, located in `/users/unison/db/tmp`, whose combined size should not exceed 450 Kb.

Remote NFS storage

UNIX installations of Oracle Calendar server support linking the calendar database on a remote NFS file system. If you choose to do so, only the `/users/unison/db` directory may be stored remotely. All other directories must remain local.

Run-time requirements

The server run-time environment consists of four UNIX daemons/multi-threaded Windows NT services: the Lock Manager (`unilckd`), Engine (`uniengd`), Synchronous Network Connections (`unisncd`), Corporate-Wide Services (`unicwsd`), and a `uniengd` server for each active user. An additional UNIX daemon/multi-threaded Windows NT service, the Directory Access Server (`unidasd`), is included in installations using an external directory. At startup, with no connections, a total of 5 Mb RAM is used by the daemons/services. Each connection requires between 500 and 900 Kb RAM, dependent on hardware and configuration, for an Engine server/thread to service that client's requests. Use 750 Kb as a general benchmark. Contact Oracle support for more information on your chosen platform. The maximum number of concurrent users is set for an installation by the `lck_users` parameter, and the number of persistent connections to the directory server is set by the `numconnect` parameter in the

`/users/unison/misc/unison.ini` file. Thus, the formula to calculate RAM requirements would be:

internal directory

$5 \text{ Mb} + (750 \text{ Kb} * \text{lck_users})$

with directory server

$5 \text{ Mb} + (750 \text{ Kb} * \text{lck_users}) + (300 \text{ Kb} * (\text{numconnect} - 1))$

Sizing Guidelines

After determining the number of nodes and servers required, use the information in this appendix to arrive at estimates of your memory and disk requirements.

As a starting point, it is important to note that:

- Each user and resource, whether for a local user or a remote user, takes 2 Kb of disk space.
- An average user needs approximately 2 Mb of disk space per year.
- An active connection requires an average of 750 Kb of virtual memory. Thus, 750 Kb of RAM per connection eliminates any swapping considerations, and no more than 750 Kb of swap space per user is required.

Note: This figure provided for virtual memory requirements varies according to the operating system.

Memory and disk requirements

To determine memory and disk requirements, the following guidelines should be followed:

Determine the amount of memory required per server. This is calculated by multiplying 750 Kb by the number of connections for that server. If the server will support more than one node, multiply 750 Kb by the number of users for all nodes. Add an additional 64 Mb for system use.

Determine the amount of permanent storage required per server. Each configured user requires approximately 2 Mb of permanent storage. If the server will support more than one node, multiply 2 Mb by the total number of users for all nodes. Add an additional 50 Mb for product files and logs, and also add an appropriate amount

for other storage not directly related to the calendar server (i.e. operating system, system swap area, etc.).

Determine the amount of temporary storage required per server. This is calculated by multiplying 750 Kb by the number of logged-on users for the server. If the server will support more than one node, multiply 750 Kb by the number of users for all nodes.

The server has additional system-level storage and memory requirements which are not affected by the number of users. For planning purposes, each server will require 64 Mb of extra memory and 50 Mb of disk space. The disk space requirement is for calendar server software, including logs and executables. These numbers do not include operating system requirements which may vary by vendor.

Determine the number of spindles required per server. One spindle per 100 active users is suggested for both permanent and temporary data. In other words, to support 100 users, one spindle for permanent data and one spindle for temporary data is recommended. This is a performance-related requirement to ensure that disk I/O is distributed across multiple drives. For optimal performance, three additional drives should be provided: one for the operating system, one for the operating system swap space, and one for the server executables and logs.

The number of spindles for permanent and temporary storage has been established based on the recommendation that one spindle be used for each 100 active users for permanent data, and one spindle for each 100 active users for temporary data. A number higher than 100 may be used, with some degradation in performance; the higher the number of active users per spindle, the greater the performance degradation.

Finally, the size of the disks for permanent and temporary storage may be identified. For each server (or for each node if separate drives will be used for each node), divide the total disk requirements by the number of spindles.

System Configuration

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 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](#)
- [Adjusting the HP-UX kernel parameters](#)
- [Adjusting the Solaris kernel parameters](#)
- [Adjusting the AIX kernel parameters](#)
- [Adjusting the Tru64 kernel parameters](#)
- [Adjusting the Windows kernel](#)
- [Using operating system clusters](#)
- [Additional reading](#)

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.

Assume that Corporate-Wide Services is disabled. Observe that in extreme situations (i.e. power failure), all server sessions will be terminated in rapid

succession. Thus, in the worst case, the maximum number of simultaneous log-ons and log-offs can be as large as `[LCK] lck_users`. The server therefore requires at least the following resources:

Table C-1 Calculating new kernel parameter values

parameter description	new value
max. number of concurrent processes per user-id:	$[2(\text{lck_users}) + 10]$ processes per user *
soft file limit per process:	$[\text{db_files} + 10]$ files per-process**
hard file limit per process:	$[\text{db_files} + 10]$ files per-process, or $[\text{number of nodes} + 10]$ files per process, whichever is larger
max. number of concurrent processes:	$[2(\text{lck_users}) + 10]$ processes (system-wide)
max. number of open files:	$[(\text{lck_users} + 1) * (\text{db_files} + 5) + 20]$ files (system-wide)

* limitations for this parameter only apply if the calendar server runs as the user "unison"

** If you have not specified a value for the parameter `[DB] db_files` in `/users/unison/misc/unison.ini`, use the default value of 30 for all calculations

If Corporate-Wide Services is enabled, further resources are required:

- add at least 10 to the max. number of concurrent processes, and to max. number of concurrent processes per user-ID
- add at least 20 to max. number of open files

The total resource requirements when Corporate-Wide Services are enabled:

Table C-2 Calculating new kernel parameter values with CWS enabled

parameter description	new value
soft file limit per process:	$[\text{db_files} + 10]$ files per-process
hard file limit per process:	$[\text{db_files} + 10]$ files per-process, or $[\text{number of nodes} + 10]$ files per process, whichever is larger
max. number of concurrent processes per user-id:	$[2(\text{lck_users}) + 20]$ processes per-user*

