Administration Guide

iPlanet Calendar Server

806-5860-10
November, 2000

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 5.0

Part Number: 806-5860-10 Rev B

Contents

Preface

This manual explains how to administer iPlanet Calendar Server 5.0 and its accompanying software components. iPlanet Calendar Server is a powerful and flexible cross-platform solution using open Internet standards that lets service providers of all sizes host group scheduling and personal calendars for their customers.

Who Should Read This Book

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

What You Need to Know

Before you install and administer iPlanet 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 iPlanet Calendar Server and how to get started.	Chapter 1, “Architecture Overview

If you want to	See
Configure your iPlanet Calendar Server.	Chapter 2, “Configuring iPlanet Calendar Server
Administer your iPlanet Calendar Server using the command line utilities.	Chapter 3, “Administering iPlanet Calendar Server
Monitor and control your iPlanet Calendar Server.	Chapter 4, “Monitoring the iPlanet 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 iPlanet Calendar Server time zones.	Appendix C, “Time Zones”
Find information in the guide.	Index

Document Conventions

`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.

`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 represents text that you enter using information that is unique to your installation. It is used for server directory paths and names as well as account IDs. For example, throughout this document you will see directory path references in the form:

`server-root``/opt/cal/bin`

In these situations, *`server-root`* represents the directory location in which you installed the server. For example:

- The default installation on Solaris uses the path: `/opt/SUNWics5/cal` as the *`server-root`* for the software files. Therefore, if you choose the default installation directory, *`server-root/`*`cal` represents:

  `/opt/SUNWics5/cal`

- On Unix systems other than Solaris, the default installation location is: `/opt/iPlanet/CalendarServer5/cal`, in which case *server-root*/`cal` represents:

  `/opt/iPlanet/CalendarServer5/cal`

- On Windows NT, the default installation location is: `c:\Program Files\iPlanet\CalendarServer5\cal`, in which case *server-root*/`cal` represents:

  `c:\Program Files\iPlanet\CalendarServer5\cal`

NOTE In this guide, the conventions for a Solaris installation are used. If you are running iPlanet Calendar Server on another Unix operating system or on Windows NT, you should assume the file path equivalents of your operating system whenever Solaris file paths are shown.

Chapter 1

Architecture Overview

iPlanet Calendar Server 5.0 is a powerful and flexible cross-platform solution using open Internet standards that lets service providers of all sizes host personal and group scheduling calendars for their customers.

Topics covered in this chapter are:

- What is iPlanet Calendar Server?
- What's New in Version 5.0?
- Calendar Server Services
- Group Scheduling
- Horizontal Scalability
- New Default Client UI: SHTML
- Architecture Basics
- Calendar Data
- Calendar Server API (CSAPI)
- Event Notification Service (ENS)
- Proxy Authentication SDK (authSDK)
- Single Sign-on (SSO)
- Web Calendar Access Protocol (WCAP)

What is iPlanet Calendar Server?

iPlanet Calendar Server is a readily deployable, LDAP-based solution that lets the ISP and Telecommunication service provider offer group scheduling features, as well as host personal event calendars, to their base of subscribed customers.

iPlanet Calendar Server 5.0 is a system of servers that can be configured to fit a variety of needs. It can stand in isolation as a stand-alone calendar server, or it can be configured with many instances, having the various services duplicated or split between them as was discussed earlier in this chapter. See "Horizontal Scalability," on page 20. iPlanet Calendar Server 5.0 stores and manages calendars, calendar properties, access control information, events, todos, and alarms. It does not manage storage for user information. The minimal system requires a directory service. It makes use of plug-ins to obtain external services. It uses a directory service to perform operations such as authentication, and storage and retrieval of user preferences. iPlanet Calendar Server ships with plug-ins to perform directory services using LDAP services. You may use your own plug-ins to support non-LDAP directory services.

Figure 1-1 shows a minimal iPlanet Calendar Server (iCS) 5.0 system. It consists of a single iPlanet Calendar Server instance, a directory service, and support for event notifications.

Figure 1-1 Minimal iPlanet Calendar Server System

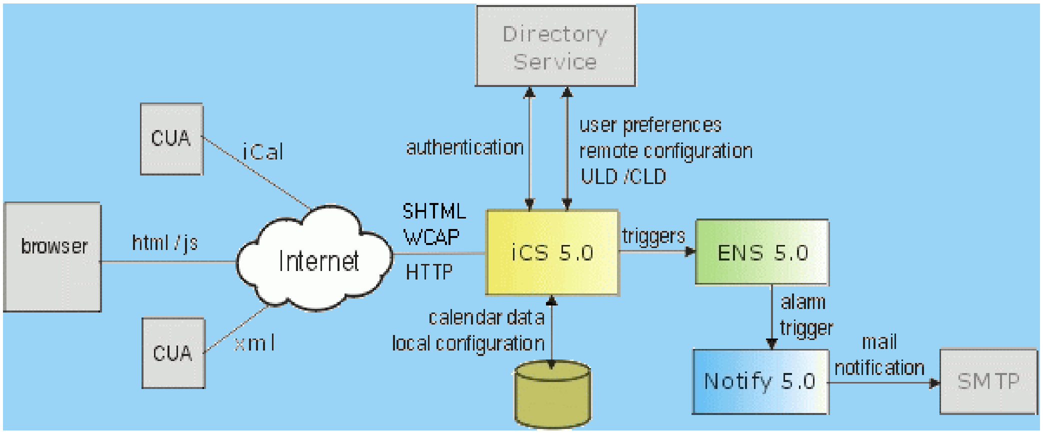


Table 1-1 is the key to the abbreviations in Figure 1-1.

Table 1-1 iPlanet Calendar Server System Key to Terms

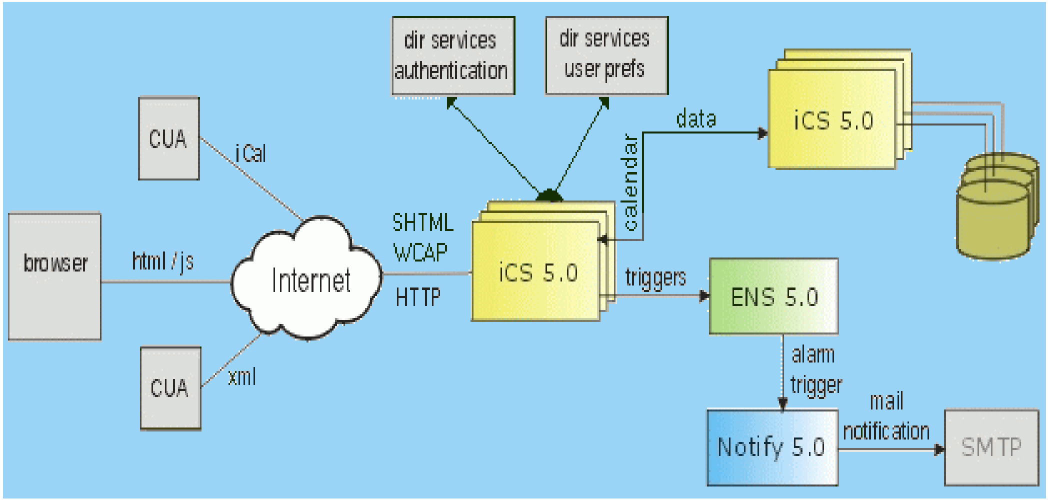
CLD	Calendar Lookup Database

Table 1-1 iPlanet Calendar Server System Key to Terms

CUA	Calendar User Agent
ENS	Event Notification Server
iCal	iCalendar-RFC 2445

iPlanet Calendar Server is scalable both vertically and horizontally. On a single machine it can run in multiple processors, or it can be split up to run on multiple machines. Figure 1-2 depicts three iPlanet Calendar Server HTTP "front end" services, using three other iPlanet Calendar Server "database" services. All six of these instances can run on separate machines. The abbreviation "dwp" in this figure stands for Database Wire Protocol. For further information on this new protocol and the various configuration possibilities using horizontal scalability, see "Horizontal Scalability," on page 20.

Figure 1-2 Scalable System



Summary of Features

Built from the ground up to provide native support for the emerging set of internet calendar standards, iPlanet Calendar Server 5.0 provides the following benefits:

- **Group Scheduling**–Organizers create an event, and invite attendees. Attendees accept or decline invitations. If the attendee is not on the calendar server, the group scheduling engine can send the scheduling message via email as an IMIP message (as described in RFC2447).
- **Internet Calendaring and Scheduling**–Native support for iCalendar calendaring standards ensures events are in a format that is easily shared across the internet.
- **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.
- **Massively Scalable**–iPlanet Calendar Server 5.0 is scalable both vertically and horizontally.

  Scaling vertically to the requirements of the largest service providers, it supports a hosting environment of up to several million personal event calendars. It runs on a single machine in multiple processors to take advantage of all the machine's processing power.

  Scaling horizontally, it also splits up to run on several computers in many combinations. For example, there could be three Calendar Server HTTP services utilizing three other Calendar Server database services, each running on a separate machine. For further information on this new feature, see "Horizontal Scalability," on page 20.

What's New in Version 5.0?

iPlanet Calendar Server 5.0 adds several new features:

- **Group scheduling**–In addition to keeping personal calendar information, users can now invite other calendar users to meetings who in turn may accept or decline the request. For more details on this new feature, see "Group Scheduling," on page 20.
- **Horizontal scalability**–The server can run on a single machine or its processes can be divided across multiple machines with a wide variety of possible configuration options. For more details on this new feature, see "Horizontal Scalability," on page 20.

- **New default client user interface**–Calendar Express, the bundled calendar client UI, now uses SHTML which achieves quicker browser rendering and response times as well as faster and easier customization that you can tailor to your site.
- **Migration from iPlanet Calendar Server 2.x**–A bundled utility that allows an administrator to import data from an existing iPlanet Calendar Server 2.x installation. The migration process is accomplished by running a separate program after the installation of version 5.0 has been completed successfully. For more details, see the *iPlanet Calendar Server Installation Guide.*

iPlanet Calendar Server 5.0 builds on the consumer focus of iPlanet Calendar Server 2.x with the introduction of multiple-machine horizontal scalability as the design focus.This new architecture allows group scheduling capabilities using notifications. To implement notifications, iPlanet Calendar Server ships with Event Notification Service (ENS). A key benefit of this architecture is that you may customize many of the components, or even develop your own applications based on your customer’s needs.

These new features required fundamental changes in some of the interfaces and tools described in this reference. Additional customization interfaces have been added to this reference. New this time are the ENS API and the Proxy Authentication SDK.

For continuity, you can still use WCAP, which supports all iPlanet Calendar Server 2.x data formats.

Calendar Server Services

The iPlanet Calendar Server 5.0 system consists of several running processes that include multiple calendar server daemons which can run on a single machine or be divided to run on multiple machines. The specific combination of modules for a given running instance are defined in the server configuration information stored in the file `ics.conf` and can be modified using command line administration tools.

The administration design supports a single service on a single machine or multiple services on multiple machines. The administrator can also issue commands remotely from a machine other than where the calendar service is running. Table 1-2 lists the Five iPlanet Calendar Server 5.0 daemons.

Table 1-2 iPlanet Calendar Server Daemons

`csadmind`	Administration service. Includes the Group Scheduling Engine (GSE), and monitors for alarms.
`csdwpd`	Interprocess database service.
`cshttpd`	HTTP service. Services SHTML and WCAP requests.
`csnotifyd`	Notification service. This is required in instances where the database resides.
`enpd`	Event Notification Service.

csadmind

This service provides alarm notifications, group scheduling requests, database checkpointing and deadlock detection, as well as disk usage and server response monitoring.

cshttpd

iPlanet Calendar Server 5.0 uses HTTP as its primary transport. This service listens for HTTP commands and retrieves and returns data to the caller. For the new 5.0 user interface, commands received with the default .shtml extension returns data formatted in HTML. Alternately, for requests received with the .wcap extension, data can be formatted either as raw calendar data in standard RFC2445 iCalendar, XML, or JavaScript embedded in HTML.

csnotifyd

The notification service (`csnotifyd`) sends calendar-based notifications of events and tasks and utilizes the Event Notification Server (ENS) as the broker for events. It subscribes to alarm events. When an alarm event occurs, it sends an SMTP message reminder to the recipients. For more information, see “Event Notification Service (ENS) Overview” in the *iPlanet Calendar Server Programmer’s Reference*.

csdwpd

This service allows multiple machines within the same system to be linked together to form a distributed calendar store. The service can run in the background on any machine on which iPlanet Calendar Server 5.0 is installed. It acts as a service accepting requests that abide by the Database Wire Protocol (DWP) for calendaring information.

This service should be run only on a server that:

- Has a local calendar store.
- Must provide network access to its calendar data from other iPlanet Calendar Server installations.

NOTE	this should only be done on a fast network. If the pipe (network) between the various databases is slow, it can seriously degrade overall system performance.

enpd

This is the other half of the Event Notification Service. It acts as the broker for event alarms. It receives notifications of alarms from the `csadmind` daemon, checks for subscriptions to this event, and notifies the event's subscribers by passing the subscribed-to alarm notifications to `csnotifyd`. It also receives and stores subscriptions and cancellations of subscriptions (unsubscribe) from `csnotifyd`.

Start Order

The iPlanet Calendar Server 5.0 daemons must be started in a specific order:

1. `enpd`–A generic event registration and notification service that can be shared by other iPlanet servers
2. `csnotifyd`–Calendar Server Notification Daemon
3. `csadmind`–Calendar Server Administration daemon (installation required on every server machine)
4. `csdwpd`–Calendar Server DataBase Daemon (only started with remote database configuration)
5. `cshttpd`–Calendar Server Daemon (at least one is required)

Group Scheduling

In version 2.x, you could schedule events and todos on your own calendar and share your calendar with others, but with iPlanet Calendar Server 5.0, you may now make scheduling requests for other calendars. You may schedule events and invite attendees and they may accept or decline the request. When an attendee accepts or declines, the server updates the calendars of all attendees, as well as the event organizer. Attendees not in the installation's database are notified by email.

Directory Server Services

By default, iPlanet Calendar Server supports users that are defined and maintained in an LDAP directory, such as Netscape Directory Server. iPlanet Calendar Server also supports the use of CSAPI plug-ins that you create to enable access for 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 for deploying iPlanet Calendar Server is to upgrade your directory server to Netscape Directory Server 4.12, and iPlanet Calendar Server installation will do the rest. Otherwise, you can modify your directory schema manually to allow your users to access iPlanet Calendar Server data. For more information on how to modify a directory schema for iPlanet Calendar Server, see "Installing and Configuring an LDAP Server" in the *iPlanet Calendar Server Installation Guide*.

Horizontal Scalability

Horizontal scalability may be achieved by spreading an installation over several machines. iPlanet Calendar Server consists of the daemons `cshttpd`, `csadmind`, `csdwpd`, `csnotifyd`, and `enpd`. These daemons may be run in different configurations to allow you great flexibility and scalability.

To facilitate horizontal scalability in iPlanet Calendar Server 5.0, the system employs an internal proprietary protocol, Database Wire Protocol (DWP), `csdwpd`. It was necessary to implement this protocol because the iPlanet Calendar Server 5.0 product uses a default Berkeley DB, which is not a networked database. The DWP protocol uses HTTP as it's base. The implementation is simply an HTTP `POST` or `GET` command, with a single binary MIME part that contains serialized binary database information.

In a future release, the calendar database API will be published. It can be implemented using any database technology. DWP will not be needed by any implementation that supports a networked database.

