

Administration Guide

iPlanet Calendar Server

Version 2.0

Part Number: 806-3476-10

November, 1999

Copyright © 1999 Sun Microsystems, Inc. Some preexisting portions Copyright © 1999 Netscape Communications Corp. All rights reserved.

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Netscape and the Netscape N logo are registered trademarks of Netscape Communications Corporation in the U.S. and other countries. Other Netscape logos, product names, and service names are also trademarks of Netscape Communications Corporation, which may be registered in other countries.

Federal Acquisitions: Commercial Software -- Government

Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun-Netscape Alliance and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Version 2.0

Part Number: 806-3476-10 Rev A

Copyright © 1999 Sun Microsystems, Inc. Some preexisting portions Copyright © 1999 Netscape Communications Corp. All rights reserved.

Contents

Preface	7
Who Should Read This Book	7
What You Need to Know	7
What's in This Guide	8
Document Conventions	8
What is iPlanet Calendar Server?	9
Calendar Server Features	10
Architectural Overview	10
HTTP Protocol	11
Web Calendar Access Protocol (WCAP)	12
Core Calendar Services	12
Calendar Server Application Program Interface (CSAPI)	12
Supported Format Encoding	13
Database Technology	14
User Authentication and Preferences	15
LDAP (Netscape DS 4.1)	15
External Directories	15
Administration	15
Chapter 1 Installation and Deployment	17
Installation Process Overview	17
Before You Begin the Installation	18
System Requirements	18
Hardware Requirements	18
Software Requirements	18
Installation Privileges	19
Installation Packages	19
Types of Installation	20
Accounts, Groups, and Administrators	20

Directory Server Accounts	21
Windows NT System Account	21
Unix System Accounts	21
Installation Guidelines	22
Transferring the Calendar Server Installation Files	22
Running the Installation Program	23
Deployment Considerations	23
Chapter 2 Configuring Calendar Server	25
Using the Server.conf file	25
Database Administration	34
Calendar Server API	35
Notification Messages	37
Special Character Sequences for Events	38
Date Sub-Formatting	40
Conditional Printing	40
Special Character Sequences for Task Notification	41
Special Character Sequences for Dates	43
Simple Event Reminder Example	44
Complex Event Reminder Example	45
Counters	46
Chapter 3 Administering Calendar Server	49
Administration Tools	49
Starting and Stopping the Server	50
Using start-cal and stop-cal	50
Using iplncal.sh (Unix only)	50
Using the Services Panel (Windows NT only)	51
Using the Command Line Utilities	51
Managing Calendar Users	52
Creating a Calendar User	53
Deleting a Calendar User	53
Enabling a Calendar User	54
Disabling a Calendar User	54
Checking if a User is Enabled	55

Displaying User Information	55
Resetting User Calendar Attributes	55
Managing Calendars	55
Creating a Calendar	56
Deleting Calendars	57
Enabling a Calendar	57
Disabling a Calendar User	57
Modifying Calendar Properties	58
Displaying Calendars	58
Resetting Calendar Properties	59
Managing Calendar Events and Tasks	59
Managing Calendar Databases	60
Importing and Exporting Calendar Data	61
Exporting Calendar Data	61
Importing Calendar Data	61
Backup and Restore Procedures	62
Backing-Up a Specified Calendar	62
Restoring a Specified Calendar	63
Restoring a Calendar from a File	63
Restoring a Calendar from a Database	63
Backing-Up a User's Default Calendar	63
Restoring a User's Default Calendar	64
Backing-Up the Calendar Database	64
Restoring the Calendar Database	64
Using Solstice or Legato Software to Backup Calendar Server	65
Backing Up Calendar Server Data	66
Restoring Calendar Server Data	67
Chapter 4 Monitoring the Calendar Server	69
Managing User Sessions	69
Logging Out a User	70
Listing Counter Statistics	70
Pinging the Calendar Server	71
Refreshing the Calendar Server Configuration	71

Appendix A Command Line Utilities	73
Command Line Utility Syntax	73
Usage Rules	74
Command Line Utility Summary	74
csbackup	75
cscal	78
cscomponents	81
csdb	83
csexport	85
csimport	87
csrestore	88
cssession	91
csstart	93
csstats	95
csstop	97
cstool	99
csuser	101
Appendix B Monitoring Tools	105
Appendix C Time Zones	109
Time Zone Administration	112
Adding a New Time zone	114
Modifying an Existing Time Zone	114
Customizing Time Zones in the Calendar Server User Interface	115
Calendar Server Time Zones Table	115
Appendix D Calendar Server LDAP Schema	117
Index	119

This manual explains how to install and administer iPlanet Calendar Server 2.0 and its accompanying software components. Calendar Server is a powerful and flexible cross-platform solution using open Internet standards that lets service providers of all sizes host personal scheduling calendars for their customers.

Who Should Read This Book

You should read this book if you are responsible for installing and deploying Calendar Server at your site.

What You Need to Know

Before you install and administer Calendar Server, you must be familiar with the following concepts:

- basic administrative procedures of your operating system (UNIX or Windows NT)
- using a web browser to access the internet
- Lightweight Directory Access Protocol (LDAP) if you plan to use an LDAP directory to store user information

What's in This Guide

If you want to...	See...
Learn more about Calendar Server and how to get started.	“What is iPlanet Calendar Server?” in this preface
Install and deploy Calendar Server.	Chapter 1, “Installation and Deployment”
Configure your Calendar Server.	Chapter 2, “Configuring Calendar Server”
Administer your Calendar Server using the command line utilities.	Chapter 3, “Administering Calendar Server”
Monitor and control your Calendar Server.	Chapter 4, “Monitoring the Calendar Server”
Learn about the complete syntax and usage of the command line utilities.	Appendix A, “Command Line Utilities”
Learn about the server monitor tools.	Appendix B, “Monitoring Tools”
Add or modify Calendar Server time zones.	Appendix C, “Time Zones”
See the LDAP schema extensions used by Calendar Server.	Appendix D, “Calendar Server LDAP Schema”
Find information in the guide.	Index

Document Conventions

Sidebar text Sidebar text marks important information. Make sure you read the information before continuing with a task.

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 filenames, distinguished names, functions, and examples.

Italicized Monospaced font—This typeface is used to represent text that you enter using information that is unique to your installation. It is used for server paths and names and account IDs.

For example, throughout this document you will see path references of the form:

server-root/cal

In these situations, *server-root* represents the directory path in which you installed Calendar Server. For example, if you install your server in the default directory location suggested by the installation program the *server-root* is *opt/SUNWicsrv* where:

- on Unix the actual path is: */opt/SUNWicsrv/cal*
- on Windows NT the actual path is: *c:\iplanet*

Also, all paths specified in this manual are in Unix format. If you are using a Windows NT-based Calendar Server, you should assume the Windows NT equivalent file paths whenever Unix file paths are shown in this book.

What is iPlanet Calendar Server?

iPlanet Calendar Server is the first readily deployable, Internet web-based solution that lets the ISP/Enterprise and Telecommunication service provider host personal event calendars to their base of subscribed customers. Using iPlanet Calendar Express, a web browser-based client application, customers can access data hosted on the Calendar Server to:

- maintain personal schedules with their own appointments
- publish calendars for viewing by other people
- general reminders
- browse other people's calendars

Built from the ground up to provide native support for the emerging set of internet calendar standards, iPlanet Calendar Server provides the following benefits:

- **Massively Scalable.** Scaling to the requirements of the largest service providers, it supports a hosting environment of up to several million personal event calendars.

- **Low Cost of Ownership.** Native support of LDAP lets a service provider centrally manage its entire customer base in a single user directory and minimizes the costs of administering the server while also providing a platform for extending the enhanced services a provider can offer its customers.
- **Internet Calendaring and Scheduling.** Native support for iCalendar calendaring standards lets users schedule personal events in a format that is easily shared across the internet.

Calendar Server Features

iPlanet Calendar Server offers the service provider the following administration capabilities:

- Works with any LDAP server.
- Enables customization of the Calendar Express client user interface (e.g., login screen, colors, and templates).
- Delivers calendar data in multiple formats (e.g., XML, iCAL, and Javascript) wrapped within HTML.
- Provides command line utilities to manage calendar data and calendar users, and to monitor and control the Calendar Server.

Architectural Overview

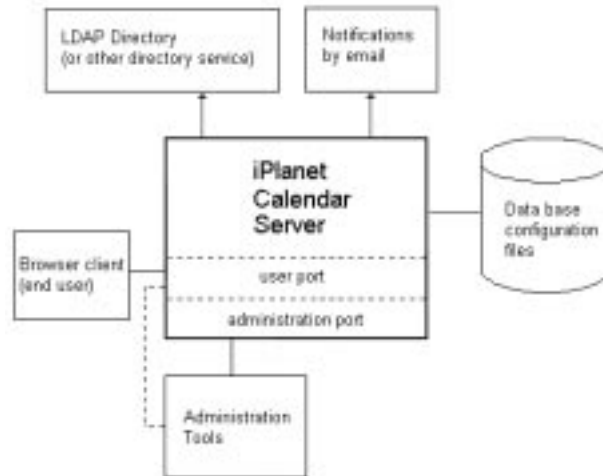
The following sections provide a general overview of the iPlanet Calendar Server product architecture which consists of the following subsystem components:

- HTTP Protocol
- Web Calendar Access Protocol (WCAP)
- Core Calendar Server services
- Calendar Server Application Program Interface (CSAPI)

- Calendar Database

Figure 1 illustrates iPlanet Calendar Server general architecture:

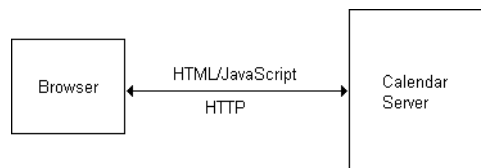
Figure 1 iPlanet Calendar Server Architectural Overview



HTTP Protocol

iPlanet Calendar Server provides a default web calendar client (Calendar Express) and a general purpose HTTP server to support HTTP clients that request calendar data. Figure 2 illustrates, at the highest level, how Calendar Server operates.

Figure 2 HTTP as Primary Transport



Calendar Server uses HTTP as its primary client. It receives commands encoded in HTML and sends replies as HTML and JavaScript. The data returned from requests is not formatted for display. The embedded JavaScript generates HTML

which it sends to a visible frame accessed by a web browser. The user experience is the same as if the HTML was loaded from a server. The advantage is that for many operations, JavaScript can handle everything locally and does not need to contact the server. This provides a faster screen update for the user while lightening the load on the Calendar Server.

Web Calendar Access Protocol (WCAP)

The Web Calendar Access Protocol (WCAP) is a simple calendar protocol for storing, retrieving, and administering calendars. WCAP interprets HTTP requests and sends them to the core calendar services for execution. WCAP then receives commands from the core services, then encodes them into HTML format and returns them to the browser client.

Core Calendar Services

Commands enter through the HTTP protocol subsystem as WCAP. The core services make data requests to the calendar database where information is stored or retrieved and then returned in HTML format. The core system then translates:

- low-level format information into the format requested by WCAP commands which is then encoded into HTML format and returned to the browser client.
- stored data into the binary format used by the calendar database.

Calendar Server Application Program Interface (CSAPI)

The Calendar Server Application Program Interface (CSAPI) enables third party application developers to add functionality to Calendar Server or change how it performs certain operations. CSAPI provides supports in four major functional areas:

- authentication

- user attributes and preferences
- data translators
- Calendar Application Program Interface (CAPI)

For example, the default mechanisms used by Calendar Server for user authentication and storing user preferences relies on LDAP, but an ISP may have an existing infrastructure for authenticating users and saving user information that is not based on LDAP. An ISP can use CSAPI to override these default mechanisms and utilize their existing authentication and user directory services.

Supported Format Encoding

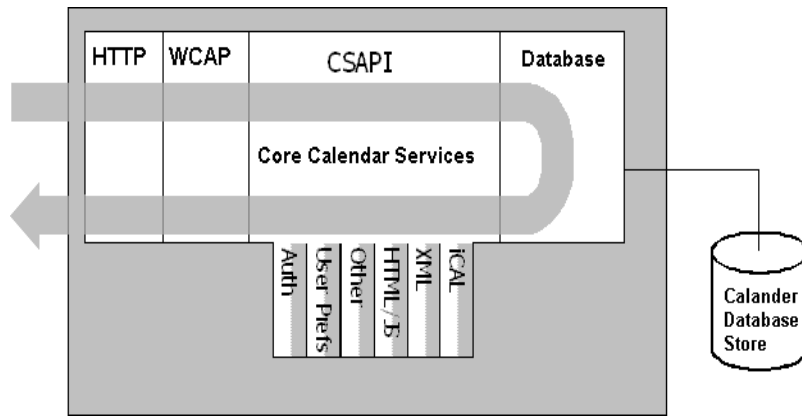
Calendar Server supports the following format encoding (all embedded within HTML):

- HTML/JavaScript (default)
- HTML/XML
- HTML/iCalendar

Other formats can be added by third parties by developing a translator DLL or shared library using the Calendar Server Application Program Interface (CSAPI), an API for extending the Calendar Server.

Figure 3 is a detailed diagram of the iPlanet Calendar Server components and data flow design.

Figure 3 Subsystems and Data Flow



Database Technology

The local database used by Calendar Server is the Berkeley DB which is included as part of the installation package and supports transactions and live backups.

Calendar Server uses three physical databases:

- A calendar database that stores information such as the properties, events and tasks, and alarm notifications belonging to each calendar. The calendar database files are located in the *server-root/cal/bin/lock* directory.
- A sessions database that stores information about user sessions such as the login state time outs of calendar users. The sessions database files are located in the *server-root/cal/bin/sessions* directory.
- A counter database that stores statistics about the Calendar Server such as the total number of successful and failed server logins. The counter database file (which is actually a memory mapped file rather than a Berkeley database) is located in the *server-root/cal/bin/* directory.

User Authentication and Preferences

By default, Calendar Server supports users that are defined and maintained in an LDAP directory, such as Netscape Directory Server. Calendar Server also supports the use of CSAPI plug-ins that you create and enable access to users defined in non-LDAP directories such as those stored in a standard Unix Authentication format or in a Windows NT User Manager database.

LDAP (Netscape DS 4.1)

If your users are already stored in an LDAP directory, the simplest solution to deploy Calendar Server is to upgrade your directory server to Netscape Directory Server 4.1 which includes the schema extension that enables users to access Calendar Server data. Otherwise, you can modify your directory schema manually to allow your users to access to Calendar Server data. For more information on how to modify a directory schema for Calendar Server, see Appendix D, “Calendar Server LDAP Schema.”

External Directories

If your users are stored in a manner other than LDAP directory format, such as an OS specific or a proprietary authentication program, you can create a plug-in that uses CSAPI to authenticate users and give them access to data hosted on the Calendar Server.

Administration

Calendar Server provides several command line utilities that let you perform administrative tasks (such as enabling and disabling access to Calendar data) and changing Calendar properties (such as owners). For more information, see Chapter 3, “Administering Calendar Server,” and Appendix A, “Command Line Utilities.”

Installation and Deployment

This chapter contains the following sections to help you prepare for installation:

- Installation Process Overview
- Before You Begin the Installation
- System Requirements
- Installation Privileges
- Installation Packages
- Installation Guidelines

Installation Process Overview

iPlanet Calendar Server 2.0 uses an installation program that provides a consistent installation interface and procedure across all supported platforms. In addition, Calendar Server provides tools for uninstalling the server, stopping and starting various server processes, and enabling users to access data hosted on the server. Instructions for using these tools are provided in the relevant sections of this document.

Note For the most up to date installation instructions, refer to the online version listed at:

<http://developer.netscape.com/docs/manuals/calendar.html>

Installing Calendar Server involves the following steps:

1. Transferring the Calendar Server Installation Files
2. Running the Installation Program

Before You Begin the Installation

Before you begin installing iPlanet Calendar Server 2.0, you should verify that the systems on which you plan to install the software meet the minimum product requirements. In addition, you should understand what the various Calendar Server components are. It is also a good idea to plan how you want to configure the software components before you begin the installation process.

System Requirements

Before you install the Calendar Server, you must make sure you have met the minimum hardware and operating system requirements.

Hardware Requirements

The basic minimum hardware requirements for a Calendar Server installation are as follows:

- Roughly 500 MB of disk space for a standard installation. For production systems, you should plan at least 1 GB to support the product binaries, database, and log files; 2 GB and or greater for very large calendar stores.
- 32 MB of RAM. For large production systems, however, plan from 256 MB to 1 GB of RAM for best performance.
- Adequate space for your calendar user data.
- RAID storage for fast access (optional).

Software Requirements

Calendar Server 2.0 supports the following operating systems at the appropriate version and patch levels:

- Solaris 2.6 with recommended patches. See:
<http://access1.sun.com/patch.recommended/rec.html>
- Microsoft Windows NT 4.0 with Service Pack 3 or later.
- HP-UX 11.00 with the following required operating system patches:
PHNE_13208
PHNE_13405
PHNE_14279

Note For the most recent software requirements and supported operating systems, see the Calendar Server 2.0 Release Notes listed online at:

<http://developer.netscape.com/docs/manuals/calendar.html>

Installation Privileges

It is recommended that you install Calendar Server logged in as root (Unix) or administrator (Windows NT). Superuser privileges (root or administrator) are required for Calendar Server 2.0 installations if you plan to use the default port numbers (which are less than 1024).

Installation Packages

The product contains the following separate software components:

iPlanet Calendar Server 2.0. Provides a powerful and flexible cross-platform solution to the Calendar Server hosting needs of service providers of all sizes. Using an open, Internet-standard approach to calendar hosting, it offers rapid processing of calendar scheduling data scalable to many thousands of simultaneous users.

Calendar Server API (CSAPI). A high performance interface to enhancing and modifying the feature-set of the Calendar Server. This API lets you create very fast runtime shared objects which outperform both system executable programs and scripts in any language in terms of speed, memory footprint, and load time. CSAPI is a shared-object runtime interface to Calendar Server functions that supports the use of plugin CSAPI modules to manipulate server data when the server receives requests and sends responses. This

package also includes Web Calendar Access Protocol (WCAP), a high level command-based protocol that clients use to communicate with iPlanet Calendar Server through CSAPI.