Table C–2 Calculating new kernel parameter values with CWS enabled

max. number of concurrent processes:	$[2(\text{lck_users}) + 20]$ processes (system-wide)
max. number of open files:	$[(\text{lck_users} + 1) * (\text{db_files} + 5) + 40]$ files (system-wide)
* limitations for this parameter only apply if the calendar server runs as the user “unison”	

Finally, consider the message and semaphore resource requirements of the server, which are independent of Corporate-Wide Services:

Table C–3 Calendar server message and semaphore requirements

parameter description	requirement
max. number of message queue identifiers:	$[\text{lck_users} + 1]$ message queue identifiers (system-wide)
max. number of message headers:	$[\text{lck_users} + 1]$ message headers (system-wide)*
max. number of semaphore identifiers:	$[1]$ semaphore identifier (system-wide)
max. number of semaphores:	$[1]$ semaphore (system-wide)
message segment size	64 bytes
* This formula presumes that each calendar server message fits into a single message segment. A message can vary in length between 56 and 114 bytes.	

The memory allocation in bytes as related to the Lock Manager daemon (unilckd):

$400 \text{ Kb} * \text{lck_users}$

A final comment concerning the uniengd daemon/service: this executable has been compiled (cc) in share mode (-n) insuring that all concurrently executing copies of uniengd share a single instruction section (sometimes called "Program text segment") to save memory.

Note: The system clock must be running and properly set.

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 assumption is that Corporate-Wide Services is enabled, and that less than 30 nodes are present on the host. The assumptions concerning the server `unison.ini` file parameters are:

- [DB] `db_files` = 30
- [SYS] `sys_owner` = unison
- [LCK] `lck_users` = N

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

Table C-4 HP-UX kernel parameters

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
<code>maxfiles</code>	soft file limit per process	X	40	$\max(40, X)$
<code>maxfiles_lim</code>	hard file limit per process	X	40	$\max(40, X)$
<code>maxuprc</code>	max. number of concurrent processes per user-ID	X	$2*N+20$	$\max((2*N+20), X)$
<code>nproc</code>	max. number of concurrent processes	X	$2*N+20$	$(2*N+20)+X$
<code>nfile</code>	max. number of open files	X	$35*N+75$	$(35*N+75)+X$
<code>msgmni</code>	max. number of message queue identifiers	X	$N + 1$	$N + 1 + X$
<code>msgssz</code>	size of message segments	X	144	$\max(144, X)$
<code>msgmnb</code>	max total size of all messages in a queue simultaneously	X	$\min(65535, (\text{msgssz} * \text{lck_users}))$	$\min(65535, (\text{msgssz} * \text{lck_users}) + X)$

Table C-4 HP-UX kernel parameters

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
msgtql	max. number of message headers	X	N + 1	N + 1 + X
semgni	max. number of semaphore identifiers	X	1	1 + X
semnns	max. number of semaphores	X	1	1 + X

The following is an actual example involving 500 users:

- [DB] db_files = 30
- [SYS] sys_owner = unison
- [LCK] lck_users = 500

Table C-5 HP-UX kernel parameters (example)

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
maxfiles	soft file limit per process	X	40	max(40,X)
maxfiles_lim	hard file limit per process	X	40	max(40,X)
maxuprc	max. number of concurrent processes per user-ID	X	1020	max((1020),X)
nproc	max. number of concurrent processes	X	1020	1020+X
nfile	max. number of open files	X	17575	17575+X
msgmni	max. number of message queue identifiers	X	N	N + X
msgssz	size of message segments	X	144	max(144,X)
msgmnb	max total size of all messages in a queue simultaneously	X	min(65535, (msgssz * lck_users))	min(65535, (msgssz * 500) + X)
msgtql	max. number of message headers	X	N	N + X
semgni	max. number of semaphore identifiers	X	1	1 + X
semnns	max. number of semaphores	X	1	1 + X

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 maxuprc=2641
set max_nprocs=2636
```

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 above changes to take effect, reboot the system.

To see a complete list of the tunable kernel parameters, run the `nm` command on the appropriate module. For example:

```
% /usr/ccs/bin/nm/kernel/genunix
```

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

```
% sysdef -i
```

The assumption is that Corporate-Wide Services is enabled, and that less than 30 nodes are present on the host. The assumptions concerning the `unison.ini` parameters are:

- [DB] db_files = 30
- [SYS] sys_owner = unison
- [LCK] lck_users = N

In the columns below, the “max” function returns the larger of the two arguments.

Table C-6 Solaris Kernel Parameters

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
<code>rlim_fd_cur</code>	soft file limit per process	X	40	$\max(40, X)$
<code>rlim_fd_max</code>	hard file limit per process	X	$\max(\text{db_files} + 10, \text{number of nodes} + 11)$	$\max(X, \text{db_files} + 10, \text{number of nodes} + 11)$
<code>maxuprc</code>	max. number of concurrent processes per user-ID	X	$2*N+20$	$\max((2*N+20), X)$
<code>max_nproc</code>	max. number of concurrent processes	X	$2*N+20$	$(2*N+20)+X$
<code>msgsys:msginfo_msgmni</code>	max. number of message queue identifiers	X	$2*N$	$(2*N) + X$
<code>msgsys:msginfo_msgtql</code>	max. number of message headers	X	N	$N + X$
<code>msgsys:msginfo_msgssz</code> †	size of message segments	X	144	$\max(144, X)$
<code>msgsys:msginfo_msgmnb</code>	number of bytes in any message queue	X	$\text{msgssz} * N$	$(\text{msgssz} * N) + X$
<code>msgsys:msginfo_msgseg</code> †	maximum message segments in all queues	X	N	$N + X$
<code>msgsys:msginfo_msgmap</code> †	Number of free space resource map entries	X	N	$N + X$
<code>semsys:seminfo_semmni</code>	max. number of semaphore identifiers	X	1	$1 + X$
<code>semsys:seminfo_semmns</code>	max. number of semaphores	X	1	$1 + X$

† Available on Solaris 2.6 and 7 only.

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

The following is an example with 500 users:

- `Number of nodes = 3`
- `[DB] db_files = 30`

- [SYS] sys_owner = unison
- [LCK] lck_users = 500

Table C-7 Solaris kernel parameters (example)

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
rlim_fd_cur	soft file limit per process	X	40	max(40,X)
rlim_fd_max	hard file limit per process	X	max(40, 14)	max(X, 40, 14)
maxuproc	max. number of concurrent processes per user-ID	X	1020	max(1020,X)
max_nproc	max. number of concurrent processes	X	1020	1020+X
msginfo_msgmni	max. number of message queue identifiers	X	2*N	(2*N) + X
msginfo_msgtql	max. number of message headers	X	N	N + X
msgsys:msginfo_msgssz †	size of message segments	X	144	max(144, X)
msgsys:msginfo_msgmnb	number of bytes in any message queue	X	msgssz * N	(msgssz * N) + X
msgsys:msginfo_msgseg †	maximum message segments in all queues	X	N	N + X
msgsys:msginfo_msgmap †	Number of free space resource map entries	X	N	N + X
seminfo_semmni	max. number of semaphore identifiers	X	1	1 + X
seminfo_semmns	max. number of semaphores	X	1	1 + X

† Available on Solaris 2.6 and 7 only.

Adjusting the AIX kernel parameters

Unlike many other UNIX operating systems, AIX does not provide the ability to directly configure most kernel parameters. Instead, the AIX kernel dynamically allocates and reallocates resources as they are needed (up to a predefined limit).

For calendar server installations, the only kernel parameter that should be altered is `maxuproc` (maximum number of processes per user-ID), which can be modified *via* SMIT (AIX's menu-based system administration utility) or the command line utility `chdev`.

To see a complete list of the parameters and their current values, run the `lsattr` command:

```
% /etc/lsattr -E -l sys0
```

To see the current values assigned to a particular kernel parameter, use the `lsattr` command with the parameter name. For example:

```
% /etc/lsattr -E -l sys0 -a maxuproc
```

To change the value of the `maxuproc` parameter, use the `chdev` command. For example:

```
% /etc/chdev -l sys0 -a maxuproc = 200
```

As stated above, the AIX kernel dynamically allocates and reallocates resources as they are needed up to a predefined limit. The known limits of the parameters are as follows:

Table C-8 AIX kernel parameters

Kernel Parameter Description	Upper Limit
soft file limit per process	2000
hard file limit per process	2000
max. number of concurrent processes per user-id:	configurable, cannot exceed 131,072
max. number of concurrent processes on the system	131,072
max. number of open files	200000
max. number of message queue identifiers	4096
max. number of message headers	unlimited; 8192 per message queue.
max. number of semaphore identifiers	4096
max. number of semaphores	unlimited; 65535 per semaphore id.

The practical limits of the server, defined as the maximum number of concurrent users on a system, are effectively determined by the upper limits of the maximum number of open files. Assuming there are no other activity or resource requirements on the system, the maximum number of concurrent users (`lck_users`) under AIX is approximately $10000 * 20 + 60 = 200060$.

Adjusting the Tru64 kernel parameters

Modify your Tru64 kernel parameters using the `/usr/bin/X11/dxkerneltuner` utility. Editing the “Current Value” field will apply changes only until the system is rebooted; editing the “Boot Time Value” field will apply changes permanently at the next boot. Altering both values will apply the changes both immediately and permanently.

For information on command-line utilities for kernel adjustment, consult the relevant documentation on the `/sbin/sysconfig` and `/sbin/sysconfigdb` commands.

The assumption is that Corporate-Wide Services is enabled, and that less than 30 nodes are present on the host. The assumptions concerning the `unison.ini` parameters are:

- `[DB] db_files = 30`
- `[SYS] sys_owner = unison`
- `[LCK] lck_users = N`

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

Table C-9 *Tru64 kernel parameters*

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
PROC/open-max-soft	soft file limit per process	X	40	$\max(40, X)$
PROC/open-max-hard	hard file limit per process	X	40	$\max(40, X)$
PROC/max_proc_per_user	max. number of concurrent processes per user-ID	X	$2 * N + 20$	$\max((2 * N + 20), X)$
PROC/maxusers	max. number of concurrent processes	X	$2 * N + 20$	$(2 * N + 20) + X$
VFS/max_vnodes	max. number of open files	X	$35 * N + 75$	$(35 * N + 75) + X$

Table C-9 Tru64 kernel parameters

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
IPC/msg_mni	max. number of message queue identifiers	X	N + 1	N + 1 + X
IPC/msg_tql	max. number of message headers	X	N + 1	N + 1 + X
IPC/sem_mni	max. number of semaphore identifiers	X	1	1 + X

The following is an actual example involving 500 users:

- [DB] db_files = 30
- [SYS] sys_owner = unison
- [LCK] lck_users = 500

Table C-10 Tru64 kernel parameters (example)

Kernel Parameter	Parameter Description	OLD setting	Calendar server requirement	NEW setting
PROC/open-max-soft	soft file limit per process	X	40	max(40,X)
PROC/open-max-hard	hard file limit per process	X	40	max(40,X)
PROC/max_proc_per_user	max. number of concurrent processes per user-ID	X	1020	max((1020),X)
PROC/maxusers	max. number of concurrent processes	X	1020	1020+X
VFS/max_vnodes	max. number of open files	X	17575	17575+X
IPC/msg_mni	max. number of message queue identifiers	X	N	N + X
IPC/msg_tql	max. number of message headers	X	N	N + X
IPC/sem_mni	max. number of semaphore identifiers	X	1	1 + X

Adjusting the Windows kernel

Oracle Calendar server is subject to the default Windows limit on memory size per process (2 Gb). You can improve performance for installations with large numbers of connections (1000 or more) by configuring your Windows kernel to allow processes up to 3 Gb of memory by following the instructions given on Microsoft's Web site at:

<http://support.microsoft.com/support/kb/articles/Q171/7/93.ASP>

This feature is available for Windows NT Server, Enterprise Edition version 4.0 and greater.

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 `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 `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.

Limitations

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

Additional reading

For more information on modifying the kernel parameters under Solaris, refer to *Administering Security, Performance, and Accounting* by Sun Microsystems, Inc.

HP-UX system parameters are described in Appendix A, System Parameters, of the HP-UX manual entitled *System Administration Tasks*.

For more information on AIX performance tuning and administration, please consult *AIX Performance Tuning* by IBM, Prentice Hall, 1996.

Information on Tru64 kernel parameters is available in the *Compaq Tru64® Unix™ System Configuration and Tuning* manual, by Compaq Computer Corporation, 1999.

An excellent source of general information on this topic is *System Performance Tuning* by Mike Loukides, O'Reilly & Associates, Inc., 1991.

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](#)
- [Directory server security](#)
- [Other security considerations](#)

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](#)" on page D-11 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](#)" on page D-4 for the relevant configuration parameters, and "[Extending the ACE framework](#)" on page D-5 for details on extending the available set of methods.