Configurations

To achieve horizontal scalability, you install various instances of iPlanet Calendar Server 5.0 across your machines. The basic requirements for every system are:

- Each instance must have `csadmind`
- All of the other daemons are required to be installed at least once.

  The exception to this is the case of a single-instance installation using a local database connection. In this simple case, the `csdwpd` daemon is not necessary.

Table 1-3 indicates three possible configurations and the services you need to install for each instance. The figures that follow the table illustrate these configurations. Other configurations are possible. You must determine what combination addresses your specific needs.

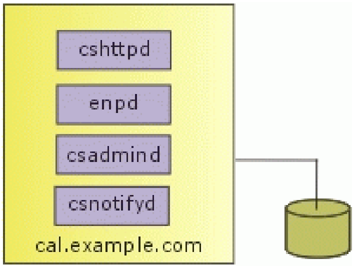
Table 1-3 Instance Configuration of Required Services

Instances	**Required Services**				
	`csadmind`	**`cshttpd`**	**`csnotifyd`**	**`csdwp`**	**`enpd`**
Simple Single Instance installation with a local database. See Figure 1-3 that follows.	Y	Y	Y	N	Y
HTTP Service-only instances. (Used only in conjunction with Database Service-only instances.) Examples: See boxes on left side of Figure 1-4, Network Front End, Database Back End, and Figure 1-5, Multiple Front Ends, Multiple Back Ends	Y	Y	N	N	N
Database Service-only instances. (Used only in conjunction with HTTP Service-only instances.) Examples: See boxes on right side of Figure 1-4, Network Front End, Database Back End, and Figure 1-5, Multiple Front Ends, Multiple Back Ends	Y	N	Y	Y	Y

Simple Single Instance

This is the simplest configuration in which iPlanet Calendar Server 5.0 can run. As illustrated in Figure 1-3, it consists of `cshttpd` to handle incoming SHTML and WCAP requests, `enpd` and `csnotifyd` for event notification, and the required `csadmind`. The entire database is local.

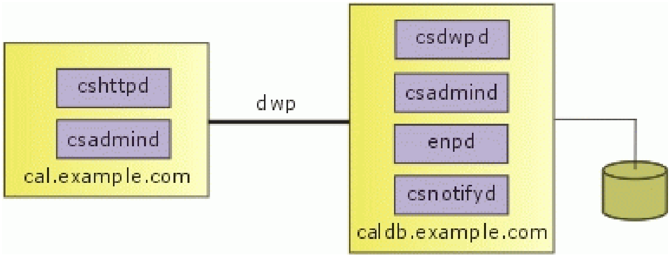
Figure 1-3 Simple Single Instance Configuration



Network Front End, Database Back End

In this configuration, illustrated in Figure 1-4, browsers and other clients connect to the calendar server at the HTTP Service front-end, labeled `cal.example.com` in this example. All requests for calendar data are routed to the Database Service, labeled `caldb.example.com` in this example. Notice that the front end requires only `cshttpd` and `csadmind` since it is not doing any database processing. The back end, conversely, does not need `cshttpd`, but does require `csdwpd`, `enpd`, and `csnotifyd`.

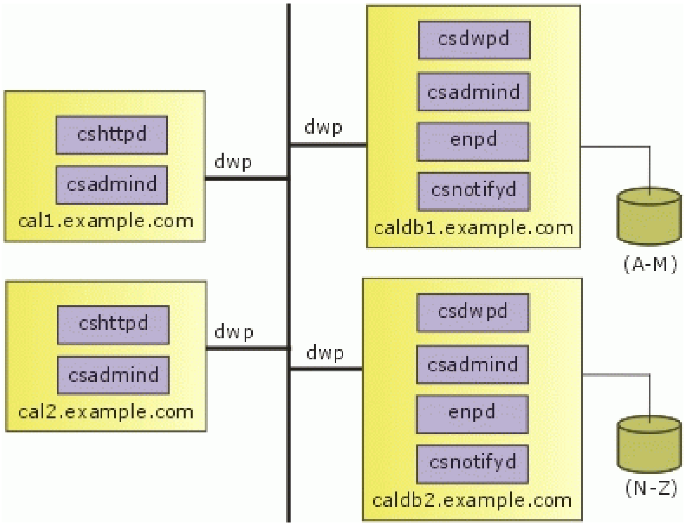
Figure 1-4 Network Front End, Database Back End



Multiple Front Ends, Multiple Back Ends

In this configuration, illustrated in Figure 1-5, clients are routed to one of the front end HTTP Services by some external mechanism that you provide. Please note that the session ID returned at login is valid only on the host where the login occurred. All requests for this session ID must be routed to the same host or the user will be forced to login again. In this example, the database has also been split, with calendars `A-M` on the `caldb1.example.com` server, and calendars `N-Z` on the `caldb2.example.com` server. A CSAPI plug-in handles mapping between calendar IDs and the name of the server on which it can be found. The default CSAPI implementation provided in iPlanet Calendar Server 5.0 uses an algorithm to associate a calendar ID with a server name.

Figure 1-5 Multiple Front Ends, Multiple Back Ends



New Default Client UI: SHTML

iPlanet Calendar Server 5.0 no longer implements the default client UI with the WCAP protocol as it did in version 2.x. The WCAP protocol generated a mixture of HTML and JavaScript, and passed it to the client for processing. Using the new SHTML commands, the server now does all of the processing, and generates and sends only formatted output (HTML) to the client. The new SHTML commands use XML prototype definitions and XSL style-sheet templates to generate HTML. For each view and dialog displayed in the UI, there is one or more corresponding pairs of text files. Each pair consists of one .xml, and one .xsl file. You may alter or replace one or both of these files in order to customize the UI.

If you already have a custom user interface that was developed for version 2.x, you may continue to use it without change. iPlanet Calendar Server 5.0 is fully backward compatible with iPlanet Calendar Server 2.x.

WCAP remains the only way to retrieve unprocessed calendar data. Clients that need raw, unformatted calendar information should submit requests in WCAP format.

Architecture Basics

iPlanet Calendar Server 5.0 is implemented by way of a collection of shared libraries. These shared libraries are bound in various combinations to produce the executable daemons `cshttpd`, `csdwpd`, `csadmind`, and `csnotifyd`.

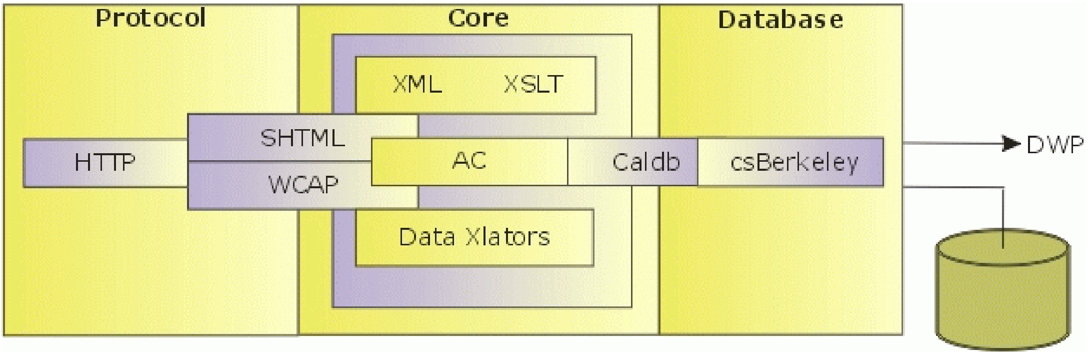
The Event Notification Service (ENS) daemon, `enpd`, is a separately installed service shipped with iPlanet Calendar Server 5.0. For further information on ENS, see“Event Notification Service (ENS),” on page 31, and Chapters 4 and 5 the *iPlanet Calendar Server Programmer’s Reference*.

The shared libraries fall into three main categories, called *subsystems*:

- Protocol
- Core
- Database

Figure 1-6 represents the logical flow through these subsystems. A section follows on each of the subsystems.

Figure 1-6 Server Architecture



SHTML and WCAP

SHTML and WCAP are based on HTTP. Requests enter through the HTTP protocol layer. This is a minimal HTTP server implementation, streamlined to support calendar requests. Clients use either SHTML or WCAP to submit requests. WCAP is an open protocol that can perform all server commands (except for certain administrative commands). It can be used by clients that need raw, unformatted calendar information. It can also be used to obtain a JavaScript based user interface. This user interface was the only one available in iPlanet Calendar Server 2.x, but in version 5.0, it has been replaced with a new SHTML-based user interface. This new approach, based on XML and XSLT specifications, generates a user interface in response to commands. In response to an incoming request, the UI generator uses an XML specification to build a document tree with calendar and user data, subject to access control. The XSLT specification then traverses the document data tree and emits HTML. One of the benefits of this design is that it results in fewer interactions between the client and server and fewer bits being sent overall. Previously, in iPlanet Calendar Server 2.x, the system returned embedded JavaScript, which generated HTML at the client. The XML/XSLT approach generates output that is rendered faster by browsers.

For backward compatibility, the iPlanet Calendar Server 2.x design will continue to work, with WCAP requests returning a combination of HTML text and JavaScript as the default. As in iPlanet Calendar Server 2.x, commands using the .wcap extension may also request output as XML wrapped in HTML, or as iCalendar wrapped in HTML.

For a description of the WCAP protocol, see the *iPlanet Calendar Server Programmer's Reference*.

Core

Inside the Core subsystem, other divisions include the Access Control subsystem, the UI Generator subsystem (either SHTML, using XML and XSLT, or WCAP, using data translators), and a Caldb Subsystem. CSAPI plug-ins reside in the Core.

The protocol processes the command and sends it to the Core for execution. There are four types of commands:

- **Calendar manipulation**–For Calendar manipulations requiring database actions, the protocol sends requests to the Core's Access Control subsystem. The Access Control subsystem uses the generic Caldb subsystem to perform calendar read and write operations.

The Caldb subsystem uses a database-technology-specific subsystem to perform its operations. For this version of iPlanet Calendar Server with the Berkeley database implementation, this subsystem is capable of dealing with a local database file or making Database Wire Protocol (DWP) requests to the appropriate machine.

The Database subsystem returns data in a low-level format. The Core UI generator (either SHTML or WCAP) translates the low-level data into the desired output. Commands with the .shtml extension always default to HTML. Commands with the .wcap extension return the output format requested. WCAP remains the only open protocol the server supports. Use it to retrieve unformatted calendar data.

NOTE `HTML/js`, which was the default UI output for iPlanet Calendar Server 2.x, has been deprecated in favor of .shtml for iPlanet Calendar Server 5.0.

- **User attributes**–Requests for user attributes go to the Core's directory service. iPlanet Calendar Server 5.0 makes use of plug-ins to obtain external services, such as directory services. The product ships with a plug-in for LDAP services. To customize your installation, you can write other plug-ins to support non-LDAP directory services.
- **Authentication**–Requests for authentication go to the Core's default LDAP directory service. iPlanet Calendar Server 5.0 ships with three different authentication options:
  - CSAPI authentication Internal authentication.
  - Proxy Authentication SDK. External plug-in authentication.
  - Single Sign-on authentication. Applications form circles of trust in a single domain.

  An API for customization exists for CSAPI authentication and the Proxy Authentication SDK. For more information, see the *iPlanet Calendar Server Programmer's Reference.*
- **Miscellaneous**–Requests for miscellaneous services go to other Core subsystems.

Database

iPlanet Calendar Server 5.0 uses the csBerkeley subsystem using the Berkeley DB from Sleepycat. The database API is not public.

Calendar Data

This section describes various aspects of calendar data:

- Calendar Data Format
- Groups
- Event Feeds
- Calendar Data Exchange
- Calendar User Preferences
- Calendar Access Control
- Supported Format Encoding

Calendar Data Format

Calendar data format is modeled after the IETF iCalendar standard RFC-2445. The Access Control layer maintains calendars, which are collections of iCalendar components. The components include events, todos, and alarms. A calendar has one primary owner, and may have other owners. User attributes are maintained by an external mechanism. The default mechanism is LDAP.

Groups

Groups are named list of calendars. Users subscribe to calendars. A user can view and modify a subscribed calendar, subject to access control. Additionally, users can create groups. They can then work with groups of calendars, rather than having to re-specify or select a list of calendars every time they wish to view them side-by-side, or invite their owners to an event.

Groups allow multiple calendar sources to be aggregated into a single calendar for display purposes. A user can, for example, have a default calendar view made up of his or her own calendar, the department calendar, the holidays calendar, and the “latest action video releases” calendar, all of which is called a “calendar group”.

Event Feeds

This infrastructure makes it possible to feed real-time event data into the database using import formats such as iCalendar or XML. Event data can be fed in from calendars such as the San Francisco 49ers season schedule, convention center schedules, concert schedules, or any schedule of events that may be of interest.

iPlanet Calendar Server provides tools for retrieving calendar data from event feeds and from the database, so that users can view and retrieve event information. The user can layer these events onto their own calendar views, providing rich event information that is maintained by third parties.

Calendar Data Exchange

All calendars and events can be referenced as URLs. Users can embed these links in email messages and web pages. Users can click on a link to see a monthly calendar view, a weekly view, a daily view, or a specific event. If the calendars are publicly readable, users will not be asked to log in.

iPlanet Calendar Server supports server-side email alarms, which can be sent to a list of recipients. The format of the email message is completely configurable. It is maintained as a server attribute, rather than as a user or calendar attribute. iPlanet Calendar Server 5.0 has limited support for the ITIP / IMIP standards [RFC-2446, RFC-2447]. It supports ITIP methods `PUBLISH`, `REQUEST`, `REPLY`, and `CANCEL` for events.

Calendar User Preferences

iPlanet Calendar Server customizes the display of calendaring information for each user according to attributes called user preferences. User preferences, as opposed to calendar preferences, refer to the user interface representation of information. User preferences include such things as email address, user name, and preferred colors to use when rendering calendar information.

Calendar Access Control

In iPlanet Calendar Server 5.0, access control is the mechanism that determines who can access a calendar when performing group scheduling. An Access Control Entry (ACE) string specifies the type of access privileges to a calendar that are granted to a user. These strings are stored in the access control calendar property `acl` (Access Control List) and collectively determine the access control of a calendar. Only users with write access to a calendar's properties can successfully change these strings. By default, only the Calendar Server administrator and the primary owner of a calendar have write access to its properties. The iPlanet Calendar Server access control model also supports the ability to act in behalf of others. For example, with this type of access granted to them, administrative assistants can invite, cancel, and reply to events on behalf of people they support.

Access control is described in more detail in the following sections:

- "Access Control," on page 81.
- "Managing Calendars," on page 94.
- "cscal," on page 121.

Supported Format Encoding

For data being stored, the Core translates the input into the binary form that Caldb uses. iPlanet Calendar Server supports the following format encodings:

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

You can add other formats by developing your own XSL translations for the UI views and dialogs, or, using CSAPI, you can develop a translator DLL or shared library for the WCAP protocol. (See the *iPlanet Calendar Server Programmer's Reference* for details on CSAP.

Calendar Server API (CSAPI)

CSAPI is a COM-like interface that allows programmers to implement customized parts of the server. Use CSAPI to modify the following areas of functionality:

- Access Control
- Authentication
- Calendar Lookup
- Data Format Translation
- User Attribute Access

For more information on CSAPI, see the *iPlanet Calendar Server Programmer's Reference*.

Event Notification Service (ENS)

In iPlanet Calendar Server, the primary use of the Event Notification Service (ENS) is as an alarm dispatcher, which detects events on an alarm queue and sends notifications of these events to its subscribers.