Types of Installation

The Calendar Server 2.0 installation program lets you install the product in the following ways:

- **Typical Installation.** This type of installation asks the questions required to install the server. Some options are automatically configured with default values, but some require that you choose or enter the value you wish to use. Typical Installation is recommended for beginning or intermediate users or for any installation that does not require custom configuration.
- **Custom Installation.** This is the most complex type of installation. You must choose and enter all configuration options. Custom Installation is recommended for expert users only. Use this type of installation if you want to perform advance activities during installation such as specifying network ports other than the default values.
- **Silent Installation.** Use this type of installation if you want to automate your installation process when installing more than one instance of Calendar Server. This feature is especially useful for installing several servers at your site.

Accounts, Groups, and Administrators

The accounts, groups, and administrators referred to in this guide are grouped into the following categories:

- **Directory server accounts.** Unless noted otherwise, these accounts were created if you installed Directory Server.
- **Windows NT System Account.** You can create this account manually or let the installation program create it for you.
- **Unix system Accounts.** If they do not exist, you will need to create these accounts on your Unix system before you install Calendar Server 2.0.

Directory Server Accounts

The accounts described in this section are those used with Netscape Directory Server. For more information, refer to the Netscape Directory Server documentation.

Calendar Server Administrator. This is the username and associated password that can make configuration changes to the Calendar Server. This user has administration privileges over the Calendar Server, but not over the Directory Server itself. The default is `calmaster`.

Directory Administrator. Also known as "Unrestricted User". If you use LDAP (the default mechanism) for user authentication, this is the username and associated password that can make changes to the Directory Server schema. This user has overall administrator privileges on the Directory Server and all servers that make use of the Directory Server, such as the Administration Server and Calendar Server, and has full administration access to all entries in the Directory Server. You enter the Directory Administrator's distinguished name (DN) when you create a Directory Server instance. The default and recommended DN is:

```
cn=Directory Manager,ou=Topology Manager,ou=your domain.com
```

Windows NT System Account

On Windows NT, you can run Calendar Server logged in as any valid user. Usually, however, you will want to run the server under the `system` account (logged in as administrator) to ensure that you have all account privileges required to manage the server. If you log in as a user other than administrator, you may not have the privileges you need to fully manage the server.

Unix System Accounts

The accounts described in this section apply only to Calendar Server installations on Unix platforms.

Calendar Server User and Group. (Unix only) These accounts are the user and group that make up the identity that the Calendar Server runs as. The calendar data files, such as the calendar data store, are owned by this user and group. This user and group should be highly secure and should be members of the Netscape group. For security reasons, these accounts should have no special privileges on the system. In the course of operation,

servers will assign some directory permissions to this user and the Netscape group for certain server-specific operations. The Calendar Server installation program suggests the default user `icsuser` and the default group `icsgroup` and will create these accounts automatically if they do not exist.

Installation Guidelines

This section contains information you need and suggested guidelines you should follow as you prepare for your Calendar Server 2.0 installation:

- **Host and domain names on Unix.** The installation program assumes that your host and domain names are defined correctly in your Unix system files. Make sure that both the `hostname` and `domainname` commands return the correct information before installing Calendar Server. If your system does not support the `domainname` command, check the domain name in the `/etc/resolv.conf` file.
- **Directories.** The Messaging, Directory, and Administration servers are installed into directories referred to as *server-roots*. These directories provide a known file location structure (file directory path) that enables the servers to identify one another.

Note Calendar Server 2.0 must be installed in its own unique *server-root*.

Transferring the Calendar Server Installation Files

To transfer and extract the product archive files which contain the installation program files in compressed format:

1. Locate the Calendar Server 2.0 archive (tar or zip) file for your platform:
 - On Unix, there are separate archive files for each supported operating system. The archive filename identifies the supported operating system.
 - On Windows NT, the installation archive file is a self-extracting zip file. For example: `ics20.zip`

You can find the archive file on the distribution CD or download it from the iPlanet Calendar Server online site.

2. Create a directory (such as `/tmp/calinstall`) on the machine that will host the Calendar Server and copy the archive file to that directory.
3. Go to the directory where you copied the archive file and then extract the contents.

On Unix, type

```
tar -xvf archive.tar
```

where *archive* identifies the name of the platform archive file you chose to transfer.

On Windows NT:

Double-click the self-extracting `ics20.exe` file included as part of your installation package.

Status messages display as the archive is unpacked.

Running the Installation Program

After you have extracted the installation files on the machine where you plan to install Calendar Server, you can run the installation program:

- On Unix, go to the directory location that contains the extracted installation files and type: `./setup`.
- On Windows NT, run the `setup.exe` program located in the directory where you extracted the installation files.

Note For the most up to date and complete installation instructions, refer to the online version listed at:

<http://developer.netscape.com/docs/manuals/calendar.html>

Deployment Considerations

The following sections describe some of the issues you should consider before you install and deploy Calendar Server at your site.

By default, Calendar Server supports users that are defined and maintained in an LDAP directory, such as Netscape Directory Server. Calendar Server also supports the use of plug-ins that you create to enable access to users defined in non-LDAP directories such as those stored in a standard Unix Authentication format or in a Windows NT User Manager database.

- If your users are already stored in an LDAP directory, the simplest solution to deploy Calendar Server is to upgrade your directory server to Netscape Directory Server 4.1 which includes the schema extension that enables users to access Calendar Server data. Otherwise, you can modify your directory schema manually to allow your users to access to Calendar Server data. For more information on how to modify a directory schema for Calendar Server, see
- If your users are stored in a manner other than LDAP directory format, such as an OS specific or proprietary user authentication method, you can create a pug-in using the CSAPI to authenticate users and give them access to data hosted on the Calendar Server.

Configuring Calendar Server

This chapter describes the settings you can configure for Calendar Server 2.0 and includes the following sections:

- Using the Server.conf file
- Database Administration
- Calendar Server API
- Notification Messages
- Counters

Using the Server.conf file

Calendar Server configuration settings are stored in file:

```
server-root/bin/cal/config/server.conf
```

This file is a plain ASCII text file, with each line defining a server parameter and its value:

- All parameters are in all lower case.
- A parameter and its value(s) are separated by a space or tab.
- Any value with spaces must be enclosed in single quotes (" ").

- A comment line must have a pound sign (#) as the first character of the line.
- If a parameter appears more than once, the values of the last parameter listed override previous values.

Note The `server.conf` file is initialized by the installation process and should be modified only as described in this manual.

Table 2.1 describes the local configuration settings stored in `server.conf`.

Table 2.1 Local Configuration Settings stored in `server.conf`

Parameter	Default Value	Description
<code>local.authldapbasedn</code>	" "	BASEDN for LDAP authentication. If not specified, <code>local.ugldapbasedn</code> is used.
<code>local.authldapghost</code>	"localhost"	Host for LDAP authentication. If not specified, <code>local.ugldapghost</code> is used.
<code>local.authldapbindcred</code>	" "	Bind credentials (password) for <code>local.authldapbinddn</code> .
<code>local.authldapbinddn</code>	" "	DN used to bind to LDAP authentication host to search for user's dn. If not specified or "", anonymous bind.
<code>local.authldapport</code>	"389"	Port for LDAP authentication. If not specified, <code>local.ugldapport</code> is used.
<code>local.authldappoolsize</code>	"1"	Minimum number of LDAP client connections that are maintained for LDAP authentication. If not specified, <code>local.ugldappoolsize</code> is used.
<code>local.authldapmaxpool</code>	"1024"	Maximum number of LDAP client connections that are maintained for LDAP authentication. If not specified, <code>local.ugldapmaxpool</code> is used.
<code>local.caldb.deadlock.autodetect</code>	"no"	Periodically checks if the Berkeley database is in a deadlock state and, if so, instructs the database to reset.
<code>local.hostname</code>	" "	Name of Calendar Server host.
<code>local.installdir</code>	" "	Directory where Calendar Server is installed.
<code>local.instancedir</code>	". "	Directory where programs and data for this instance of a Calendar Server is installed.
<code>local.pluginidir</code>	" "	Directory where CSAPI plugins are installed.
<code>local.serveruid</code>	"icsuser"	UID for Calendar Server files, e.g., counters and logs.

Table 2.1 Local Configuration Settings stored in server.conf (Continued)

Parameter	Default Value	Description
<code>local.servergid</code>	"icsgroup"	GID for Calendar Server files, e.g., counters and logs.
<code>local.sitelanguage</code>	"en"	Default language for this Calendar Server.
<code>local.supportedlanguages</code>	"en"	User languages supported by Calendar Server.
<code>local.ugldapbasedn</code>	" "	BASEDN for LDAP user preferences Must be specified and cannot be "".
<code>local.ugldaphost</code>	"localhost"	Host for LDAP user preferences.
<code>local.ugldapport</code>	"389"	Port for LDAP user preferences.
<code>local.ugldappoolsize</code>	"1"	Minimum number of LDAP client connections that are maintained for LDAP user preferences.
<code>local.ugldapmaxpool</code>	"1024"	Maximum number of LDAP client connections that are maintained for LDAP user preferences.
<code>local.enduseradmincred</code>	" "	Bind credentials (password) for LDAP user preferences authentication.
<code>local.enduseradmindn</code>	" "	DN used to bind to LDAP user preferences host. Must be specified. If "" (not specified), anonymous bind is assumed.

Table 2.2 describes the calendar store configuration settings stored in `server.conf`.

Table 2.2 Calendar Store Configuration Settings in server.conf

Parameter	Default Value	Description
<code>calstore.anonymous.calid</code>	"anonymous"	Calendar ID (CALID) used for anonymous logins.
<code>calstore.default.timezoneID</code>	"America/Los_Angeles"	<p>Timezone-id to be used when importing files. This server preference should be used as a last resort value only when:</p> <ul style="list-style-type: none"> no timezone-id supplied no calendar timezone-id found no user timezone-id found <p>An invalid value defaults causes the server to use to the GMT (Greenwich Mean Time) timezone.</p>
<code>calstore.recurrence.bound</code>	"60"	Maximum number of events that be created by a recurrence expansion.

Table 2.3 describes the calendar log information settings stored in `server.conf`.

Table 2.3 Calendar Log Settings in server.conf

Parameter	Default Value	Description
<code>logfile.admin.logname</code>	"admin.log"	Name of log file for logging administrative tools.
<code>logfile.buffersize</code>	"0"	Size of log buffers in bytes.
<code>logfile.expirytime</code>	"604800"	Number of seconds before log files are expired.
<code>logfile.flushinterval</code>	"60"	Number of seconds for flushing buffers to log files.
<code>logfile.logdir</code>	"logs"	Directory path for log files.

Table 2.3 Calendar Log Settings in server.conf

Parameter	Default Value	Description
logfile.loglevel	"Notice"	Determines which level of detail will be logged by the server. Each log entry is assigned one of the following levels (starting with the most severe): <i>Critical</i> , <i>Error</i> , <i>Warning</i> , <i>Notice</i> , <i>Information</i> , and <i>Debug</i> . If you set this preference to <i>Critical</i> , the server will log the least amount of detail. If you want the server to log the most amount of detail, specify <i>Debug</i> . For example, if you specify <i>Warning</i> , only <i>Critical</i> , <i>Error</i> , and <i>Warning</i> level log entries are logged.
logfile.maxlogfiles	"10"	Maximum number of log files in log directory.
logfile.maxlogfilesize	"2097152"	Maximum size in bytes of each log file.
logfile.maxlogsize	"20971520"	Maximum disk space in bytes for all log files.
logfile.minfreediskspace	"5242880"	Minimum free disk space in bytes that must be available for logging. When this is reached, action will be taken to free up disk space by expiring old log files. All logging will be paused if no space can be freed up.
logfile.rollovertime	"86400"	Number of seconds before log files are rotated.
logfile.http.logname	"server.log"	Name of current log file.
logfile.http.access.logname	"httpd.access"	Name of current http access log file.

Table 2.4 describes the services configuration settings stored in `server.conf`.

Table 2.4 Services Configuration Settings in server.conf

Parameter	Default Value	Description
service.http.attachdir	."	Directory relative to local.queuedir where imported files are temporarily stored.
service.authcachesize	"10000"	Maximum number of authenticated user UIDs and passwords that Calendar Server will maintain.
service.authcachettl	"900"	Number of seconds that a user UID and password are cached.

Table 2.4 Services Configuration Settings in server.conf (Continued)

Parameter	Default Value	Description
<code>service.dnsresolveclient</code>	"no"	If <code>yes</code> , client IP addresses are checked against DNS if allowed HTTP access.
<code>service.http.admin.enable</code>	"no"	If <code>yes</code> , listen on <code>service.http.admin.port</code> for HTTP requests.
<code>service.http.admin.port</code>	" "	Port for HTTP requests from Calendar Server administration.
<code>service.http.admins</code>	" "	Space separated list of user UIDs that can send in WCAP administration commands.
<code>service.http.allowadminproxy</code>	"no"	If <code>yes</code> , allow login via proxy.
<code>service.http.allowanonymouslogin</code>	"no"	If <code>yes</code> , allow anonymous (no login) access.
<code>service.http.calendarhostname</code>	" "	HTTP host for retrieving HTML documents. If not specified, the local HTTP host is used.
<code>service.http.calmaster.admin</code>	" "	USERID of the person designated as Calendar Server administrator. This value is supplied at installation and is required by the installation program. The default is <code>calmaster</code> .
<code>service.http.calmaster.cred</code>	" "	Password of the USERID specified as the Calendar Server administrator. This value is supplied at installation and is required by the installation program.
<code>service.http.dbcachesize</code>	"8388608"	Berkeley DB cache size for HTTP sessions.
<code>service.http.dbtmpdir</code>	" "	Temporary directory for HTTP sessions.

Table 2.4 Services Configuration Settings in server.conf (Continued)

Parameter	Default Value	Description
<code>service.http.domainallowed</code>	" "	If specified and not " ", filter for allowing access based on TCP domains. For example, "ALL: LOCAL .airius.com" would allow local HTTP access and anyone in the <code>airius.com</code> domain. Multiple filters are separated by CR-LF (line feed).
<code>service.http.domainnotallowed</code>	" "	If specified and not " ", filter for not allowing access based on TCP domains. For example, "ALL: LOCAL .airius.com" would deny HTTP access to anyone from <code>airius.com</code> . Multiple filters are separated by CR-LF.
<code>service.http.ipsecurity</code>	"yes"	If <code>yes</code> , all requests that reference an existing session are verified as originating from the same IP address.
<code>service.http.enable</code>	"yes"	If <code>yes</code> , listen on <code>service.http.port</code> for HTTP requests.
<code>service.http.idletimeout</code>	"120"	Number of seconds before timing out an HTTP connection.
<code>service.http.ldap.enable</code>	"yes"	If <code>yes</code> , LDAP connections for authentication and userprefs are created and maintained.
<code>service.http.logaccess</code>	"no"	If <code>yes</code> , http accesses to server are fully logged.
<code>service.http.maxsessions</code>	"5000"	Maximum number of WCAP sessions
<code>service.http.maxthreads</code>	"1000"	Maximum number of threads to service HTTP requests.
<code>service.http.numprocesses</code>	"1"	Number of processes to service HTTP requests.
<code>service.http.port</code>	"80"	Port for HTTP requests from Calendar Server users.

Table 2.4 Services Configuration Settings in server.conf (Continued)

Parameter	Default Value	Description
<code>service.http.proxydomainallowed</code>	" "	If specified and not " ", filter for allowing proxy login based on TCP domains. Same syntax as <code>service.http.domainallowed</code> .
<code>service.http.sessiontimeout</code>	"1800"	Number of seconds before timing out WCAP session.
<code>service.http.sourceurl</code>	" "	Directory relative to executable where all URL references to files are stored.
<code>service.ldapmemcache</code>	"no"	If <code>yes</code> , use cache in LDAP SDK.
<code>service.ldapmemcachettl</code>	"30"	If <code>service.ldapmemcache</code> is "yes", pass in this value to the LDAP SDK. This is the maximum number of seconds that an item can be cached. If 0, there is no limit to the amount of time that an item can be cached.
<code>service.ldapmemcachesize</code>	"131072"	If <code>service.ldapmemcache</code> is "yes", pass in this value to the LDAP SDK. This is the maximum amount of memory in bytes that the cache will consume. If 0, the cache has no size limit.
<code>service.plaintextloginpause</code>	"0"	Number of seconds to delay after successfully authenticating a user using plain text passwords.
<code>service.wcap.anonymous.allowpubliccalendarwrite</code>	"yes"	If <code>yes</code> , allow anonymous users to write to publicly writable calendars.
<code>service.wcap.allowcreatecalendars</code>	"yes"	If <code>yes</code> , allow calendars to be created.
<code>service.wcap.allowdeletecalendars</code>	"yes"	If <code>yes</code> , allow calendars to be deleted.
<code>service.wcap.allowchangepassword</code>	"no"	If <code>yes</code> , allow users to change their passwords via this server.
<code>service.wcap.allowpublicwritablecalendars</code>	"yes"	If <code>yes</code> , allow users to have publicly writable calendars.

Table 2.4 Services Configuration Settings in server.conf (Continued)

Parameter	Default Value	Description
<code>service.wcap.allowsetprefs.cn</code>	"no"	If <code>yes</code> , allow <code>set_userprefs.wcap</code> to modify the user preference "cn" (LDAP user's common name).
<code>service.wcap.allowsetprefs.givenname</code>	"no"	If <code>yes</code> , allow <code>set_userprefs.wcap</code> to modify the user preference "givenname" (LDAP user's given name)
<code>service.wcap.allowsetprefs.mail</code>	"no"	If <code>yes</code> , allow <code>set_userprefs.wcap</code> to modify the user preference "mail" (user's e-mail address).
<code>service.wcap.allowsetprefs.preferredlanguage</code>	"no"	If <code>yes</code> , allow <code>set_userprefs.wcap</code> to modify the user preference, "preferredlanguage" (LDAP user's preferred language).
<code>service.wcap.allowsetprefs.sn</code>	"no"	If <code>yes</code> , allow <code>set_userprefs.wcap</code> to modify the user preference "sn" (LDAP user's surname).
<code>service.wcap.allowsetprefs.nswccalid</code>	"no"	If <code>yes</code> , allow <code>set_userprefs.wcap</code> to modify the user preference "nswccalid", which is the user's default calendar identifier
<code>service.wcap.login.calendar.publicread</code>	"no"	If <code>yes</code> , default user calendars are initially set to public read/private write. If <code>no</code> , default user calendars are initially set to private read/private write.
<code>service.wcap.validateowners</code>	"no"	If <code>yes</code> , the server must validate that each owner of a calendar exists in the directory (through LDAP or a CSAPI compatible user directory mechanism).