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 `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](#)" on page D-5 below for information on extending the sets of supported methods. Consult the appropriate page in the calendar server Reference Manual, Appendix B, "Server Parameters," for details on these parameters.

Table D-1 ACE configuration parameters

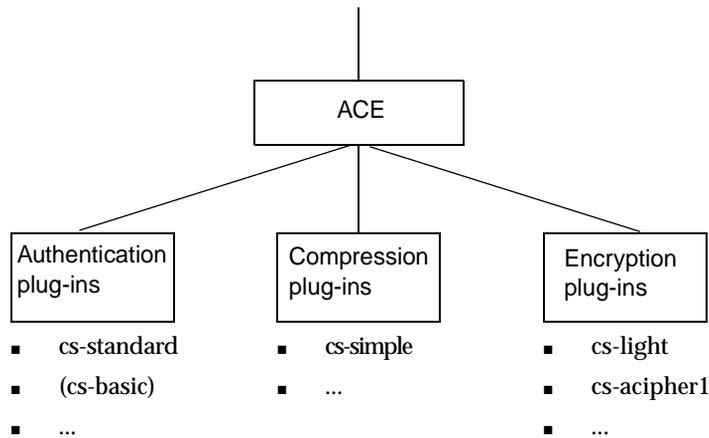
Section	Parameter	Description
[ACE]	frameworkenable	Enable the ACE framework
[AUTHENTICATION]	supported	Supported authentication methods for clients
	default	Default authentication method for clients
[COMPRESSION]	supported	Supported compression methods
	default	Default compression method for clients
	servicedefault	Default compression method for other servers
[ENCRYPTION]	supported	Supported encryption methods
	needsauthenticate	Encryption methods requiring prior authentication
	default	Default encryption method for clients

Table D-1 ACE configuration parameters

Section	Parameter	Description
	servicedefault	Default encryption method for other servers

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 D-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) libaut_cs-standard.sl (HP-UX) libaut_cs-standard.so (SunOS)
cmp_	Compression	cmp_cs-simple.dll (NT) libcmp_cs-simple.sl (HP-UX) libcmp_cs-simple.so (SunOS)
enc_	Encryption	enc_cs-acipher1.dll (NT) libenc_cs-acipher1.sl (HP-UX) 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

Plug-in Name	Keyword
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 above. Consult the plug-in documentation to determine `<sub-mechanism_name>`. For example:

Plug-in Name	Keyword
libaut_sasl.so	sasl:GSSAPI

Note that SASL and GSSAPI are 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, please consult the calendar server Reference Manual, Appendix B, "Server Parameters."

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 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. However, the installation utility generates shared keys (see below for an explanation) automatically in the server and client configuration files. It is the administrator's responsibility to ensure that these shared keys match across all calendar Web clients and server components.

How does the Web authentication plug-in work?

1. The Web authentication plug-in identifies itself to the calendar server when the Web client FastCGI is started. This is done through comparing the values of two parameters, one stored in the calendar Web client configuration file (`webcal.ini`), and one stored in the calendar server configuration file (`unison.ini`). If the values of these shared keys are the same, the calendar server will trust the Web authentication plug-in as a proxy authenticator.
2. When a user is authenticated with the Web server and attempts to access a FastCGI application, the server stores some of that user's attributes in environment variables.
3. When that user attempts to view his or her calendar, the Web calendar client application calls the Web authentication plug-in.
4. The Web authentication plug-in then retrieves the user information stored in the Web server environment variables, and uses that information to identify the current user to the calendar server. It passes the value of a specified environment variable to the calendar server, and indicates what user attribute in the calendar database that environment variable represents. Currently, the only attribute supported is the user's UID.
5. The calendar server looks for a user whose UID matches the value submitted. If it finds a match, the user is signed directly in to their Web calendar.

Configuring your calendar server

1. Edit the `users/unison/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:CAL`, to the list of supported authentication methods given in the `[AUTHENTICATION] supported` parameter. For example:

```
[AUTHENTICATION]
supported = {cs-standard, web:CAL}
```

4. Create a new section called `[ACE_PLUGINS_SERVER]` and a new parameter called `web_CAL_sharedkey` to specify the shared key to be used by the Web

authentication plug-in. The value of this parameter should be a simple plain-text password. For example:

```
[ACE_PLUGINS_SERVER]
web_CAL_sharedkey = mypassword
```

Configuring your Web calendar client

1. Edit the `webcal.ini` file.
2. Enable the Web authentication plug-in by changing the value of the `[ACE]` authentication parameter to `web:CAL`. Set this parameter as follows:

```
[ACE]
authentication = web:CAL
```

3. Create a new section called `[ACE_PLUGINS_CLIENT]` and a new parameter called `web_CAL_sharedkey` to specify the shared key to be used by the Web authentication plug-in. The value of this parameter must match the value you set for the calendar server `unison.ini` `[ACE_PLUGINS_SERVER]` `web_CAL_sharedkey` parameter. For example:

```
[ACE_PLUGINS_CLIENT]
web_CAL_sharedkey = mypassword
```

4. Add a new 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, 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.

5. Add a new 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 = SSL_CLIENT_S_DN_UID
```

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.

6. Specify the default page to load if the calendar server is unable to match the authenticated user. Set the `[href] login_fail_url` parameter to the desired path and file name, enclosed in quotes. For example:

```
[href]
login_fail_url = "<filename>"
```

Note: By default, errors and warnings will be logged to a file in the same directory as the Web calendar client application. To specify another directory for the log file, use the [ACE_PLUGINS_CLIENT] `web_logfile` parameter.