Proxy Authentication SDK (authSDK)

The authSDK is one of the three tools iPlanet Calendar Server 5.0 offers for user authentication. With authSDK, you can integrate your existing portal service with iPlanet Calendar Server 5.0, thus allowing your users to access various applications without the necessity of re-authentication. The authSDK consists of five functions packaged in a DLL/shared-object library, `libicsexp10`, and a header file, `expapi.h`. These functions perform three simple tasks:

- Initialization
- Lookup
- Cleanup

In addition, two other functions allow you to use a non-standard port, and to get the authSDK version number for troubleshooting.

NOTE Once a connection has been established between iPlanet Calendar Server and the authSDK, the relationship set in place is one of trust. Therefore, once a user has logged in and has successfully authenticated to the authSDK, Calendar Server accepts the certificate generated by the proxy for all of its applications.

Single Sign-on (SSO)

Single Sign-on (SSO) is one of the three authentication mechanisms offered by iPlanet Calendar Server 5.0. To use Single Sign-on, the client browser must support cookies and your server must support HTTP. Single Sign-on is independent from any other authentication mechanisms, session management, and resource access control.

With Single Sign-on, applications form circles of trust that share cookies and accept each others' user authentication. Each application can have its own verification interface, if necessary. Each verification authority, however, stores a cookie that is understood by the other applications' verification authority routines.

There are some limitations to this mechanism. The primary limitation is that each application must implement the verification protocol. Also, all trusted applications need to be in the same domain and on the same machine. And, finally, to switch to a different identity, the user must restart the browser because each browser session can support only one user ID.

Web Calendar Access Protocol (WCAP)

WCAP is a command based system consisting of client requests and server responses for transmitting calendaring data. WCAP 2.0 returns calendaring data via HTTP. With WCAP commands, you can get, delete, and modify calendar components, user preferences, calendar properties, and other calendar information like time zones. All times, strings, parameters, etc. follow RFC2445, RFC2446, and RFC2447 specifications, unless otherwise specified.

One of the three user authentication mechanisms offered by iPlanet Calendar Server 5.0 is the WCAP default of plain-text passwords and user names. You may replace or augment the authentication mechanism that WCAP uses.

WCAP supports the following client request and server response data formats:

- Calendar data in plain text format (HTML only). This is the new default format for the UI.
- Calendar data in `text/calendar` format (iCalendar).
- Calendar data in `text/xml` format. An XML-style version of iCalendar.
- Calendar data as `text/js` with embedded JavaScript objects. This was the default for the iPlanet Calendar Server2.x user interface.

Chapter 2

Configuring iPlanet Calendar Server

This chapter describes the settings you can configure for iPlanet Calendar Server 5.0. It includes the following sections:

- Using the ics.conf file
- Local Configuration
- Calendar Store Configuration
- Calendar Log Information Configuration
- Services Configuration
- Alarm Notification Configuration
- Calendar Store Configuration
- Calendar Locate Mechanism Configuration
- Single Sign-on Configuration
- Group Scheduling Configuration
- Database Configuration
- Calendar Server API Configuration
- Event Notification Server Configuration
- User Interface Configuration
- Notification Messages
- Counters

Using the ics.conf file

iPlanet Calendar Server configuration settings are stored in file:

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

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

- All parameters are in lower case only.
- A parameter and its value(s) are separated by an equal sign (=) with spaces and tabs allowed before or after the equal sign.
- A value must be enclosed in single quotes (" "). If a parameter allows multiple values, the entire value string must be enclosed in single quotes.
- A comment line must have an exclamation point (!) as the first character of the line. Comment lines are for informational purposes only and are ignored by the server.
- If a parameter appears more than once, the value of the last parameter listed overrides the previous value.

If you make changes to the ics.conf file, you must shut down and restart the Calendar Server in order for your new configuration settings to take effect.

.

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

Local Configuration

Table 2-1 describes the local configuration settings stored in ics.conf.

Table 2-1 Local configuration settings stored in ics.conf.

Parameter	Default Value	Description
local.authldapbasedn	" "	BASEDN for LDAP authentication. If not specified, local.ugldapbasedn is used.
local.authldaphost	"localhost"	Host for LDAP authentication. If not specified, local.ugldaphost is used.

Table 2-1 Local configuration settings stored in `ics.conf`. *(Continued)*

Parameter	Default Value	Description
`local.authldapbindcred`	`" "`	Bind credentials (password) for user specified in `local.authldapbinddn`.
`local.authldapbinddn`	`" "`	DN used to bind to LDAP authentication host to search for user's dn. If not specified or `" "`, anonymous bind.
`local.authldapport`	`"389"`	Port for LDAP authentication. If not specified, `local.ugldapport` is used.
`local.authldappoolsize`	`"1"`	Minimum number of LDAP client connections that are maintained for LDAP authentication. If not specified, `local.ugldappoolsize` is used.
`local.authldapmaxpool`	`"1024"`	Maximum number of LDAP client connections that are maintained for LDAP authentication. If not specified, `local.ugldapmaxpool` is used.
`local.caldb.deadlock.autodetect`	`"no"`	Periodically checks if the Berkeley database is in a deadlock state and, if so, instructs the database to reset.
`local.enduseradmincred`	`" "`	Bind credentials (password) for LDAP user preferences authentication.
`local.enduseradmindn`	`" "`	DN used to bind to LDAP user preferences host. Must be specified. If " " (not specified), anonymous bind is assumed.
`local.hostname`	`" "`	Host name of the machine on which iPlanet Calendar Server is installed.
`local.installeddir`	`" "`	Directory path location where iPlanet Calendar Server is installed.
`local.instancedir`	`"."`	Directory path location where programs and data for this instance of iPlanet Calendar Server are installed.
`local.plugindir.path`	`" "`	Directory path location where CSAPI plugins for this instance of iPlanet Calendar Server are installed.
`local.rfc822header.allow8bit`	`"n"`	Allow (`y`) or not allow (`n`) 8-bit headers in email messages sent by this server.
`local.servergid`	`"icsgroup"`	Unix Group ID (GID) for iPlanet Calendar Server files, e.g., counters and logs.

Table 2-1 Local configuration settings stored in `ics.conf`. *(Continued)*

Parameter	Default Value	Description
`local.serveruid`	`"icsuser"`	Unix User ID (UID) for iPlanet Calendar Server files, e.g., counters and logs.
`local.sitelanguage`	`"en"`	Default language for this instance of iPlanet Calendar Server.
`local.smtp.defaultdomain`	`" "`	Name of the default domain used to lookup an attendee's calendar-id that corresponds to an e-mail address. For example, `jsmith` resolves to `jsmith@siroe.com` if the value for this is setting is `"siroe.com"`.
`local.supportedlanguages`	`"en"`	User languages supported by this instance of iPlanet Calendar Server.
`local.ugldapbasedn`	`" "`	BASEDN for LDAP user preferences. Must be specified and cannot be " ".
`local.ugldaphost`	`"localhost"`	Host name of machine that stores LDAP user preferences.
`local.ugldapicsextendeduserprefs`	`"ceColorSet, ceFontFace, ceFontSizeDelta, ceDateOrder, ceDateSeparator, ceClock, ceDayHead, ceDayTail, ceInterval, ceToolText, ceToolImage, ceDefaultAlarmStart, ceSingleCalendarTZID, ceDefaultAlarmEmail"`	Values for iPlanet Calendar Server 5.0 LDAP schema extensions.
`local.ugldapport`	`"389"`	Port number of the machine that stores LDAP user preferences.
`local.ugldappoolsize`	`"1"`	Minimum number of LDAP client connections that are maintained for LDAP user preferences.
`local.ugldapmaxpool`	`"1024"`	Maximum number of LDAP client connections that are maintained for LDAP user preferences.

Calendar Store Configuration

Table 2-2 describes the calendar store configuration settings stored in `ics.conf`.

Table 2-2 Calendar Store Configuration Settings in ics.conf

Parameter	Default Value	Description
`calstore.anonymous.calid`	`"anonymous"`	Calendar ID (CALID) used for anonymous logins.
`calstore.calendar.default.acl`	`"@@o^a^r^g;@@o^c^wdeic^g;@^a^sf^g"`	Determines the default access control permissions used when a user creates a calendar. The format is specified by a semicolon-separated list of Access Control Entry (ACE) argument strings. For details on the ACE format, see "Access Control," on page 81. To specify Access Control Entries for one or more calendars using the command line utilities, see "Managing Calendars," on page 94 and "cscal," on page 121.
`calstore.calendar.owner.acl`	`"@@o^a^rsf^g;@@o^c^wdeic^g"`	Specifies the owners of a calendar recognized by the Calendar Express user interface. Note: When a user specifies access rights using the Privacy dialog in the Calendar Express user interface, they are applied in reverse order. For example, @@o^a^rsf^g;@@o^c^wdeic^g is applied as @@o^c^wdeic^g;@@o^a^rsf^g.
`calstore.default.timezoneID`	`"America/Los_Angeles"`	Timezone-id to be used when importing files. This server preference should be used as a last resort value only when: • no timezone-id supplied • no calendar timezone-id found • no user timezone-id found An invalid value defaults causes the server to use to the GMT (Greenwich Mean Time) timezone.
`calstore.recurrence.bound`	`"60"`	Maximum number of events that can be created by a recurrence expansion.

Table 2-2 Calendar Store Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`calstore.unqualifiedattendee.fmt1.type`	`"uid"`	Specifies how Calendar Server treats strings, such as `jdoe` or `jdoe:tv`, when performing a directory lookup for attendees of an event. Values can be: • `uid` • `cn` • `gid` • `res` • `mailto` • `cap`
`calstore.unqualifiedattendee.fmt2.type`	`"mailto"`	Specifies how Calendar Server treats strings with an at sign (@), such as `jdoe@foo.com`, when performing a directory lookup for attendees of an event. Values can be: • `uid` • `cn` • `gid` • `res` • `mailto` • `cap`
`calstore.unqualifiedattendee.fmt3.type`	`"cn"`	Specifies how Calendar Server treats strings with a space, such as `john doe`, when performing a directory lookup for attendees of an event. Values can be: • `uid` • `cn` • `gid` • `res` • `cap`

Calendar Log Information Configuration

Table 2-3 describes the calendar log information settings stored in `ics.conf`.

Table 2-3 Calendar Log Settings in ics.conf

Parameter	Default Value	Description
`logfile.admin.logname`	`"admin.log"`	Name of log file for logging administrative tools.
`logfile.buffersize`	`"0"`	Size of log buffers (in bytes).
`logfile.dwp.logname`	`“dwp.log”`	Name of log file for logging Database Wire Protocol related administrative tools.
`logfile.expirytime`	`"604800"`	Number of seconds before log files expire.
`logfile.flushinterval`	`"60"`	Number of seconds between flushing buffers to log files.
`logfile.http.logname`	`"http.log"`	Name of current log file for the `cshttpd` service.
`logfile.http.access.logname`	`"httpd.access"`	Name of current http access log file.
`logfile.logdir`	`"logs"`	Directory location of log files.
`logfile.loglevel`	`"Notice"`	Determines the level of detail the server will log. Each log entry is assigned one of the following levels (starting with the most severe): `Critical, Error, Warning, Notice, Information`, and `Debug`. If you set this preference to `Critical`, the server will log the least amount of detail. If you want the server to log the most amount of detail, specify `Debug.`For example, if you specify `Warning`, only `Critical, Error, and Warning` level log entries are logged.
`logfile.maxlogfiles`	`"10"`	Maximum number of log files in log directory.
`logfile.maxlogfilesize`	`"2097152"`	Maximum size of each log file (in bytes).
`logfile.maxlogsize`	`"20971520"`	Maximum disk space for all log files (in bytes).
`logfile.minfreediskspace`	`"5242880"`	Minimum free disk space that must be available for logging (in bytes). When this value is reached, the server will attempt to free disk space by expiring old log files. All logging will be paused if no space can be freed up.
`logfile.notify.logname`	`"notify.log"`	Name of log file for the `csnotifyd` service.
`logfile.rollovertime`	`"86400"`	Number of seconds before log files are rotated.

Services Configuration

Table 2-4 describes the services configuration settings stored in `ics.conf`.

Table 2-4 Services Configuration Settings in ics.conf

Parameter	Default Value	Description
`service.authcachesize`	`"10000"`	Maximum number of authenticated user Ids (UIDs) and passwords that iPlanet Calendar Server will maintain.
`service.authcachettl`	`"900"`	Number of seconds before a user UID and password are cached.
`service.admin.alarm`	`"yes"`	Enable (`"yes"`) or disable (`"no"`) alarm notifications for administration tools.
`service.admin.calmaster.cred`	`" "`	Password of the USERID specified as the iPlanet Calendar Server administrator. This value is supplied at installation and is required by the installation program.
`service.admin.calmaster.userid`	`"calmaster"`	USERID of the person designated as iPlanet Calendar Server administrator. This value is supplied at installation and is required by the installation program. The default is `calmaster`.
`service.admin.checkpoint`	`"yes"`	If `"yes"`, start the csadmind database checkpoint thread.
`service.admin.dbcachesize`	`"8388608"`	Maximum cache size (in bytes) for Berkeley Database for administration sessions.
`service.admin.deadlock`	`"yes"`	If "`yes`", start the `csadmind` database deadlock detection thread.
`service.admin.diskusage`	`"no"`	If `"yes"`, start the csadmind low disk space monitor thread.

Table 2-4 Services Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`service.admin.enable`	`"yes"`	If `"yes"`, start the `csadmind` service when starting all services and stop `csadmind` when stopping all services.
`service.admin.idletimeout`	`"120"`	Number of seconds before timing out an HTTP connection in `csadmind`.
`service.admin.ldap.enable`	`"yes"`	If `"yes"`, enable LDAP for user authentication of the user specified in `service.admin.calmaster.userid`.
`service.admin.maxsessions`	`"100"`	Maximum number of administration sessions allowed.
`service.admin.maxthreads`	`"10"`	Maximum number of running threads per administration session.
`service.admin.port`	`" "`	Port for administration requests from Calendar Server.
`service.admin.port.enable`	`"no"`	Enable (`"yes"`) or disable (`"no"`) remote administration.
`service.admin.resourcetimeout`	`"900"`	Number of seconds before timing out an administration connection.
`service.admin.serverresponse`	`"no"`	If `"yes"`, start the `csadmind` service response thread.
`service.admin.sessiondir.path`	`" "`	Temporary directory for administration session requests.
`service.admin.sessiontimeout`	`"1800"`	Number of seconds before timing out an HTTP session in `csadmind`.
`service.admin.starttime`	`"30"`	Number of seconds to wait for any calendar service to start.
`service.admin.stoptime`	`"30"`	Number of seconds to wait for any calendar service to stop.
`service.dnsresolveclient`	`"no"`	If `"yes"`, client IP addresses are checked against DNS if allowed HTTP access.

Table 2-4 Services Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`service.http.admins`	"no"	Space separated list of userIDs with administration rights to this Calendar Server.
`service.http.allowadminproxy`	"no"	If `"yes"`, allow login via proxy.
`service.http.allowanonymouslogin`	"yes"	If `"yes"`, allow anonymous (no login) access.
`service.http.calendarhostname`	" "	HTTP host for retrieving HTML documents. To enable users to use a fully qualified hostname to access calendar data, this value must be the fully qualified host name (including the machine name, DNS domain and suffix) of the machine on which the Calendar Server is running, such as `my_ics5@siroe.com`. If not specified, the local HTTP host is used.
`service.http.cookies`	“yes”	Tells the server to whether or to support cookies (yes/no). It must be set to `”yes”` to enable single sign-on.
`service.http.dbcachesize`	"8388608”	Maximum cache size of Berkeley DB for HTTP sessions.
`service.http.domainallowed`	" "	If specified and not `" "`, filter to allow access based on TCP domains. For example, `"ALL: LOCAL.siroe.com"` would allow local HTTP access to anyone in the `siroe.com` domain. Multiple filters are separated by CR-LF (line feed).