Database Administration

Table 2.5 shows the Calendar Server database configuration settings stored in `server.conf`.

Table 2.5 Database Configuration Settings in `server.conf`

Parameter	Default Value	Description
<code>caldb.bcounters</code>	"yes"	If <code>yes</code> , data base statistics (reads, writes, deletes) will be counted.
<code>caldb.bserveralarms</code>	"yes"	If <code>yes</code> , alarm emails will be sent.
<code>caldb.pssmtphost</code>	"localhost"	Send alarm emails to this SMTP host.
<code>caldb.pssmtpport</code>	"25"	Port for SMTP host.
<code>caldb.berkeleydb.ideadlockinterval</code>	"100"	Number of milliseconds between checking for database deadlocks that need to be broken.
<code>caldb.berkeleydb.pscaldblogdirectory</code>	". "	Directory relative to executable where database checkpoint files are kept. The default value of ". " specifies that these files are stored in the <code>server-root/cal/bin</code> directory.
<code>caldb.berkeleydb.pscaldbhomedirectory</code>	". "	Directory (relative to the location of the program executable files) where database event, task, and alarm files are kept. The default value of ". " specifies that these files are stored in the <code>server-root/cal/bin</code> directory.
<code>caldb.berkeleydb.icheckpointinterval</code>	"60"	Number of seconds between checkpointing database transactions.
<code>caldb.berkeleydb.bcircularlogging</code>	yes"	If <code>yes</code> , remove database checkpoint files after their transactions are synchronized.
<code>caldb.berkeleydb.mempoolsizemb</code>	"4"	Megabytes of shared memory for database environment.
<code>caldb.berkeleydb.logfilesizemb</code>	"10"	Maximum megabytes for a database checkpoint file.
<code>caldb.berkeleydb.maxthreads</code>	"10000"	Maximum number of threads that database environment must be prepared to accommodate.

Table 2.5 Database Configuration Settings in `server.conf` (Continued)

<code>caldb.calmaster</code>	<code>" "</code>	Email for user/alias that is responsible to administering the database.
<code>caldb.counters.maxinstances</code>	<code>100</code>	Maximum number of calendars that can have counters. A calendar is enabled for counters using the <code>cscalc</code> command line utility.

Calendar Server API

Table 2.6 shows the Calendar Server API (CSAPI) configuration settings stored in `server.conf`.

Table 2.6 CSAPI Configuration Settings in `server.conf`

Parameter	Default Value	Description
<code>csapi.plugin.loadall</code>	<code>"no"</code>	<p>If <code>yes</code>, load all plugins found in the plugins directory. For NT, these plugins have a <code>.dll</code> file name extension. For IRIX, these plugins have an <code>.sl</code> extension. For all other UNIX systems, these plugins have an <code>.so</code> extension.</p> <p>If <code>no</code>, only load the specific class of plugins flagged by their respective parameters. For example, set <code>csapi.plugin.authentication</code> to <code>yes</code> to load authentication class plugins.</p>
<code>csapi.plugin.authentication</code>	<code>"no"</code>	<p>If <code>yes</code>, load only the plugin specified in <code>csapi.plugin.authentication.name</code> or if not specified, load all authentication class plugins in alphabetical order. For authentication, use each of these plugins in alphabetical order.</p>

Table 2.6 CSAPI Configuration Settings in server.conf (Continued)

Parameter	Default Value	Description
<code>csapi.plugin.authentication.name</code>	<code>" "</code>	If <code>csapi.plugin.loadall</code> is <code>no</code> and <code>csapi.plugin.authentication</code> is <code>yes</code> , this parameter is used. If not specified or <code>" "</code> , load all authentication class plugins. Otherwise, only load this specific plugin.
<code>csapi.plugin.database</code>	<code>"yes"</code>	If <code>yes</code> , load only the plugin specified in <code>csapi.plugin.database.name</code> or if not specified, load all database class plugins in alphabetical order. For database access, use each of these plugins in alphabetical order.
<code>csapi.plugin.database.name</code>	<code>"cs_caldb_berkeley10"</code>	If <code>csapi.plugin.loadall</code> is <code>no</code> and <code>csapi.plugin.database</code> is <code>yes</code> , this parameter is used. If not specified or <code>" "</code> , load all database class plugins. Otherwise, only load this specific plugin.
<code>csapi.plugin.datatranslator</code>	<code>"yes"</code>	If <code>yes</code> , load only the plugin specified in <code>csapi.plugin.datatranslator.name</code> or if not specified, load all data translator class plugins in alphabetical order. For data translation, use each of these plugins in alphabetical order.
<code>csapi.plugin.datatranslator.name</code>	<code>"cs_datatranslatorcsv10"</code>	If <code>csapi.plugin.loadall</code> is <code>no</code> and <code>csapi.plugin.data translator</code> is <code>yes</code> , this parameter is used. If not specified or <code>" "</code> , load all data translator class plugins. Otherwise, only load this specific plugin.

Table 2.6 CSAPI Configuration Settings in `server.conf` (Continued)

Parameter	Default Value	Description
<code>csapi.plugin.userprefs</code>	"no"	If <code>yes</code> , load only the plugin specified in <code>csapi.plugin.userprefs.name</code> or if not specified, load all user preferences class plugins in alphabetical order. For user preferences, use each of these plugins in alphabetical order.
<code>csapi.plugin.userprefs.name</code>	" "	If <code>csapi.plugin.loadall</code> is <code>no</code> and <code>csapi.plugin.userprefs</code> is <code>yes</code> , this parameter is used. If not specified or " ", load all user preferences class plugins. Otherwise, only load this specific plugin.

Notification Messages

Calendar Server sends several types of email messages described in Table 2.7. The format of these messages is controlled by the associated format (`.fmt`) file listed in the table. Format files are located in specific directories for each local (such as `/en` for English and `/fr` for French). These directories are located in the `server-root/cal/bin/config` directory. For example, the English version of the task alarm message format is specified in the file:

```
/opt/SUNWicsrv/cal/bin/config/en/mail_todoalarm.fmt
```

Table 2.7 shows the Calendar Server mail settings stored in `server.conf`.

Table 2.7 Calendar Server mail formats

Message Type	Parameter	Format File (default)	Description	Recipients
Event Publish	<code>calmail.imipEventPublish.fname</code>	<code>mail_eventpublish.fmt</code>	Announces an event or a change to an existing event	Those listed in Notification
Event Cancel	<code>calmail.imipEventCancel.fname</code>	<code>mail_eventcancel.fmt</code>	Announces an event cancellation	Those listed in Notification

Table 2.7 Calendar Server mail formats (Continued)

Message Type	Parameter	Format File (default)	Description	Recipients
Event Alarm	<code>calmail.eventReminder.fname</code>	<code>mail_eventreminder.fmt</code>	Reminder for an upcoming event	Those listed in Reminder
Task Publish	<code>calmail.imipToDoPublish.fname</code>	<code>mail_todopublish.fmt</code>	Announces an task or a change to an existing task	Those listed in Notification
Task Cancel	<code>calmail.imipToDoCancel.fname</code>	<code>mail_todocancel.fmt</code>	Announces an task cancellation	Those listed in Notification
Task Alarm	<code>calmail.todoReminder.fname</code>	<code>mail_todoalarm.fmt</code>	Reminder for an upcoming task	Those listed in Reminder

Calendar Server generates notification messages by combining a particular event or task with the contents of a format file. The values of data fields within an event or task can be output to the message. The notification message can also include MIME header lines and associated special values. Using special character sequences (format notations), you can include the values of events, tasks, and MIME headers in the message. The lines in the format file are format strings comprised of special character sequences that are replaced with actual values from calendar data fields when the mail message is generated. Special character sequences consist of two characters, the first is the percent sign (%) and the second represents the specific format notation.

The following sections describe special character sequences:

- Special Character Sequences for Events
- Special Character Sequences for Task Notification
- Special Character Sequences for Dates

Special Character Sequences for Events

Table 2.8 shows the special character sequences for Calendar Server event notifications.

Table 2.8 Special Character Sequences for Event Notifications

Format Code	Meaning
%A	EXDATES in ICAL format
%a	RDATES in ICAL format
%B	Start time (also see %Z)
%b	Output the start time and end time in ICAL format. If the start time has the parameter <i>value=date</i> , only the month/day/year portion of the date is output. If end time has the same month/day/year value as the start time, only the start time is generated.
%C	create time
%c	event class
%d	event description. (also see %F)
%E	end time (also see %Z)
%e	exception rules in ICAL format
%F	event description - folded line / ICAL format (also see %d)
%G	this event's geographic location (latitude and longitude)
%g	organizer's email address. (there is no guarantee as to the authenticity of this value.)
%K	organizer email in the form of a <code>mailto:url</code>
%k	alarm count
%L	the location
%l	recurrence rules in ICAL format
%M	modify time
%N	new line
%n	"now" (the current time stamp) used with DTSTAMP
%P	priority
%r	recurrence id (blank if this event does not recur)
%S	event sequence number
%s	summary
%t	event status
%U	UID (user ID)

Table 2.8 Special Character Sequences for Event Notifications (Continued)

Format Code	Meaning
<code>%u</code>	URL to the event
<code>%Z</code>	used in conjunction with the time field code to force the time to be rendered in UTC. (%B displays the start time in local time whereas %ZB displays the start time in UTC time.)
<code>%%</code>	displays the percent (%) character
<code>%(sub-format code)</code>	specifies a subformat for the data identified by code. (For details, see “Date Sub-Formatting” on page 40.)

Date Sub-Formatting

Date-time values can be formatted in many different ways. Using a sub-format, you can provide additional information to describe how a date/time value should be formatted. If a sub-format is not specified, the server uses a default format to output the date. Using a sub-format field allows you to specify the exact format to be used.

For example, `%B` specifies that the output string includes the event’s begin time. This default format prints out the date, time, the time zone, and everything possible about the date. The sub-format string for date values is a `strftime` format string (see “Special Character Sequences for Dates” on page 43. If you were only interested in the month and year of the start time, instead of `%B`, you would use: `%(%m %Y)B`.

Example

```
The event begins %B%N
```

```
The event ends: %( %b %d, %Y %I:%M %p)E%N
```

The above example produces output that resembles the following notification:

```
The event begins Feb 02, 1999 23:30:00 GMT Standard Time
```

```
The event ends Feb 03, 1999 02:30 AM
```

Conditional Printing

Sometimes it is desirable to have a line print only under certain conditions. For example, the following lines:

```
title: %S%N
start: %B%N
end: %E%N
```

produce output that resembles the following notification:

```
title: Staff Meeting
start: Feb 04, 1999 09:00:00
end: Feb 04, 1999 10:00:00
```

There are two conditions, however, where the above example would yield misleading or incorrect results:

- if the event has no end time
- if the event is an "all-day" event, that starts and ends on the same day.

In these situations, it is best not to print the end time at all. By default, only the year, month, and day are printed when a time stamp has the attribute of being `all-day`. Furthermore, if an event start time has the `all-day` attribute and the event ends on the same day as it starts, a special conditional flag is set. Use the `?` modifier to print conditional values only when the special conditional flag is not set.

For example, if you change the lines in the above example to:

```
title: %S%N
start: %B%N
end: %?E%N
```

the last line will not be printed for all-day events for which the start day and end day are the same. It produces the following output for typical all-day events (such as birthdays or anniversaries):

```
title: Staff Meeting
start: Feb 04, 1999
```

The `?` flag can be combined with other modifiers. For example:

```
The event ends: %?(%b %d, %Y %I:%M %p)E%N
```

Special Character Sequences for Task Notification

Table 2.9 shows the special character sequences for Calendar Server task notifications.

Table 2.9 Special Character Sequences for Tasks Notifications

Format Code	Meaning
%A	exdates in ICAL format
%a	rdates in ICAL format
%B	Start time (also see %Z)
%C	create time
%c	Event class
%D	Due date/time.
%d	Event description. (also see %F)
%E	Due date/time in IMIP format
%e	Exception rules in ICAL format
%F	Event Description - folded line / ICAL format (also see %d)
%G	this event's geographic location, the latitude and longitude.
%g	Organizer's email address (cannot guarantee the authenticity of this value)
%K	Organizer Email in the form of a <code>mailto:URL</code>
%k	Alarm count
%L	the location
%l	recurrence rules in ICAL format
%M	Modify time
%N	a new line
%n	"now" (the current time stamp and used with DTSTAMP)
%P	priority
%r	the recurrence id (blank if this event does not recur)
%S	is event's Sequence Number
%s	Summary
%t	the status
%U	the UID
%u	URL to the event

Table 2.9 Special Character Sequences for Tasks Notifications (Continued)

Format Code	Meaning
%Z	used in conjunction with time field code to force the time to be rendered in UTC (%B displays the start time in local time whereas %ZB displays the start time in UTC time)
%%	displays the % character
% (sub-format code)	specify a sub-format for the data identified by code (for details, see “Date Sub-Formatting” on page 40)

Special Character Sequences for Dates

Table 2.10 shows the special character sequences for date formatting in Calendar Server notifications.

Note The special date format codes appear in this section only for convenience. Calendar Server does not re-write any of the `strftime` code, but simply uses the operating system implementation.

Table 2.10 Special Character Sequences for Dates

Format Code	Meaning
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for locale
%d	Day of month as decimal number (01 - 31)
%H	Hour in 24-hour format (00 - 23)
%I	Hour in 12-hour format (01 - 12)
%j	Day of year as decimal number (001 - 366)
%m	Month as decimal number (01 - 12)
%M	Minute as decimal number (00 - 59)
%p	Current locale's A.M./P.M. indicator for 12-hour clock
%S	Second as decimal number (00 - 59)

Table 2.10 Special Character Sequences for Dates (Continued)

Format Code	Meaning
%U	Week of year as decimal number, with Sunday as first day of week (00 - 53)
%w	Weekday as decimal number (0 - 6; Sunday is 0)
%W	Week of year as decimal number, with Monday as first day of week (00 - 53)
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without century, as decimal number (00 - 99)
%Y	Year with century, as decimal number
%z	Time-zone name or abbreviation; no characters if time zone is unknown
%%	Percent sign

Simple Event Reminder Example

The following example shows the default event reminder message format:

```

1  EVENT REMINDER
2  ~~MIME-Version: 1.0%N
3  ~~Content-Type: text/plain; charset=%s%N
4  ~~Content-Transfer-Encoding: %x%N%N
5      Summary: %s%N
6          Start: (%a, %d %b %Y %I:%M %p)B%N
7          End: (%a, %d %b %Y %I:%M %p)E%N
8      Location: %L%N%N
9      Description: %N%d%N

```

1. Line 1 is the message subject.
2. Line 2 begins with `~~` which indicates that it is a MIME wrapper line. That is, the replacement of special character sequences are those associated with an internal MIME object rather than an event or task. The special sequence `%N` is a line feed. The subject line does not need the special new line sequence, while all other lines do.
3. Line 3 is also a MIME header line. It contains the special character sequence `%s` which will be replaced by the character set associated with the event or task being mailed.

4. Line 4 is the last MIME line, %x is the content transfer encoding string needed for this message.
5. Line 5 lists the event summary and calls out the event summary with %s.
6. Line 6 lists the event start time. It makes use of a sub-format string on the special character sequence %B. For details, see “Date Sub-Formatting” on page 40.
7. Line 7 lists the event end time.
8. Line 8 lists the location of the event.
9. Line 9 lists the description of the event.

The following sample resembles the notification message generated by the above example:

```
From: jsmith@airius.com (James Smith)
Date: Wed, 15 Nov 1999 19:13:49
To: jsmith@airius.com
Subject: EVENT REMINDER
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

```
Summary: smtp_rig event 1
Start: Tues, 16 Nov 1999 02:00 PM
End: Tues, 16 Nov 1999 03:00 PM
Location: Green Conference Room
Description:
This is the description for a randomly generated event.
```

Complex Event Reminder Example

The following example shows a more complex multipart message. It has a human-readable text part and an IMIP PUBLISH part.