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.). There should be only two accounts configured on the system: unison and root (UNIX) or administrator (NT).

Password Management

Note that user passwords may not be set using calendar administration tools or calendar clients. 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).

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 unison 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 unison (server management) accounts. This would allow root/administrator to detect potential abuses by the unison 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 `/users/unison/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 10, "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.

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.

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 `/users/unison/misc/unison.ini` file controls the conversion and storage of calendar data in the UTF-8 format. By default, this feature is enabled. See the calendar server Reference Manual, Appendix B, "Server Parameters," 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 `/users/unison/misc/unison.ini` file (see the calendar server Reference Manual, Appendix B, "Server Parameters," for information on this parameter). Client character sets are either identified explicitly to the server by the application itself, or determined based on the operating system of the client. If necessary, it is possible to manually set the definition of the client character set by inserting the appropriate parameter in the `[LOCALE]` section of the `unison.ini` file.

Parameter	Value
<code>charsetwindows</code>	= <defined character set>
<code>charsetwindows32</code>	= <some character set>
<code>charsetwindows16</code>	= <some character set>
<code>charsetmac</code>	= <some character set>
<code>charsetmac68k</code>	= <some character set>
<code>charsetmacppc</code>	= <some character set>
<code>charsetCGI</code>	= <some character set>
<code>charsetmotif</code>	= <some character set>
<code>charsetmotifaix</code>	= <some character set>
<code>charsetmotifhpux</code>	= <some character set>
<code>charsetmotifirix</code>	= <some character set>
<code>charsetmotiflinux</code>	= <some character set>
<code>charsetmotifopenserver</code>	= <some character set>
<code>charsetmotifosf</code>	= <some character set>
<code>charsetmotifsolaris</code>	= <some character set>
<code>charsetmotifsunos</code>	= <some character set>
<code>charsetmotifunixware</code>	= <some character set>

Japanese server configuration

If installing the server on a Japanese operating system, the following parameters in the `/users/unison/misc/unison.ini` file should be set to the indicated

values. See the calendar server Reference Manual, Appendix B, "Server Parameters," for more detailed explanations of the function of each parameter.

```
[ENG]
localcharset= MSCP932
[CWS]
mimecontentcharset=Shift_JIS
mailhdrtoname=FALSE
[LOCALE]
charsetwindows=MSCP932
charsetCGI=Shift_JIS
```

See also the documentation concerning the `[CWS] emailcharsetmap` parameter.

Setting user language preferences

The calendar server offers users e-mail and SMS reminders in the language of their choice.

Cmd line

Set a user's language through the `uniuser` utility with the "LANG" key. Consult Reference Appendix C, "Utilities," for full information on use and syntax. The following example sets Herman Hesse's language to German:

```
uniuser -mod "S=Hesse/G=Herman" -m "LANG=de-DE" -n 14
```

Alternately, you may use the `user.ini [GEN] Language` parameter to specify a default language to apply to all new users. Consult Reference Appendix A, "User and Resource Parameters," for details on available languages and parameter values.

The Calendar Administrator

The Calendar Administrator is a new, fully customizable Web-based administration tool for calendar server management. The CGI or fastCGI application runs on a Web server, displaying HTML template files that it updates dynamically with information from the calendar server.

Through the Calendar Administrator, you can manage users, resources, holidays and event calendars. You may also customize the look and feel of the application to match your organization's identity.

While the GUI itself is designed to be intuitive and easy-to-use, there are some other issues to consider.

- [Web server configuration](#)
- [Calendar Administrator customization](#)
- [Calendar Administrator configuration](#)

Web server configuration

Ensure that the Calendar Administrator is set up on your Web server with the following configuration:

- `uniwebadm.cgi` and `uniwebadm.ini` are in the `/caladmin` directory within your Web server's `cgi-bin` directory or `fcgi-bin` directory (e.g. `/usr/local/Apache/cgi-bin/caladmin`)
- HTML template files and stylesheet files (`uniwebadm.css`) are stored in a subdirectory within the directory containing `uniwebadm.cgi` (e.g. `/usr/local/Apache/cgi-bin/caladmin/templates_domain1`)
- `uniwebadm.js` is in the `/caladmin` directory within your Web server's DocumentRoot (e.g. `/usr/local/Apache/htdocs/caladmin`)

Calendar Administrator customization

Three levels of customization exist for the Calendar Administrator. On the first level, parameters in the `uniwebadm.ini` file can be configured to control the level of administration possible through the interface. Secondly, most aspects of the look and feel of the application can be easily controlled through a cascading stylesheet (`uniwebadm.css`) that is applied to all HTML template files. Thirdly, the template files themselves can be changed to make more radical alterations to the Calendar Administrator look and feel.

By parameters

Customize the administrative operations available through the Calendar Administrator interface using the `[PROVISIONINGSETTINGS]` `manageusers`, `manageresources`, `manageholidays`, `manageeventcalendars`, and `modifyadminpassword` parameters in `uniwebadm.ini`. See "[Calendar Administrator configuration](#)" below for more details on these parameters.

By stylesheet

All HTML template files used by the Calendar Administrator link to a cascading stylesheet file, `uniwebadm.css`. For more information on cascading stylesheet use and syntax, consult the `.css` specifications available from the World Wide Web Consortium at <http://www.w3c.org>. Be advised that different browsers implement cascading stylesheets in very different ways.

By templates

You may choose to alter the HTML template files used by the Calendar Administrator. If you do so, ensure that you do not delete any tags bearing the "CADM" reference. These tags are used by `uniwebadm.cgi` to update the pages dynamically; removing them will cause errors. Use the `/users/unison/misc/domain/domain.ini [YOURDOMAINID]` `adm.templatesdir` parameter to configure different template directories for each of your domains.

Calendar Administrator configuration

The Calendar Administrator configuration file (`uniwebadm.ini`) contains a number of configurable parameters that control the behaviour of the application.

Keep this file in the same directory as `uniwebadm.cgi`. Most of these parameters are set during installation.

This table lists all `uniwebadm.ini` parameters in alphabetical order by section.