Table 2-4 Services Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
service.http.domainnotallowed	" "	If specified and not " ", filter to not allow access based on TCP domains. For example, "ALL: LOCAL.siroe.com" would deny HTTP access to anyone in the siroe.com domain. Multiple filters must be separated by CR-LF (line-feed).
service.http.attachdir.path	"."	Directory location relative to local.queuedir (or an absolute path if specified) where imported files are temporarily stored.
service.http.ipsecurity	"yes"	If "yes", all requests that reference an existing session are verified as originating from the same IP address.
service.http.enable	"yes"	If "yes", start the cshttpd service when starting all services and stop cshttpd when stopping all services.
service.http.idletimeout	"120"	Number of seconds before timing out an HTTP connection.
service.http.ldap.enable	"yes"	If "yes", LDAP connections for authentication and user preferences are created and maintained.
service.http.logaccess	"no"	If "yes", HTTP connections to server are fully logged.
service.http.maxsessions	"5000"	Maximum number of WCAP sessions.
service.http.maxthreads	"1000"	Maximum number of threads to service HTTP requests.
service.http.numprocesses	"1"	Maximum number of processes to service HTTP requests.
service.http.port	"80"	Port for HTTP requests from Calendar Server users.

Table 2-4 Services Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
service.http.proxydomainallowed	" "	If specified and not " ", filter for allowing proxy login based on TCP domains. Same syntax as service.http.domainallowed.
service.http.resourcetimeout	"900"	Number of seconds before timing out an HTTP session.
service.http.sessiondir.path	"http"	Temporary directory for HTTP sessions.
service.http.sessiontimeout	"1800"	Number of seconds before timing out WCAP session.
service.http.sourceurl	" "	Directory relative to executable where all URL references to files are stored.
service.http.uidir.path	"html"	Directory that contains the default calendar client. If allowing only WCAP access, set to "".
service.ldapmemcache	"no"	If "yes", use cache in LDAP SDK.
service.ldapmemcachettl	"30"	If service.ldapmemcache 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.
service.ldapmemcachesize	"131072"	If service.ldapmemcache 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.
service.listenaddr	"INADDR_ANY"	(UNIX only) Specifies the TCP address that http services will listen on for client requests. “INADDR_ANY” indicates any address.

Table 2-4 Services Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`service.plaintextloginpause`	`"0"`	Number of seconds to delay after successfully authenticating a user using plain text passwords.
`service.wcap.anonymous.allowpubliccalendarwrite`	`"yes"`	If `"yes"`, allow anonymous users to write to publicly writable calendars.
`service.wcap.allowcreatecalendars`	`"yes"`	If `"yes"`, allow calendars to be created.
`service.wcap.allowdeletecalendars`	`"yes"`	If `"yes"`, allow calendars to be deleted.
`service.wcap.allowchangepassword`	`"no"`	If `"yes"`, allow users to change their passwords via this server.
`service.wcap.allowpublicwritablecalendars`	`"yes"`	If `"yes"`, allow users to have publicly writable calendars.
`service.wcap.allowsetprefs.cn`	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `"cn"` (LDAP user's common name).
`service.wcap.allowsetprefs.givenname`	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `"givenname"` (LDAP user's given name).
`service.wcap.allowsetprefs.mail`	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `"mail"` (user's e-mail address).
`service.wcap.allowsetprefs.preferredlanguage`	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `"preferredlanguage"` (LDAP user's preferred language).
`service.wcap.allowsetprefs.sn`	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `"sn"` (LDAP user's surname).

Table 2-4 Services Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`service.wcap.allowsetprefs.nswccalid`	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference "`nswccalid`", which is the user's default calendar identifier
`service.wcap.login.calendar.publicread`	`"no"`	If `"yes"`, default user calendars are initially set to public read / private write. If `no`, default user calendars are initially set to private read / private write.
`service.wcap.validateowners`	`"no"`	If `"yes"`, the server must validate that each owner of a calendar exists in the directory (through LDAP or a CSAPI compatible user directory mechanism).

Alarm Notification Configuration

Table 2-5 shows the Alarm Notification Server (ENS) configuration settings stored in `ics.conf`.

Table 2-5 Alarm Notification Configuration Settings in `ics.conf`

Parameter	Default Value	Description
`alarm.diskavail.msgalarmdescription`	`"percentage calendar partition diskspace available"`	Warning text sent when the server encounters insufficient disk space.
`alarm.diskavail.msgalarmstatinterval`	`"3600"`	Number of seconds the server waits between each time it checks for available disk space.
`alarm.diskavail.msgalarmthreshold`	`"10"`	Percentage of available disk space that triggers the server to send a warning message.

Table 2-5 Alarm Notification Configuration Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`alarm.diskavail.msgalarmthresholddirection`	`"-1"`	If `alarm.diskavail.msgalarmthreshold` is allowed to be above or below the acceptable percentage, where: • `-1` is below • `1` is above
`alarm.diskavail.msgalarmwarninginterval`	`"24"`	Number of hours between sending warning messages about insufficient disk space sent out.
`alarm.msgalarmnoticehost`	`"localhost"`	The host name of the SMTP server used to send server alarms.
`alarm.msgalarmnoticeport`	`"25"`	The SMTP port used to send server alarms.
`alarm.msgalarmnoticercpt`	`"Postmaster @localhost"`	The email address to whom server alarms sent.
`alarm.msgalarmnoticesender`	`"Postmaster @localhost"`	The email address used as the sender when the server sends alarms.
`alarm.msgalarmnoticetemplate`	`"Postmaster @localhost"`	The default format used to send email alarms: `"From: %s\nTo: %s\nSubject: ALARM: %s of \"%s\" is %u\n\n%s\n"`

Calendar Store Configuration

Table 2-6 describes the store configuration settings stored in `ics.conf`.

Table 2-6 Store Configuration Settings in ics.con

Parameter	Default Value	Description
`store.partition.primary.path`	`"."`	Location of primary disk partition where calendar information is stored.

Calendar Locate Mechanism Configuration

Table 2-7 describes the calendar locate mechanism service settings stored in `ics.conf`.

Table 2-7 Calendar Locate Mechanism Settings in `ics.conf`

Parameter	Default Value	Description
`csapi.plugin.calendarlookup`	`"n"`	Enable (`"y"`) or disable (`"n"`) calendar locate mechanism plugins.
`csapi.plugin.calendarlookup.name`	`" "`	Specifies the name of a specific calendar locate mechanism plugin to load. If this value is blank (`" "`), the server loads all plugins. When using the Database Wire Protocol service, an asterisk (`*`) tells the server to load all calendar locate mechanism plugins.
`caldb.cld.type`	`"local"`	Specifies the type of calendar locate mechanism plugin to use: •`"local"` specifies that all calendars are stored on the local machine on which the Calendar Server is running and no plugin is loaded. •`"algorithmic"` loads a plugin that uses a regular expression to determine the physical machine on which a specified calendar ID is stored. The expression is specified with the setting: `caldb.cld.server.`*`[hostname]`*`.regexpr` •`"directory"` loads a plugin that uses an LDAP directory schema entry to determine the physical machine on which a specified calendar ID is stored.
`caldb.dwp.connthreshold`	`"1"`	Maximum number of backlogged requests before the server obtains a new network connection.
`caldb.dwp.forceremote`	`"n"`	Forces requests for local calendar IDs to use remote the Database Wire Protocol service. (Used only to debug local machine.)

Table 2-7 Calendar Locate Mechanism Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`caldb.dwp.initconns`	`"2"`	Initial number of connections for the Database Wire Protocol service client to make to each Database Wire Protocol service host.
`caldb.dwp.initthreads`	`"2"`	Initial number of threads for handling Database Wire Protocol service requests.
`caldb.dwp.maxcons`	`"1000"`	Maximum number of connections allowed to a server using the Database Wire Protocol service.
`caldb.dwp.maxthreads`	`"1000"`	Maximum number of threads allowed to a server using the Database Wire Protocol service.
`caldb.dwp.md5`	`"n"`	Specifies if the server performs MD5 (Message Digest 5) one-way hash checking of all Database Wire Protocol service requests. (One-way hash functions are used to create digital signatures for message authentication.) • `"n"` disables MD5 hash checking. • `"y"` enables MD5 hash checking.
`caldb.dwp.server.`*`hostname`*`.ip`	`" "`	Specifies the IP address of the server using the Database Wire Protocol service at the specified machine’s hostname.
`caldb.dwp.stacksize`	`"65536"`	Stack size for Database Wire Protocol service threads.
`caldb.localdb`	`"y"`	Specifies if a local database exists where: • `"y"` disables DWP. • `"n"` tells the server to use remote DWP only and to access the host name set for `caldb.localdb.host`, if specified.

Table 2-7 Calendar Locate Mechanism Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`caldb.localdb.host`	`"localhost"`	If the local machine supports the Database Wire Protocol service (if `caldb.localdb` is `"n"`), then access this host name (of the local machine on which the Database Wire Protocol service is running) for quicker access. When this preference is set, requests for calendar IDs stored on the local machine are retrieved locally rather than remotely.
`csapi.plugin.authentication`	`"n"`	If `"y"`, load only the plugin specified in `csapi.plugin.authentication.name` or if not specified, load all authentication class plugins in alphabetical order. For authentication, use each of these plugins in alphabetical order.
`csapi.plugin.authentication.name`	`" "`	If `csapi.plugin.loadall` is `"n"` and `csapi.plugin.authentication` is `"y"`, only load this specific plugin. If not specified or `" "`, load all authentication class plugins.
`service.dwp.maxthreads`	`"1000"`	Maximum number of concurrently running Database Wire Protocol service threads.
`service.dwp.numprocesses`	`"1"`	Maximum number of concurrently running Database Wire Protocol service processes.
`service.dwp.enable`	`"no"`	If `"yes"`, start the `csdwpd` service when starting all services and stop `csdwpd` when stopping all services
`service.dwp.idletimeout`	`"86400"`	Amount of time (in seconds) before closing the Database Wire Protocol service persistent connections that are idle.
`service.dwp.port`	`"9779"`	Port number that the Database Wire Protocol service listens to.

Table 2-7 Calendar Locate Mechanism Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`service.dwp.ldap.enable`	`"yes"`	Enable (`"yes"`) or disable (`"no"`) LDAP for remote user authentication for the Database Wire Protocol service.

Single Sign-on Configuration

Table 2-8 describes the single sign-on settings stored in `ics.conf`.

Table 2-8 Single Sign-on Configuration Settings in ics.conf

Parameter	Default Value	Description
`sso.appid`	`"ics50"`	Unique application ID for this Calendar Server. For example: `sso.appid="94043"`
`sso.appprefix`	`"ssogrp1"`	The application prefix for the Calendar Server. Each trusted application must also have a unique application ID. This value is used to find cookies generated by other trusted applications. The application prefix must end with a hyphen (-). For example: `sso.appprefix="d98-"`
`sso.cookiedomain`	`" "`	Tells the browser to send a cookie only to servers in the domain specified here. The value must begin with a period (.), for example: `".siroe.com"`
`sso.enable`	`"1"`	Enables or disables single sign-on: • `"1"` enables the server to perform single sign-on functions • `"0"` disables single sign-on functions. **Note:** If this setting is missing from `ics.conf`, the calendar ignores single sign-on functions.
`sso.singlesignoff`	`"true"`	If set to `"true"`, the server removes all single sign-on cookies for the user that match the value for `sso.appprefix` when the user logs out. If `"false"` the server removes only its single sign-on user cookie.

Table 2-8 Single Sign-on Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`sso.userdomain`	`" "`	Sets the domain used as part of the user's single sign-on authentication.

Group Scheduling Configuration

Table 2-9 describes the group scheduling settings stored in `ics.conf`.

Table 2-9 Group Scheduling Configuration Settings in ics.conf

Parameter	Default Value	Description
`calstore.calendar.default.acl`	`"@@o^a^r^g;@@o^c^wdeic^g;@^a^sf^g"`	Determines the default access control permissions used when a user creates a calendar. The format is specified by a semicolon-separated list of Access Control Entry (ACE) argument strings. For details on the ACE format, see "Access Control," on page 81. To specify Access Control Entries for one or more calendars using the command line utilities, see "Managing Calendars," on page 94 and "cscal," on page 121.
`calstore.calendar.owner.acl`	`"@@o^a^rsf^g;@@o^c^wdeic^g"`	Specifies the owners of a calendar recognized by the Calendar Express user interface. Note: When a user specifies access rights using the Privacy dialog in the Calendar Express user interface, they are applied in reverse order. For example, @@o^a^rsf^g;@@o^c^wdeic^g is applied as @@o^c^wdeic^g;@@o^a^rsf^g.

Table 2-9 Group Scheduling Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`resource.default.acl`	`@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g`	Determines the default access control permissions used when a calendar associated with a resource (such as a conference room) is created. The format is specified by a semicolon-separated list of Access Control Entry (ACE) argument strings. For details on the ACE format, see “Access Control,” on page 81. To specify Access Control Entries for one or more calendars using the command line utilities, see “Managing Calendars,” on page 94 and “cscal,” on page 121.
`gse.autorefreshreplystatus`	`"yes"`	Specifies if the auto refresh feature is enabled or disabled. If auto refresh is enabled, after an attendee replies to an event organizer, that attendee’s reply status is automatically propagated to other attendees for that scheduled event. • “`yes"` enables auto refresh. • `"no"` disables auto refresh.
`gse.belowthresholdtimeout`	`"3"`	Specifies (in seconds) how long to wait before the server scans the schedule queue for incoming jobs. If there are more jobs in the queue than the maximum threads allocated, the last thread will always scan the job queue again. Therefore, this setting only takes effect when the number of jobs is below the maximum threads allocated. Increasing this number reduces the frequency the server scans the job queue and improves overall performance.
`gse.maxthreads`	`"10"`	Specifies the maximum number of concurrent threads the server uses to process the schedule queue. Each thread processes one job in the queue.
`gse.retryexpiredinterval`	`“86400”`	Specifies (in seconds) the maximum length of time the server will retry to complete a group scheduling job. If the time exceeds the maximum length of time specified, the server treats the job as a retry expired condition and reports the error. Note that the default of 86400 seconds equals one day.

Table 2-9 Group Scheduling Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
`gse.retryinterval`	“300”	Specifies (in seconds) how often the server will retry a previous failing job. The server retries a failing job only when a network error is encountered. The server treats most such errors, however, as fatal errors and does not consider them as retries.
`gse.stacksize`	“65535”	Specifies the maximum stack size (in bytes) of a group scheduling thread.
`resource.allow.doublebook`	"no"	Determines if a calendar that belongs to a resource (such as a conference room or audio visual equipment) can have more than one event scheduled for the same time slot. • `"no"` prevents double booking. • `"yes"` allows double booking.
`user.allow.doublebook`	"yes"	Determines if a user’s calendar can have more than one event scheduled for the same time slot. • `"no"` prevents double booking. • `"yes"` allows double booking.

Database Configuration

Table 2-10 shows the iPlanet Calendar Server database configuration settings stored in `ics.conf`.

Table 2-10 Database Configuration Settings in `ics.conf`

Parameter	Default Value	Description
`caldb.berkeleydb.checkpointinterval`	"60"	Number of seconds between checkpointing database transactions.
`caldb.berkeleydb.circularlogging`	"yes"	If `"yes"` remove database checkpoint files after their transactions are synchronized.
`caldb.berkeleydb.deadlockinterval`	"100”	Number of milliseconds between checking for database deadlocks that need to be broken.