```
EVENT PUBLICATION
~MIME-Version: 1.0%N
~Content-Type: multipart/mixed; boundary="%b"%N%N
This is a multi-part message in MIME format.%N
~--%b%N
~Content-Type: text/plain; charset=%s%N
~Content-Transfer-Encoding: %x%N%N
Summary: %s%N
Start: %(%a, %d %b %Y %I:%M %p)B%N
End: %(%a, %d %b %Y %I:%M %p)E%N
Location: %L%N%N
Description: %N%d%N%N
~--%b%N
~Content-Type: text/calendar; method=%m; component=%c; charset=%s%N
~Content-Transfer-Encoding: %x%N%N
BEGIN:VCALENDAR%N
PRODID://iPlanet/Calendar Hosting Server//EN%N
METHOD:PUBLISH%N
VERSION:2.0%N
BEGIN:VEVENT%N
ORGANIZER:%K%N
```

```

DTSTAMP: %Zn%N
DTSTART: %ZB%N
DTEND: %ZE%N
SUMMARY: %s%N
UID: %U%N
%R
%A
%a
%e
%l
SEQUENCE: %S%N
LOCATION: %L%N
GEO: %G%N
%F
STATUS: %t%N
END: VEVENT%N
END: VCALENDAR%N
----%D--

```

Counters

Calendar Server counters (statistics) configuration information is stored in the file:

```
server-root/bin/cal/config/counter.conf
```

This file is a plain ASCII text file with each line defining a counter and its parameters: name, type, size, scale, and description.

The first part of a counter's name identifies the *counterobject* used with the `cscounter` and `csstats` command line utilities:

- A parameter with spaces must be enclosed in single quotes ("").
- A comment line must have a pound sign (#) as the first character of the line.

For more information on the command line utilities, see Chapter 3, "Administering Calendar Server" and Appendix A, "Command Line Utilities."

Note Do not modify the `counter.conf` file unless instructed to do so by customer support staff.

Table 2.11 shows the HTTP counter objects (`httpstat`) stored in `counter.conf`.

Table 2.11 httpstat counters in counter.conf

Name	Type	Size	Scale	Description
httpstat.currentStartTime	TIME	4	0	When Calendar Server was started.
httpstat.lastConnectionTime	TIME	4	0	Last time new client connection was accepted
httpstat.maxConnections	COUNTER	4	4	Maximum number of concurrent connections served.
httpstat.maxSessions	COUNTER	4	4	Maximum number of WCAP sessions served.
httpstat.numConnections	COUNTER	4	5	Total number of connections served.
httpstat.numCurrentConnections	GAUGE	4	2	Current number of active connections.
httpstat.numCurrentSessions	GAUGE	4	2	Current number of WCAP sessions.
httpstat.numFailedConnections	COUNTER	4	3	Total number of failed connections served.

Table 2.12 shows the authentication counter objects (authstat) stored in counter.conf

Table 2.12 authstat counters in counter.conf

Name	Type	Size	Scale	Description
authstat.lastLoginTime	TIME	4	0	Last time a user logged in.
authstat.numSuccessfulLogins	COUNTER	4	3	Total number of successful logins served.
authstat.numFailedLogins	COUNTER	4	3	Total number of failed logins served.

Table 2.13 shows the WCAP counter objects (wcapstat) stored in counter.con

Table 2.13 ttpstat counters in counter.conf

Name	Type	Size	Scale	Description
authstat.lastLoginTime	TIME	4	0	Last time a user logged in.
authstat.numSuccessfulLogins	COUNTER	4	3	Total number of successful logins served.
authstat.numFailedLogins	COUNTER	4	3	Total number of failed logins served.

Table 2.14 shows the WCAP counter object (`wcapstat`) stored in `counter.conf`.

Table 2.14 wcapstat counters in counter.conf

Name	Type	Size	Scale	Description
wcapstat.numRequests	COUNTER	4	3	Total number of WCAP requests.

Table 2.15 shows the database counter object (`dbstat`) stored in `counter.conf`.

Table 2.15 Database counters in counter.conf

Name	Type	Size	Scale	Description
dbstat.numReads	COUNTER	4	3	Total number of database reads.
dbstat.numWrites	COUNTER	4	3	Total number of database writes.
dbstat.numDeletes	COUNTER	4	3	Total number of database deletes.
dbstat.lastReadTime	TIME	4	0	Last time database did a read.
dbstat.lastWriteTime	TIME	4	0	Last time database did a write.
dbstat.lastDeleteTime	TIME	4	0	Last time database did a delete.

Administering Calendar Server

This chapter describes how to administer and maintain your Calendar Server. It includes the following sections:

- Administration Tools
- Starting and Stopping the Server
- Using the Command Line Utilities
- Backup and Restore Procedures

Administration Tools

iPlanet Calendar Server provides the following tools for server administration:

- utilities to start and stop the Calendar Server
- command line utilities to enable and disable access to calendar data, to manage calendar users, events, and tasks, to monitor Calendar Server activity, and to backup and restore calendar data.
- command line utilities to control the server

Starting and Stopping the Server

You can start and stop the Calendar Server the using:

- `start-cal` and `stop-cal`
- `iplncal.sh` (Unix only)
- the Services window (Windows NT only)

Using `start-cal` and `stop-cal`

The `start-cal` and `stop-cal` utilities are located in the `server-root/cal/bin` directory. Both can be run either on a remote machine or the local machine where the Calendar Server is installed.

To start the Calendar Server, go to the `server-root/cal/bin` directory and type: `start-cal` on the command line. For example:

1. `cd /opt/SUNWicsrv/cal/bin`
2. Type: `start-cal`

To stop the Calendar Server, go to the `server-root/cal/bin` directory and type: `stop-cal` on the command line. For example, enter:

1. `cd /opt/SUNWicsrv/cal/bin`
2. Type: `stop-cal`

Using `iplncal.sh` (Unix only)

The `iplncal.sh` program is located in `/etc` directory. It must be run on the local machine where the Calendar Server is installed.

To start the Calendar Server using `iplncal.sh`, go to the `/etc` directory and execute the program from the command line. It takes the following syntax:

```
iplncal.sh [-h] [-r ServerRoot] start|refresh|stop|list [service]
```

where:

- `-h` displays help showing the available commands
- `-r` is an optional parameter that lets you specify a directory location and *ServerRoot* represents the actual path in which the Calendar Server is installed, such as `opt/SUNWicsrv`.
- `start|refresh|stop|list` are the available options (of which you must choose one, but only one each time you issue the command) to start, stop, or refresh the server, or to list the server processes (threads) currently running.
- *service* is an optional parameter that specifies the Calendar Server service on which the command will take effect, such as HTTP.

Using the Services Panel (Windows NT only)

On Windows NT, you can open the Services dialog box from the Control Panel to start and stop the Calendar Server. Refer to the documentation provided with the operating system for instructions.

Using the Command Line Utilities

This section describes the tasks you can perform to administer Calendar Server using the command line utilities. The examples in this section use the most commonly used commands and options. For complete details on the command utilities syntax and usage, see Appendix A, “Command Line Utilities.”

This section includes the following topics:

- Managing Calendar Users
- Managing Calendars
- Managing Calendar Events and Tasks
- Managing Calendar Databases
- Importing and Exporting Calendar Data

Table 3.1 lists the command line utilities and their purpose.

Table 3.1 Command Line Utilities

Use...	To...
<code>csbackup</code>	Backup individual calendars, users, and the calendar database.
<code>cscal</code>	Manage calendars and their properties.
<code>cscomponents</code>	Manage components (events and tasks) in a calendar.
<code>csdb</code>	Manage calendar databases.
<code>csexport</code>	Export a calendar in ICAL or XML format.
<code>csimport</code>	Import a calendar in ICAL or XML format.
<code>csrestore</code>	Restore individual calendars, users, and the calendar database.
<code>cssession</code>	Manage user sessions in a Calendar Server.
<code>csstart</code>	Start the Calendar Server.
<code>csstats</code>	Display counters in a Calendar Server.
<code>csstop</code>	Stop the Calendar Server.
<code>cstool</code>	General purpose tool to ping a Calendar Server, to control and monitor the server, and to force the server to refresh its configuration.
<code>csuser</code>	Manage calendar users.

Managing Calendar Users

The following command line utilities let you perform administrative tasks on calendar users.

Use the `csuser` command line utility to:

- create a calendar user (see “Creating a Calendar User” on page 53)
- delete a calendar user (see “Displaying User Information” on page 55)
- enable a calendar user (see “Enabling a Calendar User” on page 54)
- disable a calendar user (see “Disabling a Calendar User” on page 54)
- check if a user is enabled for calendaring (see “Deleting a Calendar User” on page 53)
- list all calendar users or list a specified calendar user's configuration attributes (see “Displaying User Information” on page 55)

- reset the calendar user's configuration attributes to the default values (see “Resetting User Calendar Attributes” on page 55)

Use the `csession` command line utility to:

- list all logged-in users or the status of a specified user (see “Managing User Sessions” on page 69)
- logout a user (see “Logging Out a User” on page 70)

Creating a Calendar User

Use the `csuser` utility's `create` command to create a calendar user. For example, to create the calendar user `JSmith` using the default configuration settings, enter the following command on the command line:

```
csuser -g John -s Smith -y password -c calJSmith create JSmith
```

where:

- `-g John` represents the user's givenname (first name).
- `-s Smith` represents the user's surname (last name).
- `-y password` represents the user's password.
- `-c calJSmith` represents the user's default calendar ID (`calID`).
- `create JSmith` adds an entry with the `USERID JSmith` to the LDAP directory that stores your user information.

Deleting a Calendar User

Use the `csuser` utility's `delete` command to delete a user from the Calendar Server.

Note The `delete` command removes all of the user's information from the LDAP server. (Calendar database information can be recovered from a calendar database that has been saved to a backup. See “Backing-Up the Calendar Database” on page 64.)

For example, to delete `JSmith` from the Calendar Server, enter the following command:

```
csuser delete JSmith
```

Enabling a Calendar User

Users are automatically enabled for calendaring when they log on to the Calendar Server for the first time, if they are not already enabled. In such cases, the calendar ID (CALID) of the user's default calendar is the user's USERID. For example, if JSmith is not enabled for calendaring and logs on to the Calendar Server for the first time, the server enables JSmith for calendaring automatically and assigns this user a default calendar with the CALID `JSmith`. Additionally, the prefix `JSmith` is assigned to the CALID of each subsequent calendar that JSmith creates. For example, if JSmith, who was enabled automatically, creates a calendar called `sports`, the assigned CALID is `JSmith:sports`.

Use the `csuser` utility's `enable` command to let a user access Calendar Server data and to assign specific configuration settings such as a user's default calendar. For example, to enable JSmith for calendaring and assign a default calendar, enter the following command:

```
csuser -c calendarA enable JSmith
```

The above command enables JSmith to log on to the Calendar Server and creates a default calendar for this user with the CALID `calendarA`.

Disabling a Calendar User

Use the `csuser` utility's `disable` command to prevent a user from logging into the Calendar Server. The `disable` command prohibits a user from accessing calendar data, but does not remove the user's information from the calendar database.

For example, to disable JSmith from access to Calendar Server data, enter the following command:

```
csuser disable JSmith
```

Note If a user is currently logged on to the Calendar Server, the `disable` command has no affect until the user logs off. The `cssession` utility lets you logout a calendar user (see "Logging Out a User" on page 70).

Checking if a User is Enabled

Use the `csuser` utility's `check` command to determine if a specified user is enabled to access calendar data. For example, to check if JSmith is enabled for calendaring using the default configuration settings, enter the following command:

```
csuser check JSmith
```

Displaying User Information

Use the `csuser` utility's `list` command to list all calendar-enabled users or to view the calendar attributes of a specified user.

For example:

To list all users enabled for calendaring, enter the following command:

```
csuser list
```

To view all of the calendar attributes of a single user, such as JSmith, enter:

```
csuser -v list JSmith
```

Resetting User Calendar Attributes

Use the `csuser` utility's `reset` command to restore the default settings for all calendar attributes of a specified user. For example, to reset all calendar attributes of JSmith to the default configuration settings, enter the following command:

```
csuser reset JSmith
```

Managing Calendars

This section provides examples that show how to perform administrative tasks on calendar data using the `cscal` utility's commands. You must run the `cscal` utility on the local machine where the Calendar Server is installed.

The `cscal` utility lets you perform the following tasks:

- create or delete a calendar
- disable or enable a calendar
- modify properties of an existing calendar

- display calendar information
- reset the attributes of a calendar to the default settings

Creating a Calendar

Every calendar has a primary owner and, optionally, other owners. A primary owner can update the events and tasks of a calendar as well as change the properties of a calendar, such as the name, description, additional owners, and display settings. Other owners can update a calendar's events and tasks, but cannot change a calendar's properties.

Calendars can also be private or public. Public calendars can be viewed by any calendar-enabled user, but only the owners can make changes to the calendar. Private calendars can be viewed only by the calendar's owners.

Use the `cscal` utility's `create` command to create a calendar. For example, to create a calendar owned by JSmith that can be read by anyone, enter the following command on the command line:

```
cscal -n exampleA -o JSmith -r public create calA
```

where:

- `-n exampleA` specifies that `exampleA` is the viewable name of the calendar.
- `-o JSmith` specifies the USERID of the primary owner.
- `-r public` specifies that the calendar can be read by anyone. Write access is still private (the default setting), which means that only the owners (primary owner and other owners, if any) can make updates to the calendar.
- `-calA` is the suffix that will be used as part of the calendar's CALID (for example, `JSmith.calA`).

The following command creates a calendar similar to the example above, but also associates it with the category *sports* and specifies RJones as an other owner:

```
cscal -n exampleA -o JSmith -g sports -y RJones -r public create calA
```

where:

- `-y RJones` specifies that this user an other owner of the calendar.

- `-g sports` specifies that the calendar is associated with a category called *sports*.

Deleting Calendars

Use the `cscal` utility's `delete` command to delete one or more calendars from the Calendar Server.

Note The `delete` command removes all of the calendar's information from the calendar database and cannot be undone. (Database information, however, can be recovered from a database that has been saved to a backup. See “Backing-Up the Calendar Database” on page 64.)

The `cscal` utility lets you delete a single calendar or multiple calendars. For example, to delete a specific calendar with the CALID calendarA, enter the following command:

```
cscal delete calendarA
```

The following example deletes all calendars whose primary owner is JSmith:

```
cscalutil -o JSmith delete
```

Enabling a Calendar

Use the `cscal` utility's `enable` command to allow users to access a calendar. For example, to enable calendarA using the default configuration settings, enter the following command:

```
cscal enable calendarA
```

The following command enables all calendars owned by JSmith:

```
cscal -g John -h airius.com -p 80 enable JSmith
```

Disabling a Calendar User

Use the `cscal` utility's `disable` command to prevent users from accessing one or more calendars. The `disable` command prohibits users from accessing the calendar's data, but does not remove its information from the calendar database.

For example, to prevent users from accessing calendarA on the Calendar Server, enter the following command:

```
cscal disable calendarA
```

To prevent users from accessing all calendars whose primary owner is JSmith, enter the following command:

```
cscal -o JSmith disable
```

Modifying Calendar Properties

Use the `cscal` utility's `modify` command to change the properties of a calendar. For example, to specify that RJones is an other owner of `calendarA` and make it readable only by the owners, enter the following command:

```
cscal -y RJones-r private modify calendarA
```

where:

- `-y RJones` specifies this USERID is an additional owner.
- `-r private` specifies that the calendar is not publicly viewable (can be seen by the owners only).

Removing Properties from a Calendar

To remove a property value, specify two double quotes (" ") as the value of the property option using the `modify` command.

For example, to remove a description from `calendarA`:

```
cscal -d "" modify calendarA
```

To remove all categories from `calendarA`:

```
cscal -g "" modify calendarA
```

To remove other owners from `calendarA`:

```
cscal -y "" modify calendarA
```

Displaying Calendars

Use the `cscal` utility's `list` command to list all calendars, all calendars owned by a user, or all properties of a specific calendar. For example, to list all calendars in the calendar database, enter the following command:

```
cscal list
```

To list all calendars owned by JSmith, enter:

```
cscal -o JSmith list
```

To list all the properties of a calendar with the CALID calendarA, enter:

```
cscal -v list calendarA
```

Resetting Calendar Properties

Use the `cscal` utility's `reset` command to restore the default settings for all properties of a specified calendar or all calendars owned by a specified user. For example, to reset all the properties of calendarA to the default configuration settings, enter the following command:

```
cscal reset calendarA
```

To reset the attributes of all calendars whose primary owner is JSmith to the default configuration settings, enter the following command:

```
cscal -o JSmith reset
```

Managing Calendar Events and Tasks

The `cscomponent` utility lets you:

- list all events and tasks in a calendar.
- delete events and tasks in a calendar.

You must run the `cscomponent` utility on the local machine where your Calendar Server is installed.

Use the `cscomponent` utility's `list` command to list all events and tasks in a calendar. For example, to list the events and tasks in calendarA, enter the following command:

```
cscomponents -v list calendarA
```

where:

- `-v` specifies verbose mode which displays all information available about the events or tasks.

Use the `cscomponent` utility's `delete` command to remove events and tasks in a calendar. For example, to delete all events and tasks in calendarA occurring in 1998, enter the following command:

```
cscomponents -s 19980101T000000Z -e 19981231T000000Z delete calendarA
```

where:

- `-s 19980101T000000Z` specifies the starting date of the time period.
- `-e 19981231T000000Z` specifies the ending date of the time period.

To delete all calendar events and tasks from `calendarA`, enter:

```
cscomponents -s 0 -e 0 delete calendarA
```

where specifying 0 for both start and end dates indicates the beginning and end of time.

Managing Calendar Databases

Use the `csdb` command line utility to view the statistics of the local calendar databases, and to create, delete, backup and restore the calendar database. You must run the `csdb` utility on the local machine where your Calendar Server is installed. This section describes how to use `csdb` to view database statistics and to create and delete databases. For `csdb` backup and restore options, see “Backup and Restore Procedures” on page 62.

Warning You must stop the Calendar Server before restoring, creating, or deleting the calendar database.

For example:

To view calendar database status and statistical information, enter the following command:

```
csdb list
```

To create a new calendar database if it was accidentally deleted, enter:

```
csdb create
```

To delete the calendar database before restoring it from a backup, enter:

```
csdb delete
```

To attempt to recover a damaged calendar database in the current directory, enter the following command:

```
csdb recover
```

Importing and Exporting Calendar Data

The `csexport` and `csimport` command line utilities let you export and import calendar data to a text file using ICAL or XML format.

Note The `csexport` and `csimport` utilities must be run locally on the machine where your Calendar Server is installed and can be executed while the server is either running or stopped.

Exporting Calendar Data

Use `csexport` to export a calendar to a file. The file name extension (`.ics` or `.xml`) that you specify for the output file determines which format is used. For example:

To export the calendar with the CALID JSmithcal in ICAL (text/calendar MIME) format to a file named `jsmith.ics`, enter the following command:

```
csexport -c JSmith calendar jsmith.ics
```

To export the calendar JSmithcal in XML (text/xml MIME) format to a file named `jsmith.xml`, enter:

```
csexport -c JSmithcal calendar jsmith.xml
```

Importing Calendar Data

Use `csimport` to import calendar data from a file previously saved using the `csexport` utility. The file name extension of the import file (`.ics` or `.xml`) indicates the format in which it was saved. For example:

To import calendar data to the CALID JSmithcal from the file `jsmith.ics` that was saved in ICAL (text/calendar MIME) format, enter the following command:

```
csimport -c JSmith calendar jsmith.ics
```

To import data into the calendar JSmithcal from a file named `jsmith.xml` that was saved in XML (text/xml MIME) format, enter:

```
csimport -c JSmithcal calendar jsmith.xml
```

Note If the specified CALID already exists, its data is cleared before the new data is imported.

Backup and Restore Procedures

This section describes how to use the command line utilities to backup and restore Calendar Server data:

- Use `csbackup` to back up a specified calendar, a user's default calendar, or the calendar database.
- Use `csrestore` to recover a specified calendar, a user's default calendar, or the calendar database.

Note The `csexport` and `csrestore` utilities must be run locally on the machine where Calendar Server is installed and can be executed while the server is running or stopped.

This section includes the following topics:

- Backing-Up a Specified Calendar
- Restoring a Specified Calendar
- Backing-Up a User's Default Calendar
- Restoring a User's Default Calendar
- Backing-Up the Calendar Database
- Restoring the Calendar Database
- Using Solstice or Legato Software to Backup Calendar Server

Backing-Up a Specified Calendar

Use the `csbackup` utility's `calendar` command to backup a calendar to a plain text file using ICAL or XML format. The file name extension (`.ics` or `.xml`) that you specify for the output file determines which format is used. For example:

To backup the calendar JSmithcal in ICAL (text/calendar MIME) format, enter the following command:

```
csbackup -c JSmithcal calendar jsmith.ics
```

To backup the calendar JSmithcal in XML (text/xml MIME) format, enter:

```
csbackup -c JSmithcal calendar jsmith.xml
```

Restoring a Specified Calendar

You can restore a specified calendar the using the `csrestore` utility's:

- `calendar` command
- `database` command

Restoring a Calendar from a File

Use the `csrestore` utility's `calendar` command to restore a calendar that was saved to a backup file using the `csbackup` utility. The file name extension (`.ics` or `.xml`) of the backup file indicates the format in which it was saved. For example:

To restore the calendar `JSmithcal` that was saved in ICAL (text/calendar MIME) format to the file `jsmith.ics` located in the `backupdir` directory, enter the following command:

```
csrestore -c JSmithcal calendar backupdir/jsmith.ics
```

To restore the calendar `JSmithcal` that was saved in XML (text/calendar MIME) format to the file `jsmith.xml` located in the `backupdir` directory, enter:

```
csrestore -c JSmithcal calendar backupdir/jsmith.xml
```

Restoring a Calendar from a Database

Use the `csrestore` utility's `database` command to restore a calendar from a database that was saved to a backup directory using the `csbackup` utility. For example, to restore the calendar `JSmithcal` from the backup database directory `backupdir`, enter the following command:

```
csrestore -c JSmithcal database backupdir
```

Backing-Up a User's Default Calendar

Use the `csbackup` utility's `defcal` command to backup a user's default calendar to a plain text file using ICAL or XML format. The file name extension (`.ics` or `.xml`) that you specify for the output file determines which format is used. For example:

To backup calendar user JSmith's default calendar in ICAL (text/calendar MIME) format to a file named `jsmith.ics`, enter the following command:

```
csbackup -a JSmithcal defcal jsmith.ics
```

To backup calendar user JSmith's default calendar in XML (text/xml MIME) format to a file named `jsmith.xml`, enter the following command:

```
csbackup -a JSmithcal defcal jsmith.xml
```

Restoring a User's Default Calendar

Use the `csrestore` utility's `defcal` command to restore a a user's default calendar that was saved to a backup file using the `csbackup` utility. The file name extension (`.ics` or `.xml`) of the backup file indicates the format in which it was saved. For example:

To restore calendar user JSmith's default calendar that was saved in ICAL (text/calendar MIME) format to a file named `jsmith.ics` located in the backup directory `backupdir`, enter the following command:

```
csrestore -a JSmithcal defcal backupdir/jsmith.ics
```

To restore calendar user JSmith's default calendar that was saved in XML (text/xml MIME) format to a file named `jsmith.xml` located in the backup directory `backupdir`, enter the following command:

```
csrestore -a JSmithcal defcal backupdir/jsmith.xml
```

Backing-Up the Calendar Database

Use the `csbackup` utility's `database` command to backup the calendar database to an output directory. For example, to backup the calendar database to a directory named `backupdir`, enter the following command:

```
csbackup database backupdir
```

Restoring the Calendar Database

The following command line utilities let you restore the calendar database:

- `csrestore` lets you restore a calendar database that was previously saved to a backup directory.
- `csdb` lets you recover a calendar database that was accidentally corrupted or deleted.

The `csrestore` and `csdb` commands must be run on the local machine on which your Calendar Server is installed.

Warning You must stop the Calendar Server before restoring or recovering the calendar database. (The server may be running, however, when you backup the database.)

Use the `csrestore` utility's `database` command to restore a calendar database that was saved to a backup directory using the `csbackup` utility. For example, to restore the calendar database that was saved to a backup directory named `backupdir`, enter the following command:

```
csrestore database backupdir
```

Use the `csdb` utility's `recover` command to recover a calendar database that was accidentally corrupted or deleted. For example, to attempt to recover a damaged calendar database in the current directory, enter the following command:

```
csdb recover
```

Using Solstice or Legato Software to Backup Calendar Server

We recommend using the Sun Solaris Solstice Backup or the Legato file backup and restore products to backup the Calendar Server. If you do not have Solstice Backup, use the command line utilities `csbackup`, `csdb`, or `csexport`. See <http://docs.sun.com> for documentation on Solstice Backup.

Note Solstice Backup and the Legato Networker product are identical. The instructions here are applicable to both products. Read the Solstice Backup or Legato Networker documentation set before attempting to backup the Calendar Server.

Calendar Server provides backup and restore procedures that let you:

- backup and restore the entire database or specified calendars

- recover a calendar database that was damaged or lost such as through hardware failure.
- migrate data from one Calendar Server to another.

Backing Up Calendar Server Data

Use the following steps to backup calendar data:

1. Run the command line utilities to backup the calendar database or specified calendars.

For more information, see “Backup and Restore Procedures” on page 62).

The backup procedure creates a backup directory under the current directory. This directory is not the actual directory calendar data is stored, but a directory image of how calendars are stored. This files in this directory are empty and are used only to provide information to the backup program on how calendars will be stored on the backup media. If the backup directory already exists, it is synchronized with the directory structure of the current hierarchy.

2. Start Solstice or Legato backup.

You can use the backup program’s graphical user interface or the `save` command to back up calendar data.

Note: Do not use the Solstice Backup incremental backup feature because the backup directory is only an image of the folder structure and contains no actual data. The incremental backup feature is not supported in iPlanet Calendar Server.

Important: The `.nsr` files generated by the command line utilities contain standard Networker directives and should never be modified.

3. Automate the backup procedure.

The preceding steps describe how to run a backup manually. It is recommended that you set up the backup program’s `backup` command to run the Calendar Server `csbackup` command line utility before the running backup program’s `save` command to achieve an automated backup process.

Note: You cannot use Networker to backup a calendar with a name that contains non-ASCII characters or the forward slash (/).

Restoring Calendar Server Data

Use the Solstice *nwrestore* feature or the `recover` command to restore backed up calendar information. If you use `nwrestore`, you will receive the message:

"File already exists. Do you want to overwrite, skip, backup, or rename?"

Choose `overwrite`. (This message appears because the backup tree is just the directory hierarchy, that is, it consists of empty files and stays that way permanently.)

Note If you use the Solstice `recover` command, you can use the `-A` and `-iy` arguments to suppress this message.

Monitoring the Calendar Server

This chapter describes how to monitor Calendar Server activity. It provides examples that show how to obtain server statistics using the most common options for the `cssession`, `csstats`, and `cstool` command line utilities. For complete details on all the available options, see Appendix A, “Command Line Utilities.”

Managing User Sessions

Use the `cssession` utility’s `list` command to show the status (enabled, disabled, logged-in, logged-out) of all Calendar Server users or a specified user.

Note You can run `cssession` on the local machine where your Calendar Server is installed or from a remote machine.

For example, to show the status of all users, enter the following command on the command line:

```
cssession list
```

To show the status of a specified user such as JSmith, enter the following command:

```
cssession list JSmith
```

Logging Out a User

Use the `cssession` utility's `logout` command to log out a specified user. For example, to log out JSmith who is logged in to the Calendar Server, enter the following command:

```
cssession logout JSmith
```

Listing Counter Statistics

The `csstats` utility displays statistical information obtained from counter objects which are defined in the calendar configuration (*.conf) files. Counter objects (such as `httpstat`, `authstat`, `wcapstat`, or `dbstat`) show information about the Calendar Server such as the:

- maximum number of concurrent connections and total number of connections
- total number of successful and failed logins and connections
- number of database reads, writes and deletes

For detailed information on how to interpret Calendar Server counter statistics see, “Counters” on page 46.

To view statistical information, use the `csstats` utility's `list` command. For example:

To display basic information about the counter objects and the types available, enter the following command:

```
csstats list
```

To list statistics specifically about the `httpstat` counter object, enter:

```
csstats list http
```

To list statistics continuously every 10 seconds about the `wcapstat` counter object, enter:

```
csstats -i -1 -s 10 list wcap
```

Pinging the Calendar Server

Use `cstool` utility's `ping` command to verify that Calendar Server is listening on a specified port. (Pinging the server does not verify that it is running, but indicates if it can accept a socket connection.) For example, to ping `calserver` to see if the Calendar Server is listening on port 80, enter the following command:

```
cstool -p 80 ping calserver
```

where:

- `-p 80` is the port number the Calendar Server listens to by default.
- `calserver` is the host name of the machine on which Calendar Server is installed.

Note The Calendar Server must be running in order to run `cstool`.

Refreshing the Calendar Server Configuration

Use `cstool` utility's `refresh` command to force the Calendar Server to refresh its configuration. For example, to command the Calendar Server installed on `calserver` (the machine's host name) to refresh its configuration, enter the following command:

```
cstool refresh calserver
```

Note The Calendar Server must be running in order to run `cstool`.

Pinging the Calendar Server

A

Command Line Utilities

This appendix provides complete details on the available commands, options, values, syntax notation, and usage rules for the Calendar Server command line utilities. It provides supporting reference material for the topics described in Chapter 3, “Administering Calendar Server.”

Administration for iPlanet Calendar Server is provided by a set of command line utilities. Each utility accepts options from the command line and can be invoked from batch, shell, and scripting programs such as Perl. Where applicable, default values for the available option are taken from the `server.conf` configuration file.

Command Line Utility Syntax

Calendar Server command line utilities use the following syntax:

```
utility [-option [value]] command [target]
```

where:

- `utility` is the executable name of the utility, such as `cscal`.

- `option` determines which action the command performs. Options are in lowercase and preceded by a hyphen (-), such as `-d`. An option enclosed in brackets ([]) indicates it is available, but is not required. When available, a combination of two or more options can be used at the time.
- `value` further qualifies the action specified by an option, such as a description used with the `-d` option. A value enclosed in brackets ([]) indicates it is available, but is not required. Values that include spaces must be enclosed in quotation marks (" "). Multiple values must be enclosed in quotation marks (" ") and each separated by a space.
- `command` is an action the utility can perform, such as `list`. Commands separated by a vertical bar (|) indicate that either one (but not both) can be used at the same time.
- `target` is the object on which the command takes effect, such as `CALID` (calendar ID) or `USERID`.

Usage Rules

The following rules provide general usage guidelines for the command line utilities:

- If no command is specified, the utility lists all options and commands along with examples.
- When required passwords are not specified, the utility will prompt for it.
- The `-v` (verbose) and `-q` (quiet) options are available for every utility.
- When a command is dangerous (one that can result in data loss), the utility will prompt for confirmation before executing the command. An example of a dangerous command is deleting a calendar or user. The `-q` (quiet) option, however, disables confirmation prompting.
- The `version` command is available for every command line utility.

Command Line Utility Summary

Table A.1 summarizes the command line utilities.

Table A.1 Command Line Utilities Summary

Utility	Run from...	Use to..
<code>csbackup</code>	local machine	backup individual calendars, users, and the calendar database.
<code>cscale</code>	local machine	manage calendars and their properties.
<code>cscomponents</code>	local machine	manage components in a calendar (components are events and tasks).
<code>csdb</code>	local machine	manage the calendar database.
<code>csexport</code>	local machine	export a calendar in ICAL or XML format.
<code>csimport</code>	local machine	import a calendar in ICAL or XML format.
<code>csrestore</code>	local machine	restore individual calendars, users, and the calendar database.
<code>cssession</code>	local or remote machine	manage sessions in a Calendar Server.
<code>csstart</code>	local machine	start the Calendar Server (on Windows NT, the server can also be started using the NT services dialog accessed from the Control Panel.).
<code>csstats</code>	local machine	display counters in a Calendar Server.
<code>csstop</code>	local or remote machine	stop the Calendar Server (on Windows NT, the server can also be stopped using the NT services dialog accessed from the Control Panel).
<code>csstool</code>	local or remote machine	ping a Calendar Server or command the server to refresh its configuration.
<code>csuser</code>	local or remote machine	manage calendar users.

csbackup

The `csbackup` utility lets you backup individual calendars, users, and the calendar database. It provides the following commands:

- **database** to backup the calendar database.
- **calendar** to backup a specified calendar.
- **defcal** to backup a user's default calendar.
- **version** displays the version number of the utility currently installed.

Syntax

```
csbackup [-v | -q] database outputdir
```

```
csbackup [-v | -q] -c CALID calendar outputfile
```

```
csbackup [-v | -q] -a USERID [-b BASEDN] [-h host] [-p port] [-u adminuser] [-w password] defcal outputfile
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server can be running or stopped.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csbackup database backupdir
```

The above example backs up the calendar database to a directory named `backupdir`.

```
csbackup -c calA calendar calA.ics
```

The above example backs up the calendar with the `CALID calA` to the file `calA.ics` as `text/calendar`.

```
csbackup -a userA defcal userA.xml
```

The above example backs up the default calendar for user `userA` to the file `userA.xml` as `text/xml`.

Table A.2 describes the commands available for the `csbackup` utility.

Table A.2 `csbackup` Utility Commands

Command	Description
<code>database <i>outputdir</i></code>	Backup the calendar database to the specified output directory.
<code>calendar <i>CALID</i> <i>outputfile</i></code>	Backup the specified <code>CALID</code> to the specified output file. The data format of the file is assumed by the file extension, <code>.ics</code> for <code>text/calendar</code> and <code>.xml</code> for <code>text/xml</code> .
<code>defcal <i>USERID</i> <i>outputfile</i></code>	Backup the default calendar of the specified <code>USERID</code> to the specified target file. The data format of the file is assumed by the file extension, <code>.ics</code> for <code>text/calendar</code> and <code>.xml</code> for <code>text/xml</code> .
<code>version</code>	Displays the version of the utility.

Table A.3 describes the `csbackup` utility command options.

Table A.3 `csbackup` Utility Command Options

Option	Specifies
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-a USERID</code>	The USERID of the calendar user to backup. This option is required for the <code>defcal</code> command. There is no default.
<code>-b BASEDN</code>	BASEDN used for all LDAP users. The default value is taken from the configuration setting <code>local.ugldapbasedn</code> defined in the <code>server.conf</code> file.
<code>-c CALID</code>	CALID to backup. This option is required with the <code>calendar</code> command. There is no default.
<code>-h host</code>	The host name of the machine where the LDAP server that stores user information for this Calendar Server is running. The default value is taken from the configuration setting <code>local.ugldaphost</code> defined in the <code>server.conf</code> file.
<code>-p port</code>	The port number that the LDAP server is listening to. The default value is taken from the configuration setting <code>local.ugldapport</code> defined in the <code>server.conf</code> file.
<code>-u adminDN</code>	The LDAP distinguished name (DN) of the person with administration rights to the LDAP server that stores user authentication information accesses by this Calendar Server. The default value is taken from the configuration setting <code>local.enduseradminDN</code> defined in the <code>server.conf</code> file.
<code>-w adminpassword</code>	Password of the LDAP administrator DN specified by the <code>-u</code> option. The default value is taken from the configuration setting <code>local.enduseradmincred</code> defined in the <code>server.conf</code> file. This option is required if you use the <code>-u</code> option.

cscal

The cscal utility lets you manage calendars and their properties. It provides the following commands:

- **create** a calendar.
- **delete** a calendar.
- **disable** a calendar.
- **enable** a calendar.
- **list** calendars.
- **modify** calendar properties.
- **reset** calendar properties.
- **version** displays the version number of the utility currently installed.

Syntax

```
cscal [-v | -q] [-c charset] [-d description] [-g categories] [-l langcode]
      [-m email] [-n name] [-o owner] [-r public | private]
      [-w public | private] [-y otherowners] create | modify CALID

cscal [-v | -q] [-o owner] delete | disable | enable | list | reset [CALID]
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server can be running or stopped.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
cscal -n exampleA -o userA -r public create calA
```

The above example creates the calendar with the CALID `calA` for `userA` with the visible name `exampleA` which can be read by anyone.

```
cscal delete calA
```

The above example deletes `calA` (removes it from the calendar database).

```
cscal -o userA delete
```

The above example deletes all calendars whose primary owner is `userA` (removes them from the calendar database).

```
cscal disable calA
```

The above example disables the calendar with the CALID `calA` (users will not be allowed to read, write to, or locate it using the user interface).

```
cscal enable calA
```

The above example enables the calendar with the CALID `calA` (users are allowed to read or write to it using the user interface).

```
cscal list calA
```

The above example lists the properties of `calA`.

```
cscal -v list calA
```

The above example lists all the properties of `calA`.

```
cscal list
```

The above example lists all the calendars in the database.

```
cscal -g sports -y userB -r private -v modify calA
```

The above example modifies `calA` so that it is associated with the category *sports*, is co-owned by `userB`, and can be read only by the owners.

```
cscal reset calA
```

The above example resets the calendar with the CALID `calA` to the default configuration settings.

```
cscal -d "" modify calA
```

The above example removes a description from the calendar with the CALID `calA`.

```
cscal -g "" modify calA
```

The above example removes all categories from the calendar with the CALID `calA`.

```
cscal -y "" modify calA
```

The above example removes other owners from the calendar with the CALID `calA`.

Table A.4 describes the commands available for the `cscal` utility.

Table A.4 cscal Utility Commands

Command	Description
<code>create CALID</code>	Creates the calendar specified by CALID.
<code>delete [CALID]</code>	Deletes the calendar specified by CALID. If the <code>-o owner</code> option is specified, deletes all calendars whose primary owner is the specified owner.
<code>enable [CALID]</code>	Enables the calendar specified as CALID. If the <code>-o owner</code> option is specified, enables all calendars whose primary owner is the specified owner.
<code>disable [CALID]</code>	Disables the calendar specified as CALID. If the <code>-o owner</code> option is specified, disables all calendars whose primary owner is the specified owner.
<code>list [CALID]</code>	List properties of the calendar with the specified CALID. If the <code>-o owner</code> option is specified, lists all calendars whose primary owner is the specified owner.
<code>modify CALID</code>	Modifies the properties of the calendar specified as CALID.
<code>reset [CALID]</code>	Resets the properties of the calendar specified as CALID.
<code>version</code>	Displays the version of the utility.

Table A.5 describes the `cscal` utility command options.

Table A.5 cscal Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-c charset</code>	Character set. The default is no character set.
<code>-d description</code>	Description. The default is no description.

Table A.5 cscal Utility Command Options (Continued)

Option	Specifies...
<code>-g category</code>	Category. Multiple categories must be surrounded by quotation marks (" ") and separated by spaces. The default is no category.
<code>-l langcode</code>	Language code. The default is no language code.
<code>-m email</code>	Email address. The default is no email.
<code>-n name</code>	Name. The default is no name.
<code>-o owner</code>	Primary owner. The default is the CALID.
<code>-r public private</code>	Read access which can be <code>public</code> or <code>private</code> . The default is <code>private</code> .
<code>-w public private</code>	Write access which can be <code>public</code> or <code>private</code> . The default is <code>private</code> .
<code>-y otherowners</code>	Other owners. Multiple owners must be surrounded by quotation marks (" ") and separated by spaces. The default is no other owners.

cscomponents

The `cscomponents` utility lets you manage components (events and tasks) in a calendar. It provides the following commands:

- **delete** events and tasks in a calendar.
- **list** events and tasks in a calendar.
- **version** displays the version number of the utility currently installed.

Syntax

```
cscomponents [-v | -q] [-e endtime] [-s starttime] [-t event | task] delete | list CALID
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server can be running or stopped.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csccomponents -s 19980101T000000Z -e 19981231T000000Z delete calA
```

The above command deletes all 1998 events in the calendar with the CALID calA.