Table F-1 `uniwebadm.ini` configuration parameters

Section	Parameter	Description
[ADMINSETTINGS]	<code>adminoldpassword</code>	Old SYSOP password — disregarded
	<code>imagesdir</code>	Directory for images
	<code>caldomainserverlist</code>	List of Calendar Domain Service hosts — disregarded
	<code>cgimode</code>	CGI or FCGI operation
	<code>serverpassword</code>	SYSOP password — disregarded
	<code>sessionsdir</code>	Directory for temporary sessions files
[PROVISIONINGSETTINGS]	<code>domainlogo</code>	Domain logo
	<code>manageeventcalendars</code>	Event calendar management
	<code>manageholidays</code>	Holiday management
	<code>manageresources</code>	Resource management
	<code>manageusers</code>	User management
	<code>sitelogo</code>	Site logo
	<code>sitename</code>	Site name
	<code>supportemail</code>	Support e-mail
	<code>templatesdir</code>	Directory for HTML template files

Directory for HTML template files

Section

[PROVISIONINGSETTINGS]

Parameter

`templatesdir`

Description

Specifies a directory for HTML template files.

Accepted values

A path relative to the location of the Calendar Administrator CGI or fastCGI

An absolute path

Default value

templates

Directory for temporary sessions files**Section**

[ADMINSETTINGS]

Parameter

sessionsdir

Description

Specifies a directory for temporary files identifying sessions of the Calendar Administrator.

Accepted values

A path relative to the location of the Calendar Administrator CGI or fastCGI

An absolute path

Default value

sessions

Directory for images**Section**

[ADMINSETTINGS]

Parameter`imagesdir`**Description**

Specifies a directory for any images to be displayed in the Calendar Administrator interface.

Accepted values

A path relative to the location of your Web server's DocumentRoot

An absolute path

Default value`images`**List of Calendar Domain Service hosts — disregarded****Section**`[ADMINSETTINGS]`**Parameter**`caldomainserverlist`**Description**

Not for use in Oracle Calendar server installations. Please do not set a value for this parameter without explicit instructions from technical support personnel.

CGI or FCGI operation**Section**`[ADMINSETTINGS]`**Parameter**`cgimode`

Description

Specifies whether the Calendar Administrator should run as a CGI or FCGI application. This parameter should only be used if it becomes necessary to force a particular behaviour. By default, the Calendar Administrator will run as a CGI or FCGI application depending on the choice made at installation time and the file extension of the application.

Accepted values

cgi

fcgi

auto

Default value

auto

SYSOP password — disregarded

Section

[ADMINSETTINGS]

Parameter

serverpassword

Description

Not for use in Oracle Calendar server installations. Please do not set a value for this parameter without explicit instructions from technical support personnel.

Old SYSOP password — disregarded

Section

[ADMINSETTINGS]

Parameter

adminoldpassword

Description

Not for use in Oracle Calendar server installations. Please do not set a value for this parameter without explicit instructions from technical support personnel.

Site name**Section**

[PROVISIONINGSETTINGS]

Parameter

sitename

Description

Specifies the display name of the Calendar Administrator provider within the Web interface.

Accepted values

Any string

Default value

None

Site logo**Section**

[PROVISIONINGSETTINGS]

Parameter

sitelogo

Description

Specifies the logo of the Calendar Administrator provider within the Web interface.

Accepted values

A filename in the directory specified by the [ADMINSETTINGS] `imagesdir` parameter

A URL relative to the Web server base URL

A fully-qualified URL

Default value

None

Domain logo

Section

[PROVISIONINGSETTINGS]

Parameter

domainlogo

Description

Specifies a second logo, if desired.

Accepted values

A filename in the directory specified by the [ADMINSETTINGS] `imagesdir` parameter

A URL relative to the Web server base URL

A fully-qualified URL

Default value

None

Support e-mail

Section

[PROVISIONINGSETTINGS]

Parameter

supportemail

Description

Specifies an e-mail address for Calendar Administrator users to contact in case of errors or difficulties.

Accepted values

A valid e-mail address

Default value

None

User management**Section**

[PROVISIONINGSETTINGS]

Parameter

manageusers

Description

Determines whether user management options will be available through the Calendar Administrator interface.

Accepted values

TRUE (User management is available)

FALSE (User management is disabled)

Default value

TRUE

Resource management**Section**

[PROVISIONINGSETTINGS]

Parameter

manageresources

Description

Determines whether resource management options will be available through the Calendar Administrator interface.

Accepted values

TRUE (Resource management is available)

FALSE (Resource management is disabled)

Default value

TRUE

Holiday management

Section

[PROVISIONINGSETTINGS]

Parameter

manageholidays

Description

Determines whether holiday management options will be available through the Calendar Administrator interface.

Accepted values

TRUE (Holiday management is available)

FALSE (Holiday management is disabled)

Default value

TRUE

Event calendar management

Section

[PROVISIONINGSETTINGS]

Parameter`manageeventcalendars`**Description**

Provides a default value for the event calendar management option in the Calendar Administrator's Domain Creation page.

Determines whether event calendar management options will be available through the Calendar Administrator interface.

Accepted values

TRUE (Event calendar management is available)

FALSE (Event calendar management is disabled)

Default value

TRUE

Admin password management**Section**

[PROVISIONINGSETTINGS]

Parameter`modifyadminpassword`**Description**

Provides a default value for the Domain Administrator password management option in the Calendar Administrator's Domain Creation page..

Accepted values

TRUE (Password modification is available)

FALSE (Password modification is disabled)

Default value

TRUE

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 new cgi or fcgi-based tool for on-line calendar administration.

Calendar daemons/ services

Five 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 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.