Table 2-10 Database Configuration Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`caldb.berkeleydb.homedir.path`	`"."`	Directory (relative to the location of the program executable files or an absolute path if specified) where database event, task, and alarm files are kept. The default value of `"."` specifies that these files are stored in the *`server`*`-root/cal/bin` directory.
`caldb.berkeleydb.logfilesizemb`	`"10"`	Maximum megabytes for a database checkpoint file.
`caldb.berkeleydb.maxthreads`	`"10000"`	Maximum number of threads that database environment must be prepared to accommodate.
`caldb.berkeleydb.mempoolsizemb`	`"4"`	Megabytes of shared memory for database environment.
`caldb.calmaster`	`" "`	Email for user/alias that is responsible for administering the database.
`caldb.counters`	`"yes"`	If `"yes"`, data base statistics (reads, writes, deletes) will be counted.
`caldb.counters.maxinstances`	`"100"`	Maximum number of calendars that can have counters. A calendar is enabled for counters using the `cscal` command line utility.

`caldb.smtpmsgfmtdir`	`"en"`	Specifies the directory under *`server-root/`*`cal/bin/config` that contains the localized version of the files used to format email notifications. For example:
		`"en"` specfies that directory location of the English localized version is: *`server-root`*`/cal/bin/config/en`
		`"fr"` specifies the directory location of the French localized version is: *`server-root`*`/cal/bin/config/fr`

`caldb.smtpport`	`"25"`	Port for SMTP host.

Calendar Server API Configuration

Table 2-11 shows the Calendar Server API (CSAPI) configuration settings stored in `ics.conf`.

Table 2-11 CSAPI Configuration Settings in ics.conf

Parameter	Default Value	Description
csapi.plugin.authentication	"n"	If ("y"), load only the plugin specified in csapi.plugin.authentication.name.
csapi.plugin.accesscontrol	"n"	Enable ("y") or disable ("n") Access Control plugin.
csapi.plugin.authentication	"n"	If "y", load only the plugin specified in csapi.plugin.authentication.name or if not specified, load all authentication class plugins in alphabetical order. For authentication, use each of these plugins in alphabetical order.
csapi.plugin.authentication.name	" "	If csapi.plugin.loadall is "n" and csapi.plugin.authentication is "y", only load this specific plugin. If not specified or " ", load all authentication class plugins.
csapi.plugin.database	"y"	If "y", load only the plugin specified in csapi.plugin.database.name or if not specified, load all database plugins in alphabetical order.
csapi.plugin.database.name	"”	If csapi.plugin.loadall is "n" and csapi.plugin.database is "y", only load this specific plugin. If not specified or " ", load all database plugins in alphabetical order.
csapi.plugin.datatranslator	"y"	If "y", load only the plugin specified in csapi.plugin.datatranslator.name or if not specified, load all data translator class plugins in alphabetical order. For data translation, use each of these plugins in alphabetical order.
csapi.plugin.datatranslator.name	"cs_datatranslatorcsv10”	If csapi.plugin.loadall is "n" and csapi.plugin.datatranslator is "y", this parameter is used. If not specified or " ", load all data translator class plugins. Otherwise, only load this specific plugin.
csapi.plugin.dbtranslator	"y"	Enable ("y") or disable ("n") database-to-output format plugins.

Table 2-11 CSAPI Configuration Settings in ics.conf *(Continued)*

Parameter	Default Value	Description
csapi.plugin.dbtranslator.name	"*"	If the setting csapi.plugin.dbtranslator is "y", then either: • load all the database-to-output format plugins if this value is "*". • load only this specific plugin if this value is a library name. If csapi.plugin.dbtranslator is "n", this setting is ignored.
csapi.plugin.loadall	"n"	If "y", load all plugins found in the plugins directory. For NT, these plugins have a .dll file name extension. For IRIX, these plugins have an .sl extension. For all other UNIX systems, these plugins have an .so extension. If "n", only load the specific class of plugins flagged by their respective parameters. For example, set csapi.plugin.authentication to yes to load authentication class plugins.
csapi.plugin.userprefs	"n"	If "y", load only the plugin specified in csapi.plugin.userprefs.name or if not specified, load all user preferences class plugins in alphabetical order. For user preferences, use each of these plugins in alphabetical order.
csapi.plugin.userprefs.name	" "	If csapi.plugin.loadall is "n" and csapi.plugin.userprefs is "y", this parameter is used. If not specified or " ", load all user preferences class plugins. Otherwise, only load this specific plugin.

Event Notification Server Configuration

iPlanet Calendar Server, when configured to do so, uses an external generic service called the Event Notification Server (ENS) which accepts reports of server-level events that can be categorized into specific areas of interest and notifies other servers that have registered interest in certain categories of events. iPlanet Calendar Server uses the Event Notification Server to send and receive alarm notifications which include the creation, deletion, or modification of calendar events and tasks as well as general operational warning and error messages.

Table 2-12 shows the Event Notification Server (ENS) configuration settings stored in `ics.conf`.

Table 2-12 Event Notification Server Configuration Settings in `ics.conf`

Parameter	Default Value	Description
`alarm.diskavail.msgalarmdescription`	`"percentage calendar partition diskspace available"`	Warning text sent when the server encounters insufficient disk space.
`alarm.diskavail.msgalarmstatinterval`	`"3600"`	Number of seconds the server waits between each time it checks for available disk space.
`alarm.diskavail.msgalarmthreshold`	`"10"`	Percentage of available disk space that triggers the server to send a warning message.
`alarm.diskavail.msgalarmthresholddirection`	`"-1"`	If `alarm.diskavail.msgalarmthreshold` is allowed to be above or below the acceptable percentage, where: • `-1` is below • `1` is above
`alarm.diskavail.msgalarmwarninginterval`	`"24"`	Number of hours between sending warning messages about insufficient disk space sent out.
`alarm.msgalarmnoticehost`	`"localhost"`	The host name of the SMTP server used to send server alarms.
`alarm.msgalarmnoticeport`	`"25"`	The SMTP port used to send server alarms.
`alarm.msgalarmnoticercpt`	`"Postmaster @localhost"`	The email address to whom server alarms sent.

Table 2-12 Event Notification Server Configuration Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`alarm.msgalarmnoticesender`	`"Postmaster @localhost"`	The email address used as the sender when the server sends alarms.
`alarm.msgalarmnoticetemplate`	`"Postmaster @localhost"`	The default format used to send email alarms: `"From: %s\nTo: %s\nSubject: ALARM: %s of \"%s\" is %u\n\n%s\n"`
`service.ens.enable`	`"yes"`	If `"yes"`, start the `enpd` service when starting all services and stop `enpd` when stopping all services.
`service.ens.host`	`"localhost"`	The host name of the machine on which the Event Notification Server is running.
`service.ens.port`	`"7997"`	The port number of the machine on which the Event Notification Server is running.
`service.ens.library`	`"xenp"`	The name of the Event Notification Server plugin.
`service.notify.enable`	`"yes"`	If `"yes"`, start the `csnotifyd` service when starting all services and stop `csnotifyd` when stopping all services.
`service.notify.maxretrytime`	`"-1"`	How many times `csnotifyd` will consecutively retry and fail to contact the Event Notification Server. `"-1"` causes the alarm thread to retry indefinitely.
`service.notify.retryinterval`	`"3"`	Number (in seconds) that `csnotifyd` waits before attempting to re-contact the Event Notification Server after a connection failure.

Table 2-12 Event Notification Server Configuration Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`service.notify.startupretrytime`	`"0"`	Total number of seconds the Calendar Server keeps trying to contact the Event Notification Server before it stops. This setting is similar to `caldb.serveralarms.maxretrytime` except that it applies only when the alarm thread is first starting. Once the alarm thread has successfully started, `caldb.serveralarms.maxretrytime` is used. `"0"` tells the alarm thread to exit immediately if it fails to connnect to the Event Notification Server at startup.
`ens.initthreads`	`"0"`	Initial number of threads for the Event Notification Server.
`ens.maxthreads`	`"10"`	Maximum number of threads for the Event Notification Server.
`ens.stacksize`	`"65536"`	Stack frame size for Event Notification Server threads.
`caldb.berkeleydb.alarmretrytime`	`"300”`	Retry time in seconds after a recoverable alarm delivery error.
`caldb.berkeleydb.ensmsg.createcal`	`"no"`	If `"yes"`, create an Event Notification Service message when a calendar is created using the following format: `enp://ics/createcal?calid=cal`
`caldb.berkeleydb.ensmsg.modifycal`	`"no"`	If `"yes"`, create an Event Notification Service message when a calendar is modified using the following format: `enp://ics/modifycal?calid=cal`
`caldb.berkeleydb.ensmsg.deletecal`	`"no"`	If `"yes"`, create an Event Notification Service message when a calendar is deleted using the following format: `enp://ics/deletecal?calid=cal`
`caldb.berkeleydb.ensmsg.createevent`	`"no"`	If `"yes"`, create an Event Notification Service message when an event is created?

Table 2-12 Event Notification Server Configuration Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`caldb.berkeleydb.ensmsg.deleteevent`	`"no"`	If `"yes"`, create an ENS message when an event is deleted?
`caldb.berkeleydb.ensmsg.modifyevent`	`"no"`	If `"yes"`, create an ENS message when an event is modified?
`caldb.berkeleydb.ensmsg.createtodo`	`"no"`	If `"yes"`, create an Event Notification Service message when a task is created using the following format: `enp://ics/createtodo?uid=uid&rid=rid`
`caldb.berkeleydb.ensmsg.modifytodo`	`"no"`	If `"yes"`, create an Event Notification Service message when a task is modified using the following format: `enp://ics/modifytodo?uid=uid&rid=rid`
`caldb.berkeleydb.ensmsg.deletetodo`	`"no"`	If `"yes"`, create an Event Notification Service message when a task is deleted using the following format: `enp://ics/deletetodo?uid=uid&rid=rid`
`caldb.berkeleydb.ensmsg.qsize`	`"10000"`	Initial size of the in-memory Event Notification Server (ENS) message queue. This queue stores all ENS messages other than alarm reminders.
`caldb.berkeleydb.ensmsg.schedreq`	`"no"`	If `"yes"`, create an Event Notification Service message when a scheduling request is written to the calendar is deleted using the following format: `enp://ics/schedreq?calid=cal&method=method&type={event|todo}&uid=uid&rid=rid`
`caldb.serveralarms`	`"yes"`	If `"yes"`, alarm emails will be sent.
`caldb.serveralarms.acktimeout`	`"30"`	Specifies the number of seconds the Event Notification Server's alarm thread waits for an acknowledgment from `csnotifyd` after publishing an alarm notification. If the time-out expires, the alarm thread assumes the alarm notification is no longer processing and publishes the alarm notification again.

Table 2-12 Event Notification Server Configuration Settings in `ics.conf` *(Continued)*

Parameter	Default Value	Description
`caldb.serveralarms.dispatchtype`	`"ens"`	Specifies the dispatch type for Calendar Server alarms: • `"ens"` tells the server to use the external Event Notification Server to send and recieve alarms. • `"smtp"` tells the the server to send alarms as standard SMTP messsages and to bypass the Event Notification Server.
`caldb.serveralarms.initthreads`	`"10"`	Initial number of threads for the Event Notification Server.
`caldb.serveralarms.maxretrytime`	`"-1"`	How many times the alarm thread will consecutively retry and fail to contact the Event Notification Server. `"-1"` causes the alarm thread to retry indefinitely.
`caldb.serveralarms.maxthreads`	`"10"`	Maximum number of threads for the Event Notification Server.
`caldb.serveralarms.retryinterval`	`"5"`	Number (in seconds) that the alarm thread (in `csadmind`) waits before attempting to recontact the Event Notification Server.
`caldb.serveralarms.stacksize`	`"65536"`	Stack frame size for Event Notification Server threads.
`caldb.serveralarms.startupretrytime`	`"0"`	Total number of seconds the Calendar Server keeps trying to contact the Event Notification Server before failing. This setting is similar to the setting `caldb.serveralarms.maxretrytime` except that it applies only when the alarm thread is first starting. Once the alarm thread has started successfully, `caldb.serveralarms.maxretrytime` is used. `"0"` tells the alarm thread to exit immediately if it fails to connect to the Event Notification Server at startup.
`caldb.smtphost`	`"localhost"`	Send alarm emails to this SMTP host.

User Interface Configuration

Table 2-13 describes the user interface configuration in `ics.conf`.

Table 2-13 User Interface Configuration Settings in ics.con

Parameter	Default Value	Description
`ui.config.file`	`" "`	iPlanet Calendar Server can read an optional (xml based) configuration file at startup that can hide parts of the user interface. iPlanet Calendar Server 5.0 allows only one configuration file and the value of this setting determines the name of the configuration file to use. iPlanet Calendar Server looks for the file in the data directory where the user interface `xml` and `xslt` files are: `server-root/cal/bin/data/..` iPlanet Calendar Server 5.0 provides the following files that provide customized versions of the user interface: • `nogroup_config.xml` — disables group scheduling • `ui_config.xml` — default user interface

Notification Messages

iPlanet Calendar Server sends several types of email messages described in Table 2-14. 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 2-14 shows the iPlanet Calendar Server mail settings stored in `ics.conf`.

Table 2-14 iPlanet Calendar Server mail formats

Message Type	Parameter	Format File (default)	Description	Recipients
Event Publish	`calmail.imipeventpublish.fname`	`"mail_eventpublish.fmt"`	Announces an event or a change to an existing event	Those listed in Notification
Event Cancel	`calmail.imipeventcancel.fname`	`"mail_eventcancel.fmt"`	Announces an event cancellation	Those listed in Notification
Reply to Event	`calmail.imipeventreply.fname`	`"mail_eventreply.fmt"`	Replies to an event notification.	Those listed in Notification
Request Event	`calmail.imipeventrequest.fname`	`"mail_eventrequest.fmt"`	Subscribes to an event notification.	Those listed in Notification
Event Alarm	`calmail.eventreminder.fname`	`"mail_eventreminder.fmt"`	Reminder for an upcoming event	Those listed in Reminder
Task Publish	`calmail.imiptodopublish.fname`	`"mail_todopublish.fmt"`	Announces a task or a change to an existing task	Those listed in Notification
Task Cancel	`calmail.imiptodocancel.fname`	`"mail_todocancel.fmt"`	Announces a task cancellation	Those listed in Notification
Reply to Task	`calmail.imiptodoreply.fname`	`"mail_todoreply.fmt"`	Replies to a task notification	Those listed in Notification.
Todo Request	`calmail.imiptodorequest.fname`	`"mail_todorequest.fmt"`	Subscribes to a todo notification.	Those listed in Notification
Task Alarm	`calmail.todoreminder.fname`	`"mail_todoreminder.fmt"`	Reminder for an upcoming task	Those listed in Reminder

iPlanet 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 2-15 shows the special character sequences for iPlanet Calendar Server event notifications.

Table 2-15 Special Character Sequences for Event Notifications

Format Code	Meaning
%0	Start time in localized format
%1	End time in localized format
%A	exdates in ICalendar format
%a	rdates in ICalendar format
%B	Start time (also see %Z)
%b	Output the start time and end time in ICalendar format. If the start time has the parameter value=date, 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)

Table 2-15 Special Character Sequences for Event Notifications *(Continued)*

Format Code	Meaning
`%e`	Exception rules in ICalendar format
`%F`	Event description - folded line / ICalendar format (also see `%d`)
`%G`	The 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 `mailto:url`
`%k`	Alarm count
`%L`	Location
`%l`	Recurrence rules in ICalendar format
`%M`	Modify time
`%N`	New line
`%n`	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 (Unique Event Identifier)
`%u`	URL to the event
`%Z`	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.)
`%%`	Displays the percent (%) character
`% (sub-format code)`	Specifies a subformat for the data identified by code. (For details, see "Date Sub-Formatting" on page 67.)