```
csccomponents -v list calA
```

The above command lists all events and tasks, with details, in the calendar with the CALID calA.

Table A.6 describes the commands available for the `csccomponent` utility.

Table A.6 `csccomponent` Utility Commands

Command	Description
<code>delete CALID</code>	Deletes events in the calendar with the specified CALID.
<code>list CALID</code>	Lists events in the calendar with the specified CALID.
<code>version</code>	Prints the version of the utility to the screen.

Table A.7 describes the `csccomponent` utility command options.

Table A.7 `csccomponent` Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> displays no information if the operation is successful (errors, if they occur, will still be displayed). suppresses confirmation prompting for dangerous commands. Default is off.
<code>-e endtime</code>	Ending time of the components. An end time of 0 means to the end of time. The default is 0.
<code>-s starttime</code>	Starting time of the components. A start time of 0 means from the beginning of time. The default is 0.
<code>-t event task</code>	Type of components (events or tasks) on which the action is performed. Default is both.

csdb

The csdb utility lets you manage the calendar databases. It provides the following commands:

- **create** a new database. If a database does not exist when the server is started, Calendar Server creates one automatically.
- **delete** an existing calendar database. A database cannot be deleted while it is open (when the Calendar Server is running).
- **list** information about the database.
- **recover** a damaged calendar database.
- **version** displays the version number of the utility currently installed.

Note For more information on Calendar Server databases, see “Database Technology” on page 14.

Syntax

```
csdb [-v | -q] [-t cal | sess | stats] list
```

```
csdb [-v | -q] [-t cal | sess | stats] [-h host] [-p port] create | delete | recover [dbdir]
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server cannot be running if creating or deleting the database.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csdb -v create
```

The above command creates new, un-populated databases in the current directory.

```
csdb -v delete
```

The above command deletes the databases in the current directory.

```
csdb -v list
```

The above command lists information about the databases in the current directory.

```
csdb recover
```

The above command attempts to recover damaged databases in the current directory.

```
csdb -t sess list
```

The above command lists information about the sessions database in the current directory.

```
csdb -t stat recover
```

The above command attempts to recover a damaged statistics database in the current directory.

Table A.8 describes the commands available for the `csdb` utility.

Table A.8 `csdb` Utility Commands

Command	Description
<code>create [dbdir]</code>	Creates databases in the specified database directory. If a database directory is not specified, the current directory is used.
<code>delete [dbdir]</code>	Deletes databases in the specified database directory. If a database directory is not specified, the current directory is used.
<code>list [dbdir]</code>	Lists information about the databases in the specified database directory. If a database directory is not specified, the current directory is used.
<code>recover [dbdir]</code>	Attempts to recover damaged databases in the specified database directory. If a database directory is not specified, the current directory is used.
<code>version</code>	Displays the version of the utility.

Table A.9 describes the `csdb` utility command options.

Table A.9 csdb Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-t cal stat sess</code>	Which database to perform an action on: calendar (<code>cal</code>), statistics (<code>stat</code>), or session (<code>sess</code>). If <code>-t</code> is not specified or if <code>-t</code> is specified without a database value, all databases are assumed.
<code>-h host</code>	The host name of the machine where the LDAP server that stores user information for this Calendar Server is running. The default is taken from the setting <code>local.ugldaphost</code> defined in the <code>server.conf</code> file.
<code>-p port</code>	The port number that the LDAP server is listening to. The default is taken from the setting <code>local.ugldapport</code> defined in the <code>server.conf</code> file.

csexport

The `csexport` utility lets you export a calendar to a file in ICAL or XML format. It provides the following commands:

- **calendar** export a specified calendar.
- **version** displays the version number of the utility currently installed.

`calendar` specifies the calendar on which to perform an action

Syntax

```
csexport [-v | -q] -c CALID calendar outputfile
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server can be running or stopped.

- On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csexport -c calA calendar calA.ics
```

The above command exports the calendar with the CALID `calA` in ICAL (text/calendar) format to file named `calA.ics`.

```
csexport -c calA calendar calA.xml
```

The above command exports the calendar with the CALID `calA` in XML (text/xml) format to file named `calA.xml`.

Table A.10 describes the commands available for the `csexport` utility.

Table A.10 `csexport` Utility Commands

Command	Description
<code>calendar outputfile</code>	Exports the calendar to the specified output file. The data format of the file is determined by the specified file name extension: .ics for ICAL (text/calendar) .xml for XML (text/xml)
<code>version</code>	Displays the version of the utility.

Table A.11 describes the `csexport` utility command options.

Table A.11 `csexport` Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.

Table A.11 csexport Utility Command Options (Continued)

Option	Specifies...
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-c CALID</code>	The CALID of the calendar to export. This option is required with the <code>calendar</code> command. There is no default.

csimport

The `csimport` utility lets you import a calendar from a file in ICAL or XML format that was saved with the `csexport` utility. It provides the following commands:

- **calendar** imports a specified calendar.
- **version** displays the version number of the utility currently installed.

Syntax

```
csimport [-v | -q] -c CALID calendar inputoutputfile
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server can be running or stopped.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csimport -c calA calendar calA.ics
```

The above command imports the calendar with the CALID `calA` from the file `calA.ics` and expects ICAL (text/calendar file) format.

```
csexport -c calA calendar calA.xml
```

The above command imports the calendar with the CALID `calA` from the file `calA.xml` and expects XML (text/calendar file) format.

Table A.12 describes the commands available for the `csimport` utility.

Table A.12 `csimport` Utility Commands

Command	Description
<code>calendar inputfile</code>	Imports the calendar from the specified input file. The data format of the file is determined by the file name extension: <ul style="list-style-type: none"> <code>.ics</code> for ICAL (text/calendar) <code>.xml</code> for XML (text/xml)
<code>version</code>	Displays the version of the utility.

Table A.13 describes the `csimport` utility command options.

Table A.13 `csimport` Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-c CALID</code>	The CALID to of the calendar to import. This option is required with the <code>calendar</code> command. Note: If the specified CALID already exists, the imported data is merged with the current calendar. There is no default.

csrestore

The `csrestore` utility lets you restore individual calendars, users, and the calendar database. It provides the following commands:

- **database** restores the calendar database.
- **calendar** restores a calendar.
- **defcal** restores a user's default calendar.
- **version** displays the version number of the utility currently installed.

Syntax

```
csrestore [-v | -q] database inputdir
csrestore [-v | -q] -c CALID calendar inputfile
csrestore [-v | -q] -a USERID [-b BASEDN] [-h host] [-p port] [-u adminuser] [-w password] defcal inputfile
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server cannot be running if restoring the calendar database.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csrestore database backupdir
```

The above command restores the calendar database stored in the directory `backupdir` that was previously saved using the `csbackup` command line utility.

```
csrestore -c calA calendar backupdir/calA.ics
```

The above command restores the calendar with the `CALID calA` from the file `calA.ics` located in the directory `backupdir` that was previously saved in ICAL (text/calendar file) format using the `csbackup` or `csexport` utility.

```
csrestore -c calA calendar backupdir
```

The above command restores `calA` from the calendar database in `backupdir` that was previously saved using `csbackup`.

```
csrestore -a userA defcal backupdir/userA.ics
```

The above command restores the default calendar owned by `userA` from the file `userA.ics` located in the directory `backupdir` that was previously saved in ICAL (text/calendar file) format using the `csbackup` or `csexport` utility.

Table A.14 describes the commands available for the `csrestore` utility.

Table A.14 csrestore Utility Commands

Command	Description
<code>database inputdir</code>	Restore the calendar database from the specified input directory or input file that contains a backup calendar database. This will destroy and overwrite all previous contents of the current calendar database.
<code>calendar inputfile</code>	Restore the specified CALID from the specified input file. The data format of the file is determined by the file name extension: .ics for ICAL (text/calendar). .xml for XML (text/xml). If the specified CALID already exists, the calendar's data is cleared before it is restored.
<code>defcal inputfile</code>	Restores the default calendar of the specified USERID from the input file specified. The data format of the file is determined by the file name extension: .ics for ICAL (text/calendar). .xml for XML (text/xml).
<code>Version</code>	Displays the version of the utility.

Table A.15 describes the `csrestore` utility command options.

Table A.15 csrestore Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-a USERID</code>	The USERID to restore. This option is required with the <code>defcal</code> command. There is no default.

Table A.15 csrestore Utility Command Options (Continued)

Option	Specifies...
<code>-b BASEDN</code>	The BASEDN to be used for all LDAP users. The default is taken from the setting <code>local.ugldapbasedn</code> defined in the <code>server.conf</code> file.
<code>-c CALID</code>	The CALID to restore. This option is required with the <code>calendar</code> command. There is no default.
<code>-h host</code>	The host name of the machine where the LDAP server that stores user information for this Calendar Server is running. The default is taken from the setting <code>local.ugldaphost</code> defined in the <code>server.conf</code> file.
<code>-p port</code>	The port number that the LDAP server is listening to. The default is taken from the setting <code>local.ugldapport</code> defined in the <code>server.conf</code> file.
<code>-u adminDN</code>	The LDAP distinguished name (DN) of the person with administration rights to the LDAP server that stores user authentication information accessed by this Calendar Server. The default value is taken from the configuration setting <code>local.enduseradminDN</code> defined in the <code>server.conf</code> file. For example: <code>uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot.</code>
<code>-w adminpassword</code>	Password of the LDAP administrator DN specified by the <code>-u</code> option. The default value is taken from the configuration setting <code>local.enduseradmincred</code> defined in the <code>server.conf</code> file. This option is required if you use the <code>-u</code> option.

csession

The `csession` command line utility lets you manage sessions in a Calendar Server. It provides the following commands:

- **list** a user's status or all logged in users.
- **logout** a specified user or all users.
- **version** displays the version number of the utility currently installed.

Syntax

```
csession [-v | -q] [-h host] [-a port] [-s sessionid | -t tcpaddress] [-u calmaster] [-w password] list | logout [USERID]
```

- Requirements:**
- The Calendar Server must be running.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
cssession -p 81 -u calmaster -v -w calmasterpwd list userA
```

The above command lists the status for `userA` using the `calmaster` as the USERID of the administrator accessing the Calendar Server admin port at port number 81.

```
cssession -p 81 -u calmaster -w calmasterpwd list
```

The above command lists all logged-in users using the `calmaster` as the USERID of the administrator accessing the Calendar Server admin port at port number 81.

```
cssession -p 81 -u calmaster -w calmasterpwd logout userA
```

The above command will logout `userA` using the `calmaster` as the USERID of the administrator accessing the Calendar Server admin port at port number 81.

Table A.16 describes the commands available for the `cssession` utility.

Table A.16 `cssession` Utility Commands

Command	Description
<code>list [USERID]</code>	Displays the status for the specified USERID. If a USERID is not specified, it lists the status for all logged in users.
<code>logout [USERID]</code>	Logs out the specified USERID. If a USERID is not specified, it logs out all users.
<code>version</code>	Displays the version of the utility.

Table A.17 describes the `cssession` utility command options.

Table A.17 csession Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-h <i>host</i></code>	The host name of the machine on which Calendar Server is installed. The default is taken from the setting <code>local.hostname</code> defined in the <code>server.conf</code> file
<code>-a <i>port</i></code>	The port number that the Calendar Server administration port is listening to. The default is taken from the configuration setting <code>service.http.admin.port</code> defined in the <code>server.conf</code> file.
<code>-s <i>sessionID</i></code>	The user's current session ID. There is no default.
<code>-t <i>tcpaddress</i></code>	The user's current TCP/IP address. There is no default.
<code>-u <i>calmaster</i></code>	The USERID of the person with administration rights to this Calendar Server that was specified during installation. The default value is taken from the configuration setting <code>service.http.calmaster.admin</code> defined in the <code>server.conf</code> file. This option is required.
<code>-w <i>password</i></code>	The password for the <i>calmaster</i> USERID specified with the <code>-u</code> option. The default value is taken from the configuration setting <code>service.http.calmaster.cred</code> defined in the <code>server.conf</code> file. This option is required.

csstart

The `csstart` utility lets you start the Calendar Server. It provides the following command:

- **service** starts all services or a specified service of Calendar Server.
- **version** displays the version number of the utility currently installed.

Syntax

```
csstart [-q | -v] [-a port] [-p port] [-g groupid] [-l listenaddr] [-u USERID] service [servicename]
```

- Requirements:**
- Must be run locally on the machine where Calendar Server is installed.
 - The Calendar Server must be stopped.

Examples

```
csstart service
```

The above command starts all Calendar Server services using the default ports.

```
csstart -p 80 -a 81 service http
```

The above command starts the HTTP service listening on port 80 and the administration port listening on port 81.

Table A.18 describes the commands available for the `csstart` utility.

Table A.18 `csstart` Utility Commands

Command	Description
<code>service [servicename]</code>	Stops a specified Calendar Server service (such as <code>http</code>), or all Calendar Server services if no service name is specified.
<code>version</code>	Display the version of the utility.

Table A.19 describes the `csstart` utility command options.

Table A.19 `csstart` Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.

Table A.19 csstart Utility Command Options (Continued)

Option	Specifies...
<code>-a port</code>	<p>The Calendar Server administration port. If not specified, an administration port will not be opened and:</p> <ul style="list-style-type: none"> the Calendar Server cannot be stopped using the <code>csstop</code> utility commands cannot be sent using the <code>cssession</code> or <code>cstool</code> utilities. <p>The default value is taken from the configuration setting <code>service.http.admin.port</code> defined in the <code>server.conf</code> file.</p>
<code>-p port</code>	Specifies the Calendar Server user port. The default value is taken from the setting <code>service.http.port</code> defined in the <code>server.conf</code> file.
<code>-g groupid</code>	The Unix group that the Calendar Server is running as (such as <code>icsgroup</code>) that was specified during installation. (Unix systems only.)
<code>-l listenaddr</code>	The address that the HTTP service is listening to. The default is: <code>INADDR_ANY</code> .
<code>-u USERID</code>	The Unix user that the Calendar Server is running as (such as <code>icsuser</code>) that was specified during installation. (Unix systems only.)

csstats

The `csstats` utility lets you display the statistics of a Calendar Server and provides the following commands:

- list** displays various statistics that are recorded as counters on the Calendar Server.
- version** displays the version number of the utility currently installed.

Note For more information on counters, see “Counters” on page 46.

Syntax

```
csstats [-q | v] [-r registry] [-i iterations] [-s delay] list [subsystem]
```

Requirements:

- Must be run locally on the machine where Calendar Server is installed.

- The Calendar Server can be running or stopped.
- On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csstats list
```

The above command displays basic information about counters and what types are available.

```
csstats list http
```

The above command lists counter statistics about the HTTP service subsystem (`hpptstat`).

```
csstats -i 3600 -s 10 list wcap
```

The above command lists counter statistics about the wcap subsystem (`wcapstat`) every 10 seconds for one hour (3600 seconds).

Table A.20 describes the commands available for the `csstats` utility.

Table A.20 `csstats` Utility Commands

Command	Description
<code>list [subsystem]</code>	Lists statistics about a Calendar Server counter. If a counter is not specified, it displays basic information about what types are available.
<code>Version</code>	Displays the version of the utility.

Table A.21 describes the `csstats` utility command options.

Table A.21 csstats Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-i iterations</code>	The number of times to repeat statistical lookups. Default is 1.
<code>-r registry</code>	The counter statistics file. Default is: <code>server-root/cal/bin/counter/counter</code>
<code>-s delay</code>	The amount of time (in seconds) to wait between each statistical lookup. The default is 1 second.

csstop

The `csstop` utility lets you stop the Calendar Server. It provides the following commands:

- **service** stops all services or a specified service of Calendar Server.
- **version** displays the version number of the utility currently installed.

Syntax

```
csstop [-v | -q] [-a port] [-r] [-u calmaster] [-w password] service [servicename]
```

- Requirements:**
- Can be run on the local machine where Calendar Server is installed or from a remote machine.
 - The Calendar Server must be running.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csstop service
```

The above command stops all local Calendar Server services.

```
csstop service http
```

The above command stops a local Calendar Server http service.

Table A.22 describes the commands available for the `csstop` utility.

Table A.22 `csstop` Utility Commands

Command	Description
<code>service [servicename]</code>	Stops a specified Calendar Server service (such as <code>http</code>), or all Calendar Server services if no service name is specified.
<code>version</code>	Displays the version of the utility.

Table A.23 describes the `csstop` utility command options.

Table A.23 `csstop` Utility Command Options

Option	Specifies
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> displays no information if the operation is successful (errors, if they occur, will still be displayed). suppresses confirmation prompting for dangerous commands. Default is off.
<code>-a port</code>	The port number that the Calendar Server administration port is listening to. The default value is taken from the setting <code>service.http.admin.port</code> defined in the <code>server.conf</code> file.
<code>-h host</code>	Specifies the host name if the machine where the Calendar Server is installed. The default value is taken from the setting <code>local.hostname</code> defined in the <code>server.conf</code> file.
<code>-r</code>	To stop a remote Calendar Server. The default is to stop locally by sending UNIX signals or NT service interrupts.

Table A.23 csstop Utility Command Options (Continued)

Option	Specifies
<code>-u calmaster</code>	The USERID of the person with administration rights to this Calendar Server that was specified during installation. The default value is taken from the configuration setting <code>service.http.calmaster.admin</code> defined in the <code>server.conf</code> file.
<code>-w password</code>	The password for the <code>calmaster</code> USERID specified with the <code>-u</code> option. The default value is taken from the configuration setting <code>service.http.calmaster.cred</code> defined in the <code>server.conf</code> file. This option is required if the <code>-u</code> option is specified.

cstool

The `cstool` utility lets you ping a Calendar Server or force the Calendar Server to refresh its configuration. It provides the following commands:

- **ping** a Calendar Server to ensure it is listening on a port.
- **refresh** forces the server to refresh its configuration settings.
- **version** displays the version number of the utility currently installed.