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

access control limitations for directory servers, 2
access rights, 8
ACE, 7
ACE framework, 5
 architecture, 5
 configuration parameters, 4
 enabling, 4
 extending the set of plug-ins, 6
 keyword for plug-in, 6
 parameters, 4
active users, 2
act.log, 1
adding
 nodes to network, 7
 resources, 3 to ??
 users, 2
 users to a directory server, 3, 4
administration rights, managing, 8
administrative groups, 8
administrator, major tasks of, 6
AIX kernel parameters, 8
alerts, 1 to 4
alias, node, 8, 9, 1, 2
anonymous binding, 4
architecture, 1
 ACE framework, 5
 calendar server, 4
attachments, enabling, 7
auditing, 12
authentication, ?? to 7
 cs-basic, 6

B

backups, 3, 4
 restoring contents of, 3, 4
bandwidth, 4
base DN, 10
bind DN, 4

C

Calendar Administrator, 1 to 11
 configuring, 2
 customizing, 2
character set
 client, 2
 for database, 1
 international, 1
 Japanese, 2
 UTF-8, 1
clients, xvii
 character set, 2
 connections, 5, 6
 modifying access rights through, 8
 secure connections to, 2 to 7
clusters, 2, 9
coexistence, LDAP and non-LDAP nodes, 13
compression, 2 to 7
configured users, 2
configuring the Calendar Administrator, 2
connections
 See also secure connections
 between nodes, 2, 6
 decreasing, between nodes, 7
 increasing, between nodes, 7

- tuning, 7
- Corporate-Wide Services, 3
 - configuration, 4
 - processing, 4
- creating
 - event calendars, 2
 - holidays, 1 to 2
 - resources, 3 to ??
 - users, 2
- customizing the Calendar Administrator, 2
- cws.log, 1

D

- daemons, 2
 - unicwsd, 3
 - unidasd, 3, 4
 - uniengd, 2
 - unilckd, 2
 - unisncd, 3
- das.log, 1
- dasstats.log, 1
- database
 - disk space requirements, 1
 - remote storage, 3
- dbi.log, 2
- dbv.log, 2
- defaults
 - user, 6
- deleting
 - event calendars, 3
 - nodes, 3
 - resources, 6
 - users, 5
- deployment, 1
- designates
 - assigning, 8
- Directory Access Server, 4
- directory manager password, 2
- directory server, 1 to 5
 - access control, 2
 - adding users from, 2
 - base URL, 10
 - binding, 4
 - groups and group filters, 3

- node network and, 4
- secure connections to a, 11 to ??
- disk space, 1
 - database requirements, 1
 - directory entry requirements, 1
 - estimating yearly requirements, 1
 - remote storage, 3
 - requirements, 1 to 2
- disks, number per server, 2
- distinguished name
 - Calendar Administrators' Group, 10
 - Calendar Administrators' Parent, 10
 - Resource Relative DN, 7
- dsstats.log, 2

E

- e-mail
 - filtering, 2
- encryption, 2 to 7
 - SSL, 11
- Engine, 2
- eng.log, 2
- event calendars, 1
 - creating, 2
 - deleting, 3
 - managing, 3

F

- filtering e-mail, 2

G

- Global Directory Server, 4
- global read access, 9
- glossary, ?? to Glossary-4
- groups, 7
 - administration rights, 8
 - administrative, 8
 - directory server, 3
 - granting permission to create, 8
 - managing, 7
 - members-only, 7
 - private, 7

- public, 7
- guidelines
 - connections between nodes, 2

H

- holidays, 1 to 2
 - administration rights, 8
 - granting administration rights, 8
- host, calendar server, 1
- host, mail, 8, 10
- HP-UX kernel parameters, 4

I

- implementation, 1
- installation, 1
 - information checklist, 8, 9
- internal directory
 - adding users, 2
 - creating resources, 4

J

- Japanese
 - character set, 2
 - client character set, 2

K

- kernel parameters, 6, 1 to 13
 - AIX, 8
 - HP-UX, 4
 - Solaris, 6
 - Tru-64, 10
- keywords
 - for ACE framework plug-ins, 6

L

- lck.log, 2
- LDAP
 - coexistence of LDAP and non-LDAP nodes, 13
 - Data Interchange Format, Glossary-2
 - resource management, 1
 - support in Oracle Calendar server, xvii

- LDIF
 - for exporting user and resource data, 4
 - UNIX/NT differences, 14
- Lock Manager, 2
- log files, 1 to 2
 - interpreting, 2
 - viewing, 1 to 2
- logged-on users, 2

M

- mail
 - filtering, 2
 - host, 8, 10
 - notification, 7, 8, 9, 8
- maintenance, 1 to 5
 - daily procedures, 2
 - monthly procedures, 2
 - other procedures, 3
- managing
 - event calendars, 3
 - groups, 7
 - holidays, 1 to 2
 - node network, 1 to 15
 - resource defaults, 7
 - resources, 1 to 8
 - user defaults, 6
 - users, 1 to 10
- master node, 9
- master nodes, 2
- memory requirements, 3, 1 to 2
- migrating users. See moving users
- MIME headers, 2
- monitoring, 1
 - daily procedures, 1
- moving
 - nodes, 9 to 13
 - users, 6

N

- network. See node network
- NFS storage, 3
- node network
 - adding nodes, 7

- configuring, 2
- definition, 1
- directory server, 4
- guidelines, 2
- managing, 1 to 15
- master node, 9
- removing nodes, 7
- node-ID, 8, 9, 1
- nodes, 2
 - adding to network, 7
 - alias, 8, 9, 1, 2
 - changing SYSOP password, 3
 - clusters, 2
 - coexistence of LDAP and non-LDAP, 13
 - connecting, 2, 3, 6
 - creating, 1
 - deleting, 3
 - grouping users, 3
 - maintaining, 1 to 5
 - master nodes, 2
 - moving, 9 to 13
 - removing from network, 7
 - setting up, 1
 - size, 3
 - time zone, 1, 2
- nodes.ini, 2 to 7
 - rules, 6
 - syntax, 5
- notification, 1 to 4
 - mail, 7, 8, 9, 8

O

Oracle9iAS Wireless, 1

P

- paragraph tags
 - PT PrefaceTitle, xvii
- parameters
 - See also kernel parameters
 - activity [ENG], 1
 - allowattachments [LIMITS], 7, 13
 - caldomainserverlist [ADMINSETTINGS], 5
 - charset [LDAP], 2