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 70). 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 print a line 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 2-16 shows the special character sequences for iPlanet Calendar Server task notifications.

Table 2-16 Special Character Sequences for Tasks Notifications

Format Code	Meaning
%A	exdates in ICalendar format
%a	rdates in ICalendar format
%B	start time (also see **%Z**)
%C	create time
%c	task class
%D	due date/time.

Table 2-16 Special Character Sequences for Tasks Notifications *(Continued)*

Format Code	Meaning
`%d`	task description. (also see **%F**)
`%E`	due date/time in IMIP format
`%e`	exception rules in ICalendar format
`%F`	task description - folded line / ICalendar format (also see **%d**)
`%G`	this task's geographic location, the latitude and longitude.
`%g`	organizer's email address (cannot guarantee the authenticity of this value)
`%K`	organizer's email in the form of a `mailto:URL`
`%k`	alarm count
`%L`	the location
`%l`	recurrence rules in ICalendar 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
`%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 67)

Special Character Sequences for Dates

Table 2-17 shows the special character sequences for date formatting in iPlanet Calendar Server notifications.

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

Table 2-17 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)
`%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

Table 2-17 Special Character Sequences for Dates *(Continued)*

Format Code	Meaning
`%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 67.

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@siroe.com (James Smith)
Date: Wed, 15 Nov 1999 19:13:49
To: jsmith@siroe.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
~~--%b--
```

Counters

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

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

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

The first part of a counter's name identifies the counter object used with the csstats command line utility (for more information on the command line utilities, see Chapter 3, "Administering iPlanet Calendar Server and Appendix A, "Command Line Utilities"):

- A parameter with spaces must be enclosed in single quotes (" ").
- A comment line must have an exclamation point (!) as the first character of the line. Comment lines are for informational purposes only.

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

Table 2-19 shows the alarm usage counter objects stored in `counter.conf`.

Table 2-18 Alarm Counters in `counter.conf`

Name	Type	Size	Description
`alarm.high`	GAUGE	4	Highest ever recorded value.
`alarm.low`	GAUGE	4	lowest ever recorded value.
`alarm.current`	GAUGE	4	Current monitored valued.
`alarm.warningstate`	GAUGE	4	warning state (yes(1) or no(0).
`alarm.countoverthreshold`	COUNTER	4	number of times crossing threshold.
`alarm.countwarningsent`	COUNTER	4	Number of warnings sent.
`alarm.timelastset.desc`	TIME	4	The last time current value was set.
`alarm.timelastwarning`	TIME	4	The last time warning was sent"
`alarm.timereset`	TIME	4	The last time reset was performed.
`alarm.timestatechanged.desc`	TIME	4	The last time alarm state changed.

Table 2-19 shows the disk usage counter objects stored in `counter.conf`.

Table 2-19 Disk Usage Counters in `counter.conf`

Name	Type	Size	Description
`diskusage.availSpace`	GAUGE	54	Total space available in the disk partition.
`diskusage.lastStatTime`	TIME	4	The last time statistic was taken.
`diskusage.calPartitionPath`	STRING	512	Calendar partition path.
`diskusage.percentAvail`	GAUGE	4	Disk partition space available percentage.
`diskusage.totalSpace`	GAUGE	5	Total space in the disk partition.

Table 2-20 shows the HTTP counter objects (`httpstat`) stored in `counter.conf`.

Table 2-20 httpstat Counters in `counter.conf`

Name	Type	Size	Description
`httpstat.avgConnectionTime`	GAUGE	4	Average connection response time.

Table 2-20 httpstat Counters in `counter.conf` *(Continued)*

Name	Type	Size	Description
`httpstat.currentStartTime`	TIME	4	When iPlanet Calendar Server was started.
`httpstat.lastConnectionTime`	TIME	4	Last time new client connection was accepted
`httpstat.maxConnections`	COUNTER	4	Maximum number of concurrent connections served.
`httpstat.numConnections`	COUNTER	4	Total number of connections served.
`httpstat.numCurrentConnections`	GAUGE	4	Current number of active connections.
`httpstat.numFailedConnections`	COUNTER	4	Total number of failed connections served.
`httpstat.numGoodLogins.desc`	COUNTER	4	Number of successful logins served by the current HTTP server.
`httpstat.numFailedLogins`	COUNTER	4	Number of failed logins served by the current HTTP server

Table 2-21 shows the disk usage counter objects stored in `counter.conf`.

Table 2-21 Group Scheduling Counters in `counter.conf`

Name	Type	Size	Description
`gsestat.lastWakeUpTime`	TIME	4	Last time GSE wakes up and process job.
`gsestat.lastJobProcessedTime`	TIME	4	last time GSE processes a job.
`gsestat.numJobsProcessed`	COUNTER	4	Total number of jobs GSE processed.
`gsestat.numActiveWorkerThreads`	COUNTER	4	Total number of active Worker Threads.

Table 2-22 shows the authentication counter objects (`authstat`) stored in `counter.conf`

Table 2-22 Authentication Counters in `counter.conf`

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

Table 2-23 shows the WCAP counter object (`wcapstat`) stored in `counter.conf`.

Table 2-23 wcapstat counters in `counter.conf`

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

Table 2-24 shows the database counter object (`dbstat`) stored in `counter.conf`.

Table 2-24 Database Counters in `counter.conf`

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

Table 2-25 shows the server response counter object stored in `counter.conf`.

Table 2-25 Server Response Counters in `counter.conf`

Name	Type	Size	Scale	Description
`serverresponse.lastStatTime`	TIME	4		The last time statistic was taken.
`serverresponse.responseTime`	GAUGE	4	2	Server response time in milliseconds.

Table 2-26 shows the session status counter object stored in `counter.conf`.

Table 2-26 Sessions Status Counters in `counter.conf`

Name	Type	Size	Scale	Description
`sessstat.maxSessions.desc`	COUNTER	4	4	Maximum number of HTTP sessions served.
`sessstat.numCurrentSessions`	GAUGE	4	2	Current number of http sessions.

Chapter 3

Administering iPlanet Calendar Server

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

- Administration Tools
- Starting and Stopping the Server
- Access Control
- Calendar Locate Mechanism
- Using the Command Line Utilities
- Backup and Restore Procedures

Administration Tools

iPlanet Calendar Server provides the following command line utilities for server administration to:

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

These command line utilities should be executed only by the user who has administration rights to the system on which the Calendar Server is running. This user is typically the identity that the server will run as which was designated during installation, such as `icsuser` on Unix or the user with full administration privileges on Windows NT, such as administrator. Some command line utilities require the presence of a user designated as the iPlanet Calendar Server Administrator (such as `calmaster`) in the LDAP directory that stores user authentication and user preference information for this Calendar Server.

For more information, see the *iPlanet Calendar Server Installation Guide*.

To run these command line utilities, in most cases, you must change to the directory in which they are located (*`server-root`*`/cal/bin`), for example:

- On Solaris: `/opt/SUNWics5/cal/bin`
- On Unix systems other than Solaris: `/opt/iPlanet/CalendarServer5/cal/bin`
- On Windows NT: `c:\Program Files\iPlanet\CalendarServer5\cal\bin`

Starting and Stopping the Server

You can start and stop the iPlanet Calendar Server the using:

- `start-cal` and `stop-cal`
- 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 iPlanet Calendar Server is installed.

To start the iPlanet 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 iPlanet 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 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 iPlanet Calendar Server. Refer to the documentation provided with the operating system for instructions.

Access Control

Access control is required for group scheduling in iPlanet Calendar Server 5.0.

The four major components of access control are:

- **Who** – who is accessing it, also called the Universal Principal Name (UPN)
- **What** – what is being accessed, such as a calendar or a calendar component (an event, task, or property)
- **How**– what sort of access rights, such as read, write, or delete
- **Grant** – if a particular access right is granted or denied

The UPN value for the logged in user is the user's login name combined with the domain to which the user belongs. For example, user "`bill`" in domain "`siroe.com`" has the UPN "`bill@siroe.com`".

Who

Who is the principal value for an access control entry and indicates to whom the ACE applies: an individual, user, a domain, or a type of user. There are four types of users.

- All users is represented by the value: `@`
- Primary owners of a calendar is represented by the value: `@@p`
- Owners of a calendar is represented by the value: `@@o`
- Non-owners of a calendar is represented by the value: `@@n`

A user is defined for an access control entry by a string using the following formats:

```
user
```

```
user@domain
```

```
@domain
```

```
@
```

```
@@{n|o|p}
```

where:

- `user` refers to a specific user. For example, `jsmith`.
- `user@domain` refers to a specific user at a specific domain. For example, `jsmith@siroe.com`.
- `@domain` indicates any user in the specified domain. For example, `@siroe.com` would mean `jsmith@siroe.com`, `sally@siroe.com`, and anyone else at `siroe.com`.
- `@` means anybody.
- `@@` provides a mechanism to refer to other people with roles relative to the calendar. The currently defined roles are:
  - owner = `@@o`
  - primary owner = `@@p`
  - not an owner = `@@n`

Administrative Access

Administrators can be associated with a domain or with all domains. Administrators can perform any operation and are not subject to access control.

Anonymous Access

iPlanet Calendar Server supports the special principal name `anonymous`. The `anonymous` user can log in with any password and the access rights are set according to those defined for the password. The `anonymous` user name is not associated with any particular domain. To grant or deny access to an entire domain of users, use the form:

```
"@domain"
```

What

An ACE can be applied to the following targets:

- an entire calendar

- just the components (events or tasks) of the calendar
- just the properties of the calendar (such as its name, description or owners).

Table 3-1 shows the target value specifiers of an ACE.

Table 3-1 ACE Target Value Specifiers

Value	Specifies
c	components (events and tasks)
p	properties (name, owners, etc.)
a	all means the entire calendar including both components and properties

How

Table 3-2 shows the types of access control supported by iPlanet Calendar Server 5.0.

Table 3-2 Access Control Types

Type	Description
c	Act on behalf of for cancel access. This type grants a user the right to cancel components to which attendees have been invited on behalf of the calendar’s primary owner. This type of access does not need to be granted explicitly because it is implied when a user is designated as an owner (an owner other than the primary owner) of a calendar.
d	Delete access
e	Act on behalf of for reply access. This type grants a user the right to accept or decline invitations on behalf of the calendar’s primary owner. This type of access does not need to be granted explicitly because it is implied when a user is designated as an owner (an owner other than the primary owner) of a calendar.
f	Free / busy (availability) access only. Free / busy access means that a user may see scheduled time on a calendar, but is not allowed to see the event details. Instead, just the words “*Not Available*” appears by a scheduled time block. Blocks of time without any scheduled events are listed also, with the word “*Available*” next to them.
i	Act on behalf of for invite access. This type grants a user the right to create and modify components in which other attendees have been invited on behalf of the calendar’s primary owner. This type of access does not need to be granted explicitly because it is implied when a user is designated as an owner (an owner other than the primary owner) of a calendar.
r	Read access.

Table 3-2 Access Control Types *(Continued)*

Type	Description
s	Schedule access. Requests can be made, replies will be accepted, and other iTIP scheduling interactions will be honored.
w	Write access. This includes adding new items, deleting items, and modifying existing items.

Grant

The grant attribute specifies whether to grant or deny privileges for a specified access type, such as d (delete) or r (read). Table 3-3 shows the grant and deny values.

Table 3-3 Grant or Deny Access Control

Value	Description
"g"	grants access.
"d"	denies access.

Access Control Entries

An Access Control Entry (ACE) is defined by the four parts described above: what, who, how, and grant. The four parts of an ACE are separated by a caret (^), for example:

```
jsmith^c^wd^g
```

where:

- jsmith indicates who the ACE applies to
- c indicates what is being accessed (in this case only the components of a calenadar are accessed)
- wd indicates which access rights are to be granted or denied (in this case write and delete)
- g indicates the specified access rights (write and delete) are granted

ACE String Examples

```
jsmith^a^r^g
```

The above example grants the userID jsmith read access to both components and calendar properties

```
jsmith^c^wd^g
```

The above example grants jsmith write and delete access to components only.

```
@siroe.com^c^sfr^g
```

The above example grants all users in the siroe.com domain privileges to schedule, free/busy, and read access to components only.

```
@@o^c^wd^g
```

The above example grants owners write and delete access to components only.

```
jsmith^a^sfdwr^d
```

The above example denies jsmith all access to calendar data.