Syntax

```
cstool [-q | -v] [-p port] [-t timeout] ping [hostname]
```

```
cstool [-q | -v] [-p port] [-r] [-t timeout] [-u calmaster] [-w password] refresh [hostname]
```

- Requirements:**
- The Calendar Server must be running.
 - On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
cstool -p 80 ping calserver
```

The above command pings the machine with the host name `calserver` to see if a Calendar Server is listening on port 80.

```
cstool refresh
```

The above command forces a local Calendar Server to refresh its configuration.

Table A.24 describes the commands available for the `cstool` utility.

Table A.24 `cstool` Utility Commands

Command	Description
<code>ping [hostname]</code>	Ping a Calendar Server running on a remote machine with the specified host name, or, if no host name is specified, ping the local Calendar Server.
<code>refresh [hostname]</code>	Force a Calendar Server running on a remote machine with the specified host name to refresh its configuration, or, if no host name is specified, force the local Calendar Server to refresh its configuration.
<code>version</code>	Display the version of the utility.

Table A.25 describes the `cstool` utility command options.

Table A.25 `cstool` Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.
<code>-p port</code>	The Calendar Server port. The default value is taken from the setting <code>service.http.port</code> defined in the <code>server.conf</code> file.
<code>-r</code>	To command a remote Calendar Server. The default is to send the command locally by sending UNIX signals or NT service interrupts.
<code>-t timeout</code>	The amount of time (in seconds) to wait for a response from the server. The default is 120 seconds.

Table A.25 `csuser` Utility Command Options (Continued)

Option	Specifies...
<code>-u calmaster</code>	The USERID of the person with administration rights to this Calendar Server that was specified during installation. The default value is taken from the configuration setting <code>service.http.calmaster.admin</code> defined in the <code>server.conf</code> file.
<code>-w password</code>	The password for the <code>calmaster</code> USERID specified with the <code>-u</code> option. The default value is taken from the configuration setting <code>service.http.calmaster.cred</code> defined in the <code>server.conf</code> file. This option is required if the <code>-u</code> option is specified.

csuser

The `csuser` utility lets you manage calendar user information stored in an LDAP directory server. It provides the following commands:

- **check** if user is enabled for calendaring.
- **create** and enable a user for calendaring.
- **delete** a user.
- **disable** a user from logging in to the Calendar Server.
- **enable** a user to log on to the Calendar Server.
- **list** a user's calendar attributes.
- **reset** a user's calendar attributes to the default settings.

Syntax

```
csuser [-v | -q] [-b BASEDN] [-c CALID] -g givenname [-h host] [-p port]
      -s surname [-u ldapadmin] [-w password] -y userpassword create USERID
```

```
csuser [-v | -q] [-b BASEDN] [-h host] [-p port] [-u adminuser] [-w password]
      check | delete | disable | enable | list | reset [USERID]
```

Requirements:

- The Calendar Server can be running or stopped.
- The LDAP server that stores calendar user information must be running.
- On Unix systems, must be run while logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csuser check JSmith
```

The above example checks if the calendar user `JSmith` is enabled for calendaring (if the existing calendar user has access to calendar data on this Calendar Server).

```
csuser -g John -s Smith -y password -c JSmithcal create JSmith
```

The above example creates an LDAP user with the USERID `JSmith` with the givenname `John`, surname `Smith`, and the CALID `JSmithcal`.

```
csuser delete JSmith
```

The above example deletes the user `JSmith`.

```
csuser disable JSmith
```

The above example disables the calendar user `JSmith` from logging in to the Calendar Server.

Note The command in this example prevents `JSmith` from being able to log in to the Calendar Server to access calendar data, but `JSmith`'s data is not deleted from the local calendar database. If `JSmith` is currently logged in to the Calendar Server, `JSmith` retains access to calendar data until he logs off or is logged off by the administrator. (The `cssession` utility's `logout` command lets an administrator log off a user. For more information see, "cssession" on page 91.)

```
csuser enable JSmith
```

The above example enables `JSmith` for calendaring (lets existing calendar user `JSmith` log on to the Calendar Server).

```
csuser -v list JSmith
```

The above example lists all calendar attributes for `JSmith`.

```
csuser reset JSmith
```

The above example resets all calendar attributes for `JSmith` to the default configuration settings.

Table A.26 describes the commands available for the `csuser` utility.

Table A.26 csuser Utility Commands

Command	Description
<code>check [USERID]</code>	Check if the specified USERID is enabled for calendaring. If a USERID is not specified, it checks all users.
<code>create USERID</code>	Create the specified USERID and enable this user to log in to the Calendar Server.
<code>delete [USERID]</code>	Delete the specified USERID. If no USERID is specified, all users are deleted.
<code>disable [USERID]</code>	Disable the specified USERID for calendaring. The utility defines <code>http</code> as the value of the <code>nswcalDisallowAccess</code> attribute. If no USERID is specified, all users are disabled.
<code>enable [USERID]</code>	Enable the specified USERID for calendaring. (The utility adds the specified CALID to the <code>nswcalCALID</code> attribute.) If a USERID is not specified, all users are enabled.
<code>list [USERID]</code>	List all the calendar attributes for the specified USERID. If no USERID is specified, it lists all enabled users.
<code>reset [USERID]</code>	Reset all calendar attributes for a USERID to the default settings. If no USERID is specified, it resets the attribute of all users.
<code>version</code>	Display the version of the utility.

Table A.27 describes the `csuser` utility command options.

Table A.27 csuser Utility Command Options

Option	Specifies...
<code>-v</code>	Verbose mode which displays all available information about the command being performed. Default is off.
<code>-q</code>	Quiet mode which: <ul style="list-style-type: none"> • displays no information if the operation is successful (errors, if they occur, will still be displayed). • suppresses confirmation prompting for dangerous commands. Default is off.

Table A.27 csuser Utility Command Options (Continued)

Option	Specifies...
<code>-b BASEDN</code>	The BASEDN to be used for all LDAP users. The default value is taken from the setting <code>local.ugldapbasedn</code> defined in the <code>server.conf</code> file.
<code>-c CALID</code>	The CALID of the default calendar to associate with the specified USERID. The default is the USERID. Note: This command does not create a calendar.
<code>-g givenname</code>	The user's LDAP givenname (first name). This option is required if you use the <code>create</code> command. There is no default.
<code>-h host</code>	The host name of the machine where the LDAP server that stores user information for this Calendar Server is running. The default value is taken from the setting <code>local.ugldaphost</code> defined in the <code>server.conf</code> file.
<code>-p port</code>	The port number that LDAP server is listening to. The default value is taken from the setting <code>local.ugldapport</code> defined in the <code>server.conf</code> file.
<code>-s surname</code>	The user's LDAP surname (last name). This option is required if you use the <code>create</code> command. There is no default.
<code>-u adminDN</code>	The LDAP distinguished name (DN) of the person with administration rights to the LDAP server that stores user authentication information accessed by this Calendar Server. The default value is taken from the configuration setting <code>local.enduseradminDN</code> defined in the <code>server.conf</code> file.
<code>-w password</code>	Password of the LDAP administrator DN specified by the <code>-u</code> option. The default value is taken from the configuration setting <code>local.enduseradmincred</code> defined in the <code>server.conf</code> file. This option is required if you use the <code>-u</code> option.
<code>-y userpassword</code>	The calendar user's password. This option is required if you use the <code>create</code> command. There is no default.

Monitoring Tools

The following table lists system tools you can use to monitor your server environment. These tools are available on various Unix platforms. For more information about these tools, see the man pages delivered with your Unix system.

Table B.1 General Unix tools

Tool	Description
<code>iostat</code>	Provides information about disk I/O and CPU usage.
<code>lsof</code>	Provides information about open file descriptors. (Available in source from <code>:ftp://vic.cc.purdue.edu/pub/tools/unix.</code>)
<code>lslk</code>	Provides information about file system locks. (Available in source from <code>:ftp://vic.cc.purdue.edu/pub/tools/unix.</code>)
<code>netstat</code>	Provides statistics about network functions.
<code>nslookup</code>	Allows you to query DNS servers for information about hosts and domains; for example you can print a list of hosts in a particular domain; also provides an IP address-to-host name mapping function (and vice versa).
<code>ping</code>	Allows you to query the status of a remote host or network gateway.

Table B.1 General Unix tools (Continued)

Tool	Description
<code>sar</code>	Unix SysV performance monitoring tool. Useful for gathering system information over a longer period of time to use in long term planning, for example.
<code>tcpdump</code>	Public domain tools used in debugging and to monitor network traffic.
<code>top</code>	Provides quick, easy monitoring of processes and CPU activities. (This is a public domain tool that works on most Unix platforms.)
<code>trace</code>	Similar to <code>truss</code> on Solaris. Sometimes included by the vendor; otherwise, you can download this tool from an Internet site.
<code>traceroute</code>	Determines the path a packet takes throughout the Internet to reach its final destination.
<code>vmstat</code>	Provides statistics about process, virtual memory, disk, trap, and CPU activity.

Table B.2 System monitoring tools - Sun Solaris

Tool	Description
<code>lockstat</code>	Provides information on OS and application locking. Available on Solaris 2.6 only.
<code>mpstat</code>	Provides statistics about each processor on the system
<code>pmap</code>	Provides breakdown on how much memory a process is using so you can see how much is shared and how much is private. (Located in <code>/usr/proc/bin</code> .)
<code>proctool</code>	Monitors processes and threads. (Available from Sun's web site.)
<code>snoop</code>	Monitors network traffic; indispensable when debugging low-level packets.

Table B.2 System monitoring tools - Sun Solaris

Tool	Description
SymbEL/Virtual Adrian	A very powerful system monitoring toolkit. Provides the functionality of the above listed tools and more. Can be used to tune the <code>ncsize</code> and <code>ufs_ninode</code> parameters and even has a mode to tune the operating system automatically.
<code>truss</code>	Provides information about which system calls a process makes.

Table B.3 System monitoring tools - HP-UX

Tool	Description
<code>glance</code>	Provides detailed system information about open file descriptors, locks, threads, and so on.
<code>gpm</code>	Provides detailed system information about open file descriptors, locks, threads, and so on.
<code>tusc</code>	A system call trapper. Might not be available on all systems.
<code>sysdef</code>	Provides information about kernel parameters.
<code>landiag</code>	A tool for monitoring network statistics.
<code>sam</code>	System Administration Manager. A tool for general system administration.

Table B.4 System monitoring tools - SGI Irix

Tool	Description
<code>dkstat</code>	Similar to <code>iostat</code> . Provides information about disk I/O and CPU use.
<code>gmemusage (Irix 6.x)</code>	X windows tool for viewing information about virtual memory.
<code>netstat -C</code>	Provides real-time, full-screen data.

Table B.4 System monitoring tools - SGI Irix

Tool	Description
<code>osview</code>	Provides full-screen information; combines functionality of <code>vmstat</code> , <code>mpstat</code> , and <code>netstat</code> .
<code>par</code>	Similar to <code>truss</code> on Solaris. Provides information about system calls made by a process.
Performance Copilot	An SGI add-in package.

Time Zones

This appendix describes how Calendar Server processes time zones. It also describes how an administrator can add a new time zone table or modify the time zone tables supplied with Calendar Server.

The file `data/timezones_libnls.ics` shown (see “Calendar Server Time Zones Table” on page 115) contains the representation of the time zones supported by Calendar Server. This file is located in the `server-root/cal/bin/data` directory (for example, `opt/SUNWicsrv/cal/bin/data`). The 91 time zones that this file contains are based off the time zone list defined in `libnls3.0`.

At startup, the Calendar Server reads the file `timezones_libnls`, then generates 91 in-memory time zones and stores them in an array (`cal_array`). Therefore, all the time zones are kept in memory while the server is running.

If a WCAP command includes a time zone ID (`tzid`), it should reference a time zone in the `cal_array`. For more information on WCAP, refer to the *iPlanet Calendar Server Programmer's Guide*.

For example, if a command such as `storeevents` or `fetchcomponents_by_range` specifies a `tzid` parameter, the parameter value must be a `tzid` that is defined in the list of 91 time zones. The server will return data in that time zone and all data will have dates applied to that time zone.

If the command specifies an unrecognized `tzid`, the server will return a GMT time zone by default. All data returned on that command will have its dates applied in GMT.

For example, suppose the following WCAP command is sent to the server:

```
fetchcomponents_by_range.wcap?id=12345&calid=JSmithcal&dtstart=0&dtend=0&tzid=America/New_York&brief=1
```

The above command asks the server to return all data in the CALID JSmithcal with all dates returned in the time zone defined in `America/New_York`. Since this time zone is recognized by the server, the data will be returned successfully as follows:

```
HTTP/1.1 200
Date: Tue, 17 Aug 1999 23:58:29 GMT
Content-type: text/html; charset=iso-8859-1
Content-Length: 4206
Last-modified: Tue, 17 Aug 1999 23:58:29 GMT
Pragma: no-cache
Expires: 0
Cache-Control: no-cache
Connection: Keep-Alive
<html><head><script>
function color(s) { if (s) document.bgColor=s }
var id='n3l0m05sbeee8uh'
var userid='malika'
var calid='pub'
var errno=new Array()
var timezoneList = new Array()
var calprops = new Array()
var layer_errno = new Array()
function iso(p) {
  if (!p) return null;
  var y = p.substring(0,4)
  var m = p.substring(4,6)
  var d = p.substring(6,8)
  var h = p.substring(9,11)
  var i = p.substring(11,13)
  var s = p.substring(13,15)
  return new parent.ceDate(y, m-1, d, h, i, s, 0)
}
function Ec(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13){
this.uid=a1
this.rid=a2
this.calid=a3
this.dtstart=iso(a4)
this.dtend=iso(a5)
this.isAllDay=a6
this.summary=a7
this.created=iso(a8)
this.lastMod=iso(a9)
this.desc=a10
this.location=a11
this.tzid=a12
this.linkCalid=a13
}function Tc(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15){
this.uid=a1
this.rid=a2
this.calid=a3
```

```

this.dtstart=iso(a4)
this.due=iso(a5)
this.isAllDay=a6
this.summary=a7
this.created=iso(a8)
this.lastMod=iso(a9)
this.desc=a10
this.location=a11
this.completed=iso(a12)
this.percent=a13
this.tzid=a14
this.linkCalid=a15
}
function TZ(a1,a2,a3,a4,a5,a6) {
  this.tzid=a1
  this.sName=a2
  this.dName=a3
  this.sOffset=a4
  this.dOffset=a5
  this.crossOver=a6
}
function CP(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15) {
  this.calid=a1
  this.name=a2
  this.parent=a3
  this.tzid=a4
  this.read=a5
  this.write=a6
  this.charset=a7
  this.lang=a8
  this.master=a9
  this.desc=a10
  this.lastMod=iso(a11)
  this.created=iso(a12)
  this.primaryOwner=a13
  this.owners=a14
  this.categories=a15
}
var event=new Array()
var status_types=new Array('confirmed', 'tentative', 'cancelled')
timezonelist[0] = new TZ('America/New_York',
'EST',
'EDT',
'-0500',
'-0400',
new Array('19880403T070000Z','19881030T060000Z','19890402T070000Z','19891029T060000Z',
'19900401T070000Z','19901028T060000Z','19910407T070000Z','19911027T060000Z',
'19920405T070000Z','19921025T060000Z','19930404T070000Z','19931031T060000Z',
'19940403T070000Z','19941030T060000Z','19950402T070000Z','19951029T060000Z',
'19960407T070000Z','19961027T060000Z','19970406T070000Z','19971026T060000Z',
'19980405T070000Z','19981025T060000Z','19990404T070000Z','19991031T060000Z',
'20000402T070000Z','20001029T060000Z','20010401T070000Z','20011028T060000Z',
'20020407T070000Z','20021027T060000Z','20030406T070000Z','20031026T060000Z',
'20040404T070000Z','20041031T060000Z','20050403T070000Z','20051030T060000Z',
'20060402T070000Z','20061029T060000Z','20070401T070000Z','20071028T060000Z',
'20080406T070000Z','20081026T060000Z','20090405T070000Z','20091025T060000Z',
'20100404T070000Z','20101031T060000Z','20110403T070000Z','20111030T060000Z',
'20120401T070000Z','20121028T060000Z','20130407T070000Z','20131027T060000Z',
'20140406T070000Z','20141026T060000Z','20150405T070000Z','20151025T060000Z',
'20160403T070000Z','20161030T060000Z','20170402T070000Z','20171029T060000Z',
'20180401T070000Z','20181028T060000Z','20190407T070000Z','20191027T060000Z',
'20200405T070000Z','20201025T060000Z','20210404T070000Z','20211031T060000Z',
'20220403T070000Z','20221030T060000Z','20230402T070000Z','20231029T060000Z',

```

```

'20240407T070000Z','20241027T060000Z','20250406T070000Z','20251026T060000Z',
'20260405T070000Z','20261025T060000Z','20270404T070000Z','20271031T060000Z',
'20280402T070000Z','20281029T060000Z','20290401T070000Z','20291028T060000Z',
'20300407T070000Z','20301027T060000Z','20310406T070000Z','20311026T060000Z',
'20320404T070000Z','20321031T060000Z','20330403T070000Z','20331030T060000Z',
'20340402T070000Z','20341029T060000Z','20350401T070000Z','20351028T060000Z',
'20360406T070000Z','20361026T060000Z','20370405T070000Z','20371025T060000Z')
calprops[0] = new CP('pub',
'Public Calendar',
'',
'',
'',
'0',
'0',
'',
'',
'',
'',
'',
'19990817T235829Z',
'19990817T235829Z',
'malika',
new Array(),
new Array(),
new Array())
event[0]=new Ec('TZ_all_params_event',
'0',
'pub',
'19991111T093000',
'19991111T163000',
0,'testing-timezones_events',
'19990817T195829','19990817T195829',
'testing out timezones events',
'Mountain-View',
'America/New_York','')
layer_errno[0]=0
var size=1
var dtstartrange="0"
var dtendrange="0"
errno[0]=0
parent.ceCB(window.name)
</script></head></html>

```

Time Zone Administration

This section describes how to:

- modify the Calendar Server time zone list
- add a new time zone to Calendar Server

To modify the Calendar Server time zone list, an administrator must edit the file `timezones_libnls.ics` located in the `server-root/cal/db/data` directory. This file contains the Calendar Server format of 91 time zones. (See iCalendar Specification for more details about VTIMEZONE format).