- charsetCGI [LOCALE], 2
- charsetmac [LOCALE], 2
- charsetmac68k [LOCALE], 2
- charsetmacppc [LOCALE], 2
- charsetmotif [LOCALE], 2
- charsetmotifaix [LOCALE], 2
- charsetmotifhpux [LOCALE], 2
- charsetmotifirix [LOCALE], 2
- charsetmotiflinux [LOCALE], 2
- charsetmotifopensever [LOCALE], 2
- charsetmotifosf [LOCALE], 2
- charsetmotifsolaris [LOCALE], 2
- charsetmotifsunos [LOCALE], 2
- charsetmotifunixware [LOCALE], 2
- charsetwindows [LOCALE], 2
- charsetwindows16 [LOCALE], 2
- charsetwindows32 [LOCALE], 2
- dac_itemget [ENG], 2
- db_files [DB], 2, 3, 4, 6, 10
- default [AUTHENTICATION], 4
- default [COMPRESSION], 4
- default [ENCRYPTION], 4
- dir_internal_nodes [ENG], 14
- domainlogo [PROVISIONINGSETTINGS], 8
- enable [DAS], 14
- frameworkenable [ACE], 7, 4
- imagesdir [ADMINSETTINGS], 4
- lck_users [LCK], 3, 2, 3, 4, 6, 10
- localcharset [ENG], 3
- mail [LIMITS], 7
- mailhdrtoname [CWS], 3
- manageeventcalendars
 - [PROVISIONINGSETTINGS], 10
- manageholidays
 - [PROVISIONINGSETTINGS], 10
- manageresources
 - [PROVISIONINGSETTINGS], 9
- manageusers [PROVISIONINGSETTINGS], 9
- maxattachmentsize [LIMITS], 7, 13
- mimecontentcharset [CWS], 3
- modifyadminpassword
 - [PROVISIONINGSETTINGS], 11
- needsauthenticate [ENCRYPTION], 4
- numconnect [YOURHOSTNAME, unidas], 3
- resourceconflicts [LIMITS], 8

- resourcerelatedn [LDAP], 7
- serveroldpassword [ADMINSETTINGS], 6
- serverpassword [ADMINSETTINGS], 6
- servicedefault [COMPRESSION], 4
- servicedefault [ENCRYPTION], 5
- sessionsdir [ADMINSETTINGS], 4
- sitelogo [PROVISIONINGSETTINGS], 7
- sitename [PROVISIONINGSETTINGS], 7
- snc_tr_block [SNC], 8
- stats [ENG], 3, 2
- supported [AUTHENTICATION], 4, 7
- supported [COMPRESSION], 4, 7
- supported [ENCRYPTION], 4, 7
- supportemail [PROVISIONINGSETTINGS], 8
- sys_owner [SYS], 4, 6, 10
- templatesdir [PROVISIONINGSETTINGS], 3
- trace [CWS], 1
- usermailmap [ENG], 8
- utf8_autoconvert [ENG], 1

passwords

- changing SYSOP, 3
- directory manager, 2
- management, 11
- node, 9
- SYSOP, 7, 2

planning, 1

plug-ins, 5

- extending the available set of, 6

preface

- conventions table sample, xxi

private groups, 7

PT PrefaceTitle, xvii

public groups, 7

published types, 9

R

reminders, 1 to 4

remote storage, 3

removing

- nodes from network, 7

requirements

- memory, 3, 1 to 2

- virtual memory, 1

Resource Relative DN, 7

resources

- booking, 7

- creating, 3 to ??

- default profile for, 7

- definition, 3

- deleting, 6

- managing, 1 to 8

- restoring contents of backup, 3, 4

S

script.log, 2

secure connections, 1 to ??

- clients supporting, 2

- configuration parameters, 4

- enabling, to calendar clients and other calendar servers, 4

- negotiating with a calendar client, 2

- negotiating with another calendar server, 3

- to a directory server, 11 to ??

- to calendar clients and other calendar servers, 2 to 7

security, 7, 1 to 13

- application, 13

- auditing, 12

- backup and recovery, 13

- dedicated server, 11

- networking, 12

- password management, 11

- secure connections, 1 to ??

- trust management, 12

server

- architecture, 1

- status, 2

services. See daemons

snc.log, 2

Solaris kernel parameters, 6

SSL encryption, 11

starting the calendar server, 1

stats.log, 2

status

- server, 2

stopping the calendar server, 1

SuperUser DN, 10

swap space

- recommended amount of, 2
- requirements, 1
- Synchronous Network Connection, 3
- SYSOP, 1
 - password, 7, 2

T

- time zones, 8, 9
 - node, 1, 2
- Tru-64 kernel parameters, 10
- tuning connections, 7

U

- unicwsd, 3
- unidasd, 3, 4
- uniengd, 2
- unilckd, 2
- unisncd, 3
- uniwebadm.ini, 2

See also parameters

- URL, base for directory server, 10
- user.ini, 6, 7
- users
 - active, 2
 - adding, 2
 - adding to a directory server, 3, 4
 - concurrent, number of, 8, 9
 - configured, 2
 - creating, 2
 - default profile for, 6
 - deleting, 5
 - grouping, 3
 - logged-on, 2
 - logical groupings, 3
 - managing, 1 to 10
 - modifying, 4
 - moving, 6
 - viewing, 4
 - viewing logged-on, 2
- UTF-8, 1
- utilities
 - uniaddnode, 2

- uniadmrights, 8
- unib2lendian, 10
- unicheck, 3
- unicksum, 12
- unclean, 4
- uniclr_ipc, 4
- unicpinr, 6
- unicpinu, 5
- unicpoutr, 6
- unicpoutu, 5
- unidbbackup, 3
- unidbconv, 10
- unidbfix, 10, 3
- unidbrestore, 4
- unidsdiff, 5, 3, 5
- unidssearch, 3
- unidssync, 2, 5
- unil2bendian, 10
- unimvuser, 4, 6
- uninode, 2, 3, 10
- unipasswd, 3
- unireqdump, 3
- unires, 4, 5, 6, 7, 8
- unirmold, 2
- unistart, 1
- unistatus, 2, 1
- unistop, 2, 1
- uniuser, 2, 5, 7
- uniwho, 2

V

- virtual memory requirements, 1

W

- web server, 1
- WebGUI. See Calendar Administrator
- Wireless, 1