```
@^c^r^g
```

The above example grants read access to all users.

Access Control Lists

There may be multiple ACEs that apply to a particular target (a calendar, its properties, or its event and todo components). An Access Control List (ACL) is a semicolon-separated list of all the ACE strings that collectively apply to a particular target.

Placement of ACEs in an Access Control List

When iPlanet Calendar Server reads an ACL, it uses the first ACE it encounters that either grants or denies access to the target. Therefore, the placement order of an ACL is significant and it should be ordered so that the more specific ACEs are positioned before the more general ACEs.

For example, suppose the first ACE in an ACL for the calendar jsmith:sports grants read access to all owners and that user bjones is one of the owners. Then, the server encounters a second ACE that denies bjones read access to this calendar. In this case, the server grants bjones read access to this calendar and ignores the second ACE because it is a conflict. Therefore, to ensure that an access right for a specific user such as bjones is honored, the ACE for bjones should be positioned in the ACL before more global entries such as an ACE that applies to all owners of a calendar.

NOTE Calendars are created only when done so through the command line utilities or when a user logs in to the Calendar Server for the first time. Therefore, when a user sends a meeting request to another user, unless the user being invited has previously logged in to the Calendar Server or the administrator has explicitly created a calendar for the user being invited, the user sending the meeting request will receive an ACL error: *Calendar not found.*

Group Name Access Control

The Calendar Server Application Programming Interface (CSAPI) provides a call that accepts a group name and a "who" value in the URL form of a group name. The result of the call is a boolean value indicating whether or not the "who" value is a member of the group. For more information, see the iPlanet Calendar Server Programmers’ Reference.

Built-in Access Control

A primary owner has access to everything in a calendar. No access control checks are required for a primary owner accessing their own calendar. Only the primary owner of a calendar can set Access Control Entries on that calendar.

iPlanet Calendar Server 5.0 does not specify how access control is handled in a hierarchy. Working on behalf of another is allowed only by other owners of a calendar on behalf of the primary owner.

Example Access Control Operation

Table 3-4 shows an example of an ACL for the calendar with calendar ID "bill".

Table 3-4 Sample iPlanet Calendar Server ACL

What	who	how	grant
bill	`sally@siroe.com`	s	d
bill	`@siroe.com`	s	g
bill	`ldap://...`	s	g

Table 3-4 Sample iPlanet Calendar Server ACL *(Continued)*

What	who	how	grant
bill	`@siroe.com`	r	g
bill	`anonymous`	f	g

If `sally@siroe.com` attempts an operation that requires read access to calendar `"bill"`, the operation returns `"g"` (grant) because there is an ACE that grants anyone in the domain `siroe.com` read access to calendar `"bill"`. If, however, `sally@siroe.com` attempts to schedule with calendar `"bill"`, the operation returns `"d"` (deny) because the first entry in the ACL denies scheduling access to the exact UPN `sally@siroe.com`.

Suppose `jimbo@company22.com` attempts to schedule with "bill". The "who" values for the first and second ACEs are not in URL form, and their domains do not match the principal owner's domain. In this case, they are ignored. The third "who" value is in the form of a URL. Therefore, a CSAPI call is made to determine whether or not `jimbo@company22.com` is a member of the "who" value. If `jimbo@company22.com` is a member of the "who" value, then scheduling permission is granted. Otherwise, scheduling permission is denied because no other ACE grants or denies scheduling access.

Mapping iPlanet Calendar Server 2.x Access Control Data

Included with iPlanet Calendar Server 5.0 is a migration utility called `ics2migrate` (located in the *`server-root`*`\cal\bin` directory) that lets you automatically migrate calendar database and LDAP user information from version 2.x to version 5.0. For details on using `ics2migrate`, see Chapter 4 of the *iPlanet Calendar Server Installation Guide.*

In iPlanet Calendar Server 2.x, the following access control rules applied:

- Primary Owner has full access - no verification checking.
- Other Owners (synonymous with all owners) had full access to events and tasks, but only read access to calendar properties.

Table 3-5 describes how to migrate calendar access control data from iPlanet Calendar Server version 2.x to version 5.0.

Table 3-5 iPlanet Calendar Server version 2.x Access Control Migration

Read	Write	Who	What	How	Grant
Public	Private	@	<calid>	R	1
		@@O	<calid> COMPONENTS	WD	1
Private	Public	@	<calid> COMPONENTS	WD	1
		@@O	<calid>	R	1
Public	Public	@	<calid> COMPONENTS	WD	1
		@	<calid>	R	1
Private	Private	@@O	<calid>	R	1
		@@O	<calid> COMPONENTS	WD	1
Primary	Public	@	<calid> COMPONENTS	WD	1
Primary	Private	@@O	<calid> COMPONENTS	WD	1
Public	Primary	@	<calid>	R	1
Private	Primary	@@O	<calid>	R	1

Access Control Rules

Supporting iTIP access control is separate from how iPlanet Calendar Server implements Access Control Entries. Some examples of iTIP rules are:

- The organizer is the only attendee of a group event that can redistribute a change of the event to other attendees.
- Attendees can only change the parameters to their own attendee entries and must resubmit changes to the attendee line back to the organizer.
- Attendees can delegate their attendance to one or more other people. The delegates may not be listed in the attendee list of the event as sent by the organizer.
- Party crashing results in new, uninvited attendees being added to the attendee list. The organizer must approve the party crashing and will redistribute the updated event.

Calendar Locate Mechanism

The calendar locate mechanism is a plug-in that enables an arbitrary mechanism decide where a physical calendar resides. Its purpose is to bypass the need to know where a user’s physical calendar is stored, but instead to manage it programmatically.

The Database Wire Protocol calls into Calendar Lookup service to fully qualify a calendar ID (calid). By fully qualifying the calid, it can determine from the url the physical location of the calendar, along with its access protocol. For more information, see “Horizontal Scalability,” on page 20 and “Calendar Locate Mechanism Configuration,” on page 48.

iPlanet Calendar Server 5.0 provides the following implementations of the Calendar Lookup service:

- **local** – references calid's from a local database in the current running server instance. This is the mechanism used by iPlanet Calendar Server 2.x and it is the default configuration at installation.
- **algorithmic** – uses regular expression search criteria to decide what server hosts the calendar.

Using the Command Line Utilities

This section describes the tasks you can perform to administer iPlanet 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 line utilities syntax and usage, see Appendix , “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
- Managing LDAP Attributes
- Managing Calendar Resources
- Managing the Calendar Group Scheduling Queue

- Managing Plug-ins
- 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
- Using Solstice or Legato Software to Backup iPlanet Calendar Server

Table 3-6 lists the command line utilities and their purpose.

Table 3-6 Command Line Utilities

Use…	To…
`csattribute`	Manage the calendar attributes in the LDAP server.
`csbackup`	Backup individual calendars, users, and the calendar database.
`cscal`	Manage calendars and their properties.
`cscomponents`	Manage components (events and tasks) in a calendar.
`csdb`	Manage calendar databases.
`csexport`	Export a calendar in ICalendar or XML format.
`csimport`	Import a calendar in ICalendar or XML format.
`csplugin`	View, enable, or disable configured Calendar Server API (CSAPI) plug-ins.
`csresource`	Manage calendar resource objects such as conference rooms and video equipment.
`csrestore`	Restore individual calendars, users, and the calendar database.
`csschedule`	Manage schedule entries in the Group Scheduling Engine (GSE) queue.
`csstart`	Start the iPlanet Calendar Server.
`csstats`	Display counters in a iPlanet Calendar Server.
`csstop`	Stop the iPlanet Calendar Server.
`cstool`	General purpose tool to ping a running iPlanet Calendar Server instance to control and monitor the server, and to force the server to refresh its configuration.
`csuser`	Manage calendar users.

Managing Calendar Users

The csuser command line utility let you perform the following administrative tasks on calendar users:

- create a calendar user (see "Creating a Calendar User" on page 91)
- delete a calendar user (see "Deleting a Calendar User" on page 92)
- enable a calendar user (see "Enabling a Calendar User" on page 92)
- disable a calendar user (see "Disabling a Calendar User" on page 93)
- check if a user is enabled for calendaring (see "Checking if a User is Enabled" on page 93)
- list all calendar users or list a specified calendar user's configuration attributes (see "Displaying User Information" on page 93)
- reset the calendar user's configuration attributes to the default values (see "Resetting a User's Calendar Attributes" on page 93)

NOTE User IDs are sometimes case sensitive and sometimes not case sensitive depending on the application. For example, in iPlanet Calendar Server 5.0, the user `jsmith` can login as `JSMITH` or `jsmith`, but other users cannot invite `JSMITH` as an attendee, but instead must specify `jsmith`.

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.

NOTE Calendars are created only when done so through the command line utilities or when a user logs in to the Calendar Server for the first time. Therefore, when a user sends a meeting request to another user, unless the user being invited has previously logged in to the Calendar Server or the administrator explicitly has created a calendar for the user being invited, the user sending the meeting request will receive the error: *Calendar not found*.

Deleting a Calendar User

Use the `csuser` utility's `delete` command to delete a user from the iPlanet 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 106.)

For example, to delete JSmith from the iPlanet 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 iPlanet 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 unless a calendar by that name already exists. For example, if JSmith is not enabled for calendaring and logs on to the iPlanet 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 iPlanet 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 calendarA enable JSmith
```

The above command enables JSmith to log on to the iPlanet Calendar Server.

Disabling a Calendar User

Use the `csuser` utility's `disable` command to prevent a user from logging into the iPlanet 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 iPlanet Calendar Server data, enter the following command:

```
csuser disable JSmith
```

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 a User's Calendar Attributes

Use the `csuser` utility's `reset` command to restore the default settings of all calendar LDAP attributes for 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 iPlanet 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, such as its name and owners and who can access it
- display calendar information

NOTE Calendar Ids are case sensitive. For example, if you type `JSMITH` and the calendar ID is actually `jsmith`, you will not find the right calendar. This distinction differs from email addresses which are not case sensitive, such as `jsmith`@siroe.com which is equivalent to JSMITH@SIROE.COM.

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. By default, owners other than the primary owner are granted the access control permissions to a calendar for group scheduling that enable them to act on behalf of the primary owner. Other owners of a calendar can create events and todos and invite attendees, and they can accept or decline published events and todos from other calendar users on behalf of the primary owner. For example, a manager can designate other owners who can make scheduling changes to the manager's (primary owner) calendar such as creating events and accepting or declining events proposed by others. The default access control permissions are defined in the `ics.conf` file by the setting `calstore.calendar.default.acl`. For more information, see "Calendar Store Configuration" on page 47.

Use the `cscal` utility's `create` command to create a calendar. For example, to create a calendar with the visible name `exampleA` that is owned by `JSmith` and uses the default access control settings for group scheduling, enter the following command on the command line:

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

where:

- *`-n exampleA`* specifies that `exampleA` is the viewable name of the calendar.
- `-o JSmith` specifies the USERID of the primary owner.
- `calA` is the name that will be used as the second 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*, enables double booking, and specifies `RJones` as an other owner.

```
cscal -n exampleA -o JSmith -g sports -k yes -y RJones create calA
```

where:

- `-y RJones` specifies that this user is an owner (an owner othan the primary owner) of the calendar.
- `-g sports` specifies that the calendar is associated with a category called `sports`.

The following command creates a calendar similar to the example above and sets specific access control settings for group scheduling:

```
cscal -n exampleA -o JSmith -a "@@o^asfr^g" create calA
```

where:

- `-a "@@o^a^sfr^g"` grants owners schedule, free/busy, and read access privileges to both the components and calendar properties of this calendar for group scheduling.

Deleting Calendars

Use the cscal utility's delete command to delete one or more calendars from the iPlanet 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 106.)

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:

```
cscal -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 the calendar calA and does not allow double booking:

cscal -k no enable calA

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 iPlanet Calendar Server, enter the following command:

```
cscal disable calendarA
```

Modifying Calendar Properties

Use the cscal utility's modify command to change the properties of a calendar. For example, to change the group scheduling access control settings of calendarA and specify that RJones is an other owner, enter the following command:

```
cscal -a "@@o^c^wd^g" -y RJones calendarA
```

where:

- -a "@@o^c^wd^g" grants owners write and delete access only to the components (events and tasks) of calendarA.
- -y RJones specifies this USERID is an additional owner.

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
```

Managing Calendar Events and Tasks

The cscomponents utility lets you:

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

You must run the cscomponents utility on the local machine where your iPlanet Calendar Server is installed.

Use the cscomponents 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 iPlanet 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 103.

CAUTION **YOU MUST STOP THE CALENDAR SERVER BEFORE RESTORING, CREATING, OR DELETING THE CALENDAR DATABASE.**

The `csdb` utility lets you choose from three types of databases:

- `caldb` (calendar)
- `sessdb` (session)
- `statdb` (statistics)

For example:

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

```
csdb list
```

To create all new calendar databases if they were accidentally deleted, enter:

```
csdb create
```

To list information about the calendar database in the current directory in verbose mode, enter:

```
csdb -v -t caldb list
```

To delete the all databases 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 -t caldb 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 ICalendar or XML format.

NOTE The csexport and csimport utilities must be run locally on the machine where your iPlanet 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 ICalendar (text/calendar MIME) format to a file named jsmith.ics, enter the following command:

```
csexport -c JSmithcal 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 ICalendar (text/calendar MIME) format, enter the following command:

```
csimport -c JSmithcal 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.

Managing LDAP Attributes

Use the csattribute command line utility to manage the LDAP attributes used by Calendar Server.

For example:

To add the LDAP attribute icsCalendar with the value calA to the userID userA, enter the following command on the command line:

```
csattribute -a icsCalendar=calA add userA
```

To delete the LDAP attribute icsCalendar from userA, enter:

```
csattribute -a icsCalendar delete userA
```

Managing Calendar Resources

Use the csresource command line utility to manage the properties of a calendar associated with a resource object such as meeting rooms and equipment. You can run the csresource utility on the local machine where your iPlanet Calendar Server is installed or from remote machine. This section describes how to use csresource to view, add, delete, enable, or disable calendar resources.

For example:

To display a list of all resource calendars on the Calendar Server and their LDAP attributes, enter the following command on the command line:

```
csresource list
```

To create a resource calendar with the calendar ID Room1A and the viewable name (LDAP cn attribute) Galaxy, enter:

```
csresource -c Room1A create Galaxy
```

To following command performs the same action as the above example, but also allows double booking on this calendar:

```
csresource -k yes -c Room1A create Galaxy
```

The following command performs the same action as the above examples, but also specifies group scheduling access rights that let anyone read and write to it:

```
csresource -a "@^a^rw^g" -k yes -c Room1A create Galaxy
```

To delete the Galaxy resource, enter:

```
csresource delete Galaxy
```

To disable the Galaxy resource, enter:

```
csresource disable Galaxy
```

To enable the Galaxy resource, enter:

```
csresource enable Galaxy
```

To display a list of all LDAP attributes for the calendar Galaxy, enter:

```
csresource -v list Galaxy
```

Managing the Calendar Group Scheduling Queue

Use the `csschedule` command line utility to manage entries in the Group Scheduling Engine (GSE) queue. You must run the `csschedule` utility on the local machine where your iPlanet Calendar Server is installed. This section describes how to use `csschedule` to view or remove entries stored in the GSE queue.

For example:

To list all entries in the GSE queue, enter the following command on the command line:

```
csschedule list
```

To list the first ten entries stored in the GSE queue, enter:

```
csschedule -c 10 list
```

To list all entries in the GSE queue for a calendar with the calid `calendarA`, enter:

```
csschedule -v list calendarA
```

Managing Plug-ins

Use the `csplugin` command line utility to manage the plug-ins that have been configured for use with your Calendar Server.

Table 3-7 shows of the types of plug-ins supported by iPlanet Calendar Server 5.0.

Table 3-7 Plug-in Types Supported

Type	Description
`ac`	Augments or overrides the default group scheduling access control mechanism.

Table 3-7 Plug-in Types Supported

Type	Description
`attr`	Augments or overrides the mechanism for storing and retrieving user attributes.
`auth`	Augments or overrides the login authentication mechanism.
`locate`	Retrieves a calendar ID for the specified qualified URL.
`lookup`	Augments or overrides the default calendar lookup mechanism.
`xlate`	Augments or overrides the format translation of incoming and outgoing data.

For example:

To list all plug-ins configured for this server instance, enter the following command on the command line:

```
csplugin list
```

To activate the `lookup` type plug-in with the filename `mylookup`, enter:

```
csplugin activate -t lookup mylookup
```

To disable the group scheduling access control type plug-in with the filename `xyz_access` and delete it from the `plugin` directory, enter:

```
csplugin deactivate -t ac xyz_access -r
```

Backup and Restore Procedures

This section describes how to use the command line utilities to backup and restore iPlanet 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.

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

NOTE The version file caldb.conf located in the specified backup directory shows the version number of the database that was backed up.

Do not attempt to restore data backed up using iPlanet Calendar Server version 2.x because the utilities provided in versions 2.x and 5.0 are incompatible and doing so could result in data loss.

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 iPlanet Calendar Server

Backing-Up a Specified Calendar

Use the csbackup utility's calendar command to backup a calendar to a plain text file using ICalendar 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 ICalendar (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 ICalendar (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 bcakupdir 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 ICalendar 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 ICalendar (text/calendar MIME) format to a file named jsmith.ics, enter the following command:

```
csbackup -a JSmith 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 JSmith 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 ICalendar (text/calendar MIME) format to a file named jsmith.ics located in the backup directory backupdir, enter the following command:

```
csrestore -a JSmith 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 JSmith 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 accidently corrupted or deleted.

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

CAUTION You must stop the iPlanet 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 default directory (for example, `/var/opt/SUNWics5/csdb` on Solaris or `c:\Program Files\iPlanet\CalendarServer\var\csdb`), enter the following command:

```
csdb recover
```

Using Solstice or Legato Software to Backup iPlanet Calendar Server

We recommend using the Sun Solaris Solstice Backup or the Legato file backup and restore products to backup the iPlanet 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 iPlanet Calendar Server.

iPlanet 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 iPlanet Calendar Server to another.

Backing Up iPlanet 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 using the `-l` option.

For more information, see “Backup and Restore Procedures” on page 103).

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. The 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 iPlanet 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 iPlanet 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.)