Timezones are identified by the property TZID. For example, Calendar Server recognizes the Pacific Standard Time Zone (PST/PDT) as the TZID "America/Los_Angeles". The following time zone defines the "America/Los_Angeles" time zone.

```
BEGIN:VTIMEZONE
TZID:America/Los_Angeles
BEGIN:STANDARD
DTSTART:19671025T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0700
TZOFFSETTO:-0800
TZNAME:PST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZOFFSETFROM:-0800
TZOFFSETTO:-0700
TZNAME:PDT
END:DAYLIGHT
X-NSCP-TZCROSS:19880403T100000Z;19881030T090000Z;19890402T100000Z;19891029T090000Z;
19900401T100000Z;19901028T090000Z;19910407T100000Z;19911027T090000Z;
19920405T100000Z;19921025T090000Z;19930404T100000Z;19931031T090000Z;
19940403T100000Z;19941030T090000Z;19950402T100000Z;19951029T090000Z;
19960407T100000Z;19961027T090000Z;19970406T100000Z;19971026T090000Z;
19980405T100000Z;19981025T090000Z;19990404T100000Z;19991031T090000Z;
20000402T100000Z;20001029T090000Z;20010401T100000Z;20011028T090000Z;
20020407T100000Z;20021027T090000Z;20030406T100000Z;20031026T090000Z;
20040404T100000Z;20041031T090000Z;20050403T100000Z;20051030T090000Z;
20060402T100000Z;20061029T090000Z;20070401T100000Z;20071028T090000Z;
20080406T100000Z;20081026T090000Z;20090405T100000Z;20091025T090000Z;
20100404T100000Z;20101031T090000Z;20110403T100000Z;20111030T090000Z;
20120401T100000Z;20121028T090000Z;20130407T100000Z;20131027T090000Z;
20140406T100000Z;20141026T090000Z;20150405T100000Z;20151025T090000Z;
20160403T100000Z;20161030T090000Z;20170402T100000Z;20171029T090000Z;
20180401T100000Z;20181028T090000Z;20190407T100000Z;20191027T090000Z;
20200405T100000Z;20201025T090000Z;20210404T100000Z;20211031T090000Z;
20220403T100000Z;20221030T090000Z;20230402T100000Z;20231029T090000Z;
20240407T100000Z;20241027T090000Z;20250406T100000Z;20251026T090000Z;
20260405T100000Z;20261025T090000Z;20270404T100000Z;20271031T090000Z;
20280402T100000Z;20281029T090000Z;20290401T100000Z;20291028T090000Z;
20300407T100000Z;20301027T090000Z;20310406T100000Z;20311026T090000Z;
20320404T100000Z;20321031T090000Z;20330403T100000Z;20331030T090000Z;
20340402T100000Z;20341029T090000Z;20350401T100000Z;20351028T090000Z;
20360406T100000Z;20361026T090000Z;20370405T100000Z;20371025T090000Z;
20380404T100000Z;20381031T090000Z;20390403T100000Z;20391030T090000Z
END:VTIMEZONE
```

Time zones with daylight savings usually contain two parts: STANDARD and DAYLIGHT.

Note The "RRULE" property defines the pattern of the STANDARD and DAYLIGHT rules. The "TZOFFSETFROM" and "TZOFFSETTO" properties define the offset from GMT before and after the DAYLIGHT to STANDARD or STANDARD to DAYLIGHT change occurs.

The "TZNAME" property is an abbreviated representation of the time zone. (The iCalendar Specification provides additional information about the representation of VTIMEZONE.)

Adding a New Time zone

The X-NSCP-TZCROSS property contains a large list of dates that indicate when the time zone crosses over from DAYLIGHT to STANDARD and STANDARD to DAYLIGHTT:

- odd number dates indicate STANDARD to DAYLIGHT changes.
- even number dates indicate DAYLIGHT to STANDARD changes.

The Calendar Server's Web browser-based user interface uses the dates in X-NSCP-TZCROSS to signify when to display a change in the time zone.

Use the following steps to add a new time zone to the Calendar Server

1. Create TZID name not already included in the TZID list.
2. Create a representation of the time zone.
3. To support the Calendar Web browser user interface, you must generate the X-NSCP-TZCROSS list for the new time zone.

Modifying an Existing Time Zone

To modify an existing time zone:

1. Modify a TZID to represent the desired time zone data.
2. To support the Calendar Express client, you must generate the X-NSCP-TZCROSS list for the modified time zone.

Customizing Time Zones in the Calendar Server User Interface

To modify the Web browser -based user interface to use a customized naming scheme for the time zone, you must add JavaScript code to map the new names to the time zone names supplied with Calendar Server.

For example, an administrator who wants to use a customized time zone table called `US_Pacific` instead of the supplied `America/Los_Angeles` table, must provide a program to map the `US_Pacific` dates to the `America / Los_Angeles` table.

It is recommended, however, administrators should not modify the time zone list unless absolutely necessary.

Calendar Server Time Zones Table

The Calendar Server time zone table is defined in a plain text file called `timezones_<lib>.ics` located in the `server-root/cal/bin/data` directory (for example, `/opt/SUNWicsrv/cal/bin/data`). It includes 91 time zones based on JDK1.1.

The following example shows the first few sections of the time zone table.

```
BEGIN:VCALENDAR
BEGIN:VTIMEZONE
TZID:Pacific/Apia
BEGIN:STANDARD
DTSTART:19970101T000000
TZOFFSETFROM:-1100
TZOFFSETTO:-1100
TZNAME:WST
TZNAME:SST
TZNAME:NUT
END:STANDARD
END:VTIMEZONE

BEGIN:VTIMEZONE
TZID:Pacific/Honolulu
BEGIN:STANDARD
DTSTART:19970101T000000
TZOFFSETFROM:-1000
TZOFFSETTO:-1000
TZNAME:HST
TZNAME:TKT
TZNAME:TAHT
END:STANDARD
END:VTIMEZONE
```

```
BEGIN:VTIMEZONE
TZID:America/Adak
BEGIN:STANDARD
DTSTART:19671025T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0900
TZOFFSETTO:-1000
TZNAME:HAST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZOFFSETFROM:-1000
TZOFFSETTO:-0900
TZNAME:HADT
END:DAYLIGHT
X-NSCP-TZCROSS:19880403T120000Z;19881030T110000Z;19890402T120000Z;19891029T110000Z;
19900401T120000Z;19901028T110000Z;19910407T120000Z;19911027T110000Z;
19920405T120000Z;19921025T110000Z;19930404T120000Z;19931031T110000Z;
19940403T120000Z;19941030T110000Z;19950402T120000Z;19951029T110000Z;
19960407T120000Z;19961027T110000Z;19970406T120000Z;19971026T110000Z;
19980405T120000Z;19981025T110000Z;19990404T120000Z;19991031T110000Z;
20000402T120000Z;20001029T110000Z;20010401T120000Z;20011028T110000Z;
20020407T120000Z;20021027T110000Z;20030406T120000Z;20031026T110000Z;
20040404T120000Z;20041031T110000Z;20050403T120000Z;20051030T110000Z;
20060402T120000Z;20061029T110000Z;20070401T120000Z;20071028T110000Z;
20080406T120000Z;20081026T110000Z;20090405T120000Z;20091025T110000Z;
20100404T120000Z;20101031T110000Z;20110403T120000Z;20111030T110000Z;
20120401T120000Z;20121028T110000Z;20130407T120000Z;20131027T110000Z;
20140406T120000Z;20141026T110000Z;20150405T120000Z;20151025T110000Z;
20160403T120000Z;20161030T110000Z;20170402T120000Z;20171029T110000Z;
20180401T120000Z;20181028T110000Z;20190407T120000Z;20191027T110000Z;
20200405T120000Z;20201025T110000Z;20210404T120000Z;20211031T110000Z;
20220403T120000Z;20221030T110000Z;20230402T120000Z;20231029T110000Z;
20240407T120000Z;20241027T110000Z;20250406T120000Z;20251026T110000Z;
20260405T120000Z;20261025T110000Z;20270404T120000Z;20271031T110000Z;
20280402T120000Z;20281029T110000Z;20290401T120000Z;20291028T110000Z;
20300407T120000Z;20301027T110000Z;20310406T120000Z;20311026T110000Z;
20320404T120000Z;20321031T110000Z;20330403T120000Z;20331030T110000Z;
20340402T120000Z;20341029T110000Z;20350401T120000Z;20351028T110000Z;
20360406T120000Z;20361026T110000Z;20370405T120000Z;20371025T110000Z
END:VTIMEZONE
```

Calendar Server LDAP Schema

This appendix shows the LDAP schema use by Calendar Server.

```
# Used by Netscape Calendar Hosting Server

# Login calendar URI for this user
attribute nswcalCALID                2.16.840.1.113730.3.1.537 cis
single

# Calendar client specific user preferences for this user
attribute nswcalExtendedUserPrefs    2.16.840.1.113730.3.1.538 cis

# Lists calendar protocols not allowed to be used by this user
attribute nswcalDisallowAccess       2.16.840.1.113730.3.1.539 cis single

# Calendar host for this user's login calendar
attribute nswcalHost                  2.16.840.1.113730.3.1.540 cis

# Quota associated with this user's calendars
attribute nswcalQuota                 2.16.840.1.113730.3.1.541 cis
single

# Used to designate a LDAP entry as representing a Netscape Calendar
# Hosting Server user account. These first 10 attributes are
# referenced by Netscape Calendar Hosting Server and the last 3
# attributes are reserved for future use.
objectClass nswcalUser
```

```
oid 2.16.840.1.113730.3.2.83
```

```
requires
```

```
    objectClass
```

```
allows
```

```
    cn,  
    givenName,  
    mail,  
    preferredlanguage,  
    sn,  
    uid,  
    userPassword,  
    nswcalCALID,  
    nswcalDisallowAccess,  
    nswcalExtendedUserPrefs,  
    nslicensedfor,  
    nswcalHost,  
    nswcalQuota
```

```
# From http://www.imc.org/draft-ietf-calsch-locating
```

```
attribute calCalURI          1.2.840.113556.1.4.478 cis  
attribute calFBURL          1.2.840.113556.1.4.479 cis  
attribute calCAPURI        1.2.840.113556.1.4.480 cis  
attribute calCalAdrURI     1.2.840.113556.1.4.481 cis  
attribute calOtherCalURIs 1.2.840.113556.1.4.482 cis  
attribute calOtherFBURLs   1.2.840.113556.1.4.483 cis  
attribute calOtherCAPURIs  1.2.840.113556.1.4.484 cis  
attribute calOtherCalAdrURIs 1.2.840.113556.1.4.485 cis
```

```
# Used to designate a LDAP entry as representing a calendar user.
```

```
objectClass calEntry
```

```
    oid 1.2.840.113556.1.5.87
```

```
requires
```

```
    objectClass
```

```
allows
```

```
    calCalURI,  
    calFBURL,  
    calCAPURI,  
    calCalAdrURI,  
    calOtherCalURIs,  
    calOtherFBURLs,  
    calOtherCAPURIs,  
    calOtherCalAdrURIs
```

Index

A

- accounts
 - Configuration Administrator 21
 - Directory Manager 21
 - on Unix 21
 - Windows NT administrator 21
- alarms 37
- architectural overview 10
- authentication counter statistics 47
- authstat 47

B

- backup
 - a user's default calendar 63
 - calendars 62, 63
 - databases 64
 - Legato 65
 - Solstice 65
- backup procedures 62

C

- calendars
 - backing up 62
 - backup 63
 - categories 56
 - creating 56
 - cscal 55
 - deleting 57
 - deleting events and tasks 59
 - disabling 57
 - displaying 58
 - enabling 57
 - event and task management 59
 - listing events and tasks 59
 - log information 28

- managing 55
- modifying properties 58
- other owners 56
- primary owner 56
- private 56
- public 56
- removing category values 58
- removing event values 58
- removing property values 58
- removing task values 58
- resetting properties 59
- restoring 63, 64
- user's default 54

- Calendar Server API (CSAPI) configuration 35
- Calendar Server Group 21
- Calendar Server User 21
- calendar store configuration 27
- calmaster 30, 93
- categories 56
 - removing from a value from a calendar 58
- checking users 55
- command line utilities
 - csbackup 75
 - cscal 55, 78
 - cscomponents 81
 - csdb 83
 - csexport 85
 - csimport 87
 - csrestore 88
 - cssession 91
 - csstart 93
 - csstats 95
 - csstop 97
 - cstool 99
 - csuser 101
 - how to use 51
 - summary of 74

- syntax 73
 - usage rules 74
- computer requirements 18
- Configuration Administrator 21
- conventions
 - used in this document 8
- counter.conf file 46
- counter statistics 46, 70
 - authentication 47
 - authstat 47
 - cscounter 46
 - csstats 46
 - database 48
 - dbstat 48
 - HTTP 46
 - httpstat 46
 - WCAP 47
 - wcapstat 47
- creating
 - calendars 56
 - users 53
- csbackup 75
 - calendars 62, 63
 - databases 64
- cscal 78
 - creating calendars 56
 - deleting calendars 57
 - disabling calendars 57
 - displaying calendar properties 58
 - enabling calendars 57
 - managing calendars 55
 - modifying calendar properties 58
 - resetting calendar properties 59
- cscomponent
 - event and task management 59
- cscomponents 81
- cscounter
 - counter statistics 46
- csdb 83
 - database management 60
 - recovering databases 65
- csdb command line utility 65

- csexport 85
 - exporting data 61
- csimport 87
 - importing data 61
- csrestore 88
 - calendars 63, 64
 - databases 65
- cssession 91
 - listing user status 69
 - logging out a user 70
 - managing users 53
- cssession command line utility 53
- csstart 93
- csstats 95
 - counter statistics 46
 - listing counter statistics 70
- csstop 97
- cstool 99
 - pinging the server 71
 - refreshing server configuration 71
- csuser 101
 - checking user logon status 55
 - disabling a user 54
 - displaying user information 55
 - enabling a user 54
 - managing users 52
 - resetting user attributes 55
- csuser command line utility 52
- custom install, defined 20

D

- database configuration 34
- database counter statistics 48
- databases
 - backing up 64
 - creating 60
 - deleting 60
 - listing number of reads, writes and deletes 70
 - management 60
 - recovering 60, 65
 - restoring 63, 65

- viewing status and statistics 60
- dbstat 48
- default calendars
 - backing up 63
 - enabling 54
 - restoring 64
- deleting calendars 57
- Directory Manager 21
- directory server
 - installation requirements 18
- Directory Server Accounts 21
- disabling a user 54
- disabling calendars 57
- displaying calendar properties 58
- dkstat 107
- document conventions 8
 - monospaced font 8
 - sidebar text 8

E

- enabling
 - calendars 57
 - default calendars 54
 - users 54
- events
 - deleting 59
 - listing 59
 - managing 59
 - removing a value from a calendar 58
- exporting calendar data 61

F

- failed logins 70

G

- glance 107
- gmemusage (Irix 6.x) 107
- gpm 107

- group account (Unix) 21

H

- HTTP counter statistics 46
- httpstat 46

I

- icsgroup 21, 22
- icsuser 21, 22
- identity, to run server as on Unix 21
- importing calendar data 61
- installation
 - components 19
 - preparing for 18
 - types of 20
- iostat 105
- iplncal.sh (Unix only) 50

L

- landiag 107
- LDAP schema 117
- Legato 65
- listing all users 55
- local configuration settings 26
- lockstat 106
- logging out a user 70
- logging out a user session 70
- log information settings 28
- logon status 55
 - listing 69
- lslk 105
- lsdf 105

M

- managing
 - calendars 55
 - events 59

- tasks 59
- users 52, 53
- modifying calendar properties 58
- monitoring server activity 69
- monospaced font 8
- mpstat 106

N

- netstat 105, 107
- notification messages 37
- nslookup 105
- NT system account 21

O

- osview 108
- other owners (of a calendar) 56
- overview, of architecture 10

P

- par 108
- Performance Copilot 108
- ping 105
- pinging the server 71
- ports
 - using Custom Installation 20
 - using default numbers 19
- preparing for installation 18
- primary owner (of a calendar) 56
- private calendars 56
- proctool 106
- properties
 - removing a value from a calendar 58
- public calendars 56

R

- recovering

- a damaged database 60
- databases 65
- refreshing server configuration 71
- removing
 - property values from a calendar 58
- resetting calendar properties 59
- restore procedures 62
- restoring
 - a user's default calendar 64
 - calendars 63, 64
 - calendars from a database 63
 - calendars from a file 63
 - databases 65

S

- sam 107
- sar 106
- schema 117
- server.conf file
 - calendar log information settings 28
 - calendar store configuration settings 27
 - CSAPI configuration settings 35
 - database configuration settings 34
 - local configuration settings 26
 - services configuration settings 29
 - using 25
- server-root
 - defined 9
- servers
 - computer requirements for 18
- services configuration 29
- sessions 69, 70
- sidebar text 8
- silent install, defined 20
- snoop 106
- Solstice 65
- start-cal 50
- starting the server
 - on Unix 50
 - on Windows NT 51

- start-cal 50
 - using iplncal.sh 50
- stop-cal
 - stopping the server 50
- SymbEL/Virtual Adrian 107
- sysdef 107
- system accounts
 - on Unix 21
 - Windows NT 21
- system monitoring tools 105
- system requirements 18

T

- tasks
 - deleting 59
 - listing 59
 - managing 59
 - removing from a value from a calendar 58
- tcpdump 106
- time zones 109
 - adding 114
 - modifying 112, 114
 - tzid format 109
- top 106
- trace 106
- traceroute 106
- transferring installation files 22
- truss 107
- tusc 107
- typical install, defined 20
- tzid 109

U

- Unix
 - user account 21
- Unix group account 21
- Unix system accounts 21
- unrestricted user 21

- user account (Unix) 21
- users
 - checking logon status 55
 - creating 53
 - cssession utility 53
 - csuser utility 52
 - default calendars 54
 - disabling 54
 - displaying information about 55
 - listing all logged in 55
 - logon status 55
 - managing 52, 53
 - resetting attributes 55
- user sessions 69

V

- vmstat 106

W

- WCAP counter statistics 47
- wcapstat 47
- Windows NT administrator 21

X

- X-NSCP-TZCROSS property 114