Chapter 4

Monitoring the iPlanet Calendar Server

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

- Listing Counter Statistics
- Monitoring the iPlanet Calendar Server Log Files
- Pinging the iPlanet Calendar Server
- Refreshing the iPlanet Calendar Server Configuration

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 iPlanet 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 iPlanet Calendar Server counter statistics see, "Counters" on page 73.

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 about the `wcapstat` counter object every 10 seconds for one hour, enter:

```
csstats -i 360 -s 10 list wcap
```

Monitoring the iPlanet Calendar Server Log Files

Each calendar service writes status information it's own log file. Each log files is named after its associated service name as described in Table 4-1:

Table 4-1 iPlanet Calendar Server Log Files

Service Name	Log File name
`csadmind`	`admin.log`
`csdwpd`	`dwp.log`
`cshttpd`	`http.log`
`csnotifyd`	`notify.log`

The log files are stored in their own log directory:

- On Solaris, the default is:

  `/var/opt/SUNWics5/logs`

- On Unix systems other than Solaris, the default is:

  `/var/opt/iPlanet/CalendarServer5/logs`

- On Windows NT, the default is:

  `c:\Program Files\iPlanet\CalendarServer5\var\logs`

Each log file is rolled-over to a new log file with a new name based on the configured time and size limits as follows:

`<ServiceName>.<TimeStamp>.<#>`

For example, `admin.20000801115354 .1`, `http.20000801115354 .2`, and etc.

Log Event Severity Levels

iPlanet Calendar Server provides eight levels of severity for events reported to the log files as described in Table 4-2.

Table 4-2 iPlanet Calendar Server Log Error Severity Levels

Severity Level	Meaning
EMERGENCY	System is unusable. This level indicates events with the highest (most critical) severity.
ALERT	Action must be taken immediately.
CRITICAL	Critical condition.
ERROR	Error conditions.
WARNING	Warning conditions.
NOTICE	Normal, but signification condition. This is the default reporting level for each calendar service.
INFORMATION	Informational.
DEBUG	Debug-level message.

A log event is represented by a single line that shows the associated time stamp, server host name, severity level, process name (process ID), type of event, priority, and description. You can specify the level of severity of the events that Calendar Server reports to the log files by modifying certain configuration settings in the `ics.conf` file. For more information, see “Calendar Log Information Configuration” on page 39.

The log files should be inspected on a regular basis for EMERGENCY, ALERT, CRITICAL, ERROR, and WARNING level errors and, if found, these events should be examined for possible problems with the operation of the Calendar Server. The NOTICE and INFORMATION level log events are generated during normal operation of the Calendar Server and are provided to help you monitor server activity.

NOTE When requesting technical support for Calendar Server, you may be asked to provide the log files for help in resolving problems.

Pinging the iPlanet Calendar Server

Use `cstool` utility's `ping` command to verify that a iPlanet Calendar Server service is listening on a specified port number. (Pinging a service does not verify that a service is actually running, but indicates if it can accept a socket connection.) The service options are:

- admin (`csdamind` service)
- dwp (`csdwpd` service)
- ens (`enpd` service)
- http (`cshttpd` service) — the default if a service is not specified
- notify (`csnotifyd` service)

For example, to ping the `cshttpd` service to verify it is listening on port 80, enter the following command:

```
cstool -p 80 ping http
```

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 iPlanet Calendar Server Configuration

Use `cstool` utility’s `refresh` command to force a Calendar Server service to refresh its configuration. For example, to command the `cshttpd` service to refresh it's configuration, enter the following command:

```
cstool refresh
```

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

Appendix A

Command Line Utilities

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

Command Line Utility Syntax

iPlanet 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 same 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 value separated by a space unless indicated otherwise, such as the use of a semiclon delimited list in some cases.

- `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...
`csattribute`	local machine	manage the LDAP attributes of a calendar user or a calendar resource.
`csbackup`	local machine	backup individual calendars, users, and the calendar database.
`cscal`	local machine	manage calendars and their properties.
`cscomponents`	local machine	manage components in a calendar (components are events and tasks).
`csdb`	local machine	manage the calendar database.
`csexport`	local machine	export a calendar in iCalendar or XML format.

Table A-1 Command Line Utilities Summary *(Continued)*

Utility	Run from...	Use to...
`csimport`	local machine	import a calendar in iCalendar or XML format.
`csplugin`	local machine	view, enable, or disable configured Calendar Server API (CSAPI) plug-ins.
`csresource`	local machine	manage calendar resource objects such as conference rooms and equipment.
`csrestore`	local machine	restore individual calendars, users, and the calendar database.
`csschedule`	local machine	manage scheduling entries in the Group Scheduling Engine (GSE) queue.
`csstart`	local machine	start the iPlanet Calendar Server (on Windows NT, the server can also be started using the NT services dialog accessed from the Control Panel).
`csstats`	local machine	display counters in a Calendar Server.
`csstop`	local machine	stop the iPlanet Calendar Server (on Windows NT, the server can also be stopped using the NT services dialog accessed from the Control Panel).
`cstool`	local machine	ping a running iPlanet Calendar Server instance or command the server to refresh its configuration.
`csuser`	local machine	manage calendar users.

csattribute

The `csattribute` utility lets you manage the iPlanet Calendar Server attributes in the LDAP Server. It provides the following commands:

- `add` creates an LDAP attribute and value for a specified object.
- `delete` removes an LDAP attribute of a specified object.
- `list` displays the LDAP attributes of a specified object.

Syntax

```
csattribute [-q|-v] -a attribute=val [-t resource | user] add target
```

or

```
csattribute [-q|-v] -a attribute[=val] [-t resource | user] delete target
```

Requirements:

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

Examples

```
csattribute -a icsCalendar=calA add userA
```

The above command adds the LDAP attribute `icsCalendar` with the value `calA` to the userID `userA`.

```
csattribute -a icsCalendar delete userA
```

The above command deletes the LDAP attribute `icsCalendar` from `userA`.

```
csattribute list userA
```

The above command displays the attributes of `userA`.

Table A-2 describes the commands available for the `csattribute` utility.

Table A-2 `csattribute` Utility Commands

Command	Description
`add` *`target`*	Add an LDAP attribute and value to a specified target (user or resource object).
`list` *`target`*	List the attributes of a target object.
`delete` *`target`*	Delete an attribute from a target.
`version`	Display the version of the utility.

Table A-3 describes the `csattribute` utility command options.

Table A-3 `csattribute` Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.

Table A-3 `csattribute` Utility Command Options *(Continued)*

Option	Specifies…
`-q`	Quiet mode which: • 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.

`-a` *`attribute`*`=`*`val`*
or
`-a` *`attribute`*`=[`*`val`*`]`
	An LDAP attribute and value:
	• An attribute is required when using the `-a` option.
	• A value is required when the `-a` option is used with the `add` command, but is optional when the `-a` option is used with the `delete` and `list` commands.

`-t user|resource`	The type of target (a user or a resource object). The default is `user`.

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.

NOTE The version file `caldb.conf` located in the specified backup directory shows the version number of the database that was backed up.

Do not attempt to restore data backed up using iPlanet Calendar Server version 2.x because the utilities provided in versions 2.x and 5.0 are incompatible and doing so could result in data loss.

Syntax

```
csbackup [-q|-v] [-f] database target
```

or

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

or

```
csbackup [-q|-v] -a USERID [-b BASEDN] defcal target
```

Requirements:

- Must be run locally on the machine where iPlanet Calendar Server is installed.
- The iPlanet Calendar Server can be running or stopped.
- On Unix systems, must be run while logged in as the user and group under which the iPlanet 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-4 describes the commands available for the `csbackup` utility.

Table A-4 csbackup Utility Commands

Command	Description
`database` *`target`*	Backup the calendar database to the specified target database output directory.
`calendar` *`CALID target`*	Backup the specified CALID to the specified target output file. The data format of the file is assumed by the file extension, `.ics` for text/calendar and `.xml` for text/xml.
`defcal` *`USERID target`*	Backup the default calendar of the specified USERID to the specified target file. The data format of the file is assumed by the file extension, `.ics` for text/calendar and `.xml` for text/xml.
`version`	Display the version of the utility.

Table A-5 describes the `csbackup` utility command options.

Table A-5 `csbackup` Utility Command Options

Option	Specifies
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-a` *`USERID`*	The USERID of the calendar user to backup. This option is required for the `defcal` command. There is no default.
`-b` *`BASEDN`*	The base DN to be used for this user. The default is taken from the setting `local.ugldapbasedn` defined in the `ics.conf` file. The Base DN (distinguished name) is the entry in your LDAP directory used as the starting point from which searches will occur. For example, if you specify a base DN of `ou=people, o=siroe.com`, all LDAP search operations executed by iPlanet Calendar Server will examine only the `ou=people` subtree in the `o=siroe.com` directory tree.
`-c` *`CALID`*	The CALID to backup. This option is required with the `calendar` command. There is no default.
`-f`	To force any existing backup files to be deleted.
`-l`	To prepare the backup file for use with the Solstace or Legato backup programs. For more information, see “Backup and Restore Procedures” on page 103.

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 and group scheduling access control
- `reset` calendar properties to the default settings
- `version` displays the version number of the utility currently installed

Syntax

```
cscal [-q|-v] [-a ACEs] [-c charset] [-d description] [-g categories] [-k yes|no]
[-l langcode] [-m email] [-n name] [-o owner] [-y otherowners] create|modify CALID
```

or

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

or

```
cscal [-q|-v] [-k yes|no] [-o owner] enable [CALID]
```

NOTE Calendar Ids are case sensitive. For example, if you type `JSMITH` and the calendar ID is actually `jsmith`, you will not find the right calendar. This distinction differs from email addresses which are not case sensitive, such as `jsmith`@siroe.com which is equivalent to JSMITH@SIROE.COM.

Requirements:

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

Examples

```
cscal -o userA -n calA create calA
```

The above example creates the calendar with the CALID `calA` with `userA` as the primary owner with the visible name `exampleA` using the default access control settings (as defined by the setting `calstore.calendar.default.acl` in the `ics.conf` file).

```
cscal -a "@^a^rw^g" -g sports -y userB modify calA
```

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

```
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 -k no 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), but does not allow double-booking.

```
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 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-6 describes the commands available for the `cscal` utility.

Table A-6 cscal Utility Commands

Command	**Description**
`create` *`CALID`*	Creates the calendar specified by CALID.

Table A-6 cscal Utility Commands *(Continued)*

Command	Description
`delete [`*`CALID`*`]`	Deletes the calendar specified by CALID. If the `-o owner` option is specified, deletes all calendars whose primary owner is the specified owner.
`enable [`*`CALID`*`]`	Enable the calendar specified as CALID. If the `-o owner` option is specified, enables all calendars whose primary owner is the specified owner.
`disable [`*`CALID`*`]`	Disable the calendar specified as CALID. If the `-o owner` option is specified, disables all calendars whose primary owner is the specified owner.
`list [`*`CALID`*`]`	List properties of the calendar with the specified CALID. If the `-o owner` option is specified, lists all calendars whose primary owner is the specified owner.
`modify` *`CALID`*	Modify the properties of the calendar specified as CALID.
`reset [`*`CALID`*`]`	Reset the properties of the calendar specified as CALID to the default configuration settings.
`version`	Display the version of the utility.

Table A-7 describes the `cscal` utility command options.

Table A-7 cscal Utility Command Options

Option	Specifies...
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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-7 cscal Utility Command Options *(Continued)*

Option	Specifies...
`-a [`*`ACEs`*`]`	Access Control Entries (ACE) for a specified calendar which determines who can access a calendar for group scheduling and what types of permissions they have, such as create, delete, read, and write privileges. An ACE string, called an Access Control List (ACL), must be enclosed in quotation marks (`""`). Default is taken from the setting `calstore.calendar.default.acl` defined in the `ics.conf` file. For details on the ACE format, see "Access Control," on page 81.
`-c` *`charset`*	Character set. The default is no character set.
`-d` *`description`*	Description (a viewable comment about the purpose of the calendar). The default is no description.
`-g` *`category`*	Category. Multiple categories must be enclosed in quotation marks (`""`) and separated by spaces. The default is no category.
`-k yes|no`	If double booking is allowed for a calendar. For example, `yes` means the calendar can have more than one event scheduled for the same time slot. The default is taken from the setting `user.allow.doublebook` defined in the `ics.conf` file
`-l` *`langcode`*	Language code. The default is no language code.
`-m` *`email`*	Email address. The default is no email.
`-n` *`name`*	Name. The default is no name.
`-o` *`owner`*	Primary owner. The default setting is the CALID which is usually the same as the USERID.
`-y` *`otherowners`*	Other owners. Multiple owners must be enclosed in quotation marks (`""`) and separated by spaces. The default is no other owners.

cscomponents

The `cscomponents` utility lets you manage components (events and todos) 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 iPlanet Calendar Server is installed.
- The iPlanet Calendar Server can be running or stopped.
- On Unix systems, must be run while logged in as the user and group under which the iPlanet Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

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

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

```
cscomponents -v list calA
```

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

Table A-8 describes the commands available for the `cscomponent` utility.

Table A-8 `cscomponent` Utility Commands

Command	Description
`delete` *`CALID`*	Deletes events in the calendar with the specified CALID.
`list` *`CALID`*	Lists events in the calendar with the specified CALID.
`version`	Prints the version of the utility to the screen.

Table A-9 describes the `cscomponent` utility command options.

Table A-9 `cscomponent` Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.

Table A-9 `cscomponent` Utility Command Options *(Continued)*

Option	Specifies…
`-q`	Quiet mode which: • 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.
`-e` *`endtime`*	Ending time of the components. An end time of `0` means to the end of time. The default is `0`.
`-s` *`starttime`*	Starting time of the components. A start time of `0` means from the beginning of time. The default is `0`.
`-t event|task`	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, iPlanet Calendar Server creates one automatically.
- `delete` an existing calendar database. A database cannot be deleted while it is open (when the iPlanet Calendar Server is running).
- `list` information about the database.
- `recover` a damaged calendar database.
- `version` displays the version number of the utility currently installed.

Syntax

```
csdb [-q|-v] [-t caldb|sessdb|statdb] create|delete [dbdir]
```

or

```
csdb [-q|-v] [-t caldb|sessdb|statdb] list [dbdir]
```

or

```
csdb [-q|-v] [-f] [-t caldb|sessdb|statdb] recover [dbdir]
```

Requirements:

- Must be run locally on the machine where iPlanet Calendar Server is installed.
- The iPlanet 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 iPlanet Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Examples

```
csdb -t caldb create
```

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

```
csdb -t caldb delete
```

The above command deletes the databases in the current directory.

```
csdb -v -t caldb list
```

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

```
csdb recover
```

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

```
csdb -t sessdb list
```

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

```
csdb -t statdb recover
```

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

Table A-10 describes the commands available for the `csdb` utility.

Table A-10 `csdb` Utility Commands

Command	Description
`create [`*`dbdir`*`]`	Create databases in the specified database directory. If a database directory is not specified, the current directory is used.
`delete [`*`dbdir`*`]`	Delete databases in the specified database directory. If a database directory is not specified, the current directory is used.

Table A-10 `csdb` Utility Commands *(Continued)*

Command	Description
`list [`*`dbdir`*`]`	List information about the databases in the specified database directory. If a database directory is not specified, the current directory is used.
`recover [`*`dbdir`*`]`	Attempt to recover damaged databases in the specified database directory. If a database directory is not specified, the current directory is used.
`version`	Display the version of the utility.

Table A-11 describes the `csdb` utility command options.

Table A-11 `csdb` Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-f`	To force the recovery of the calendar database.
`-t caldb|sessdb|statdb`	Which database to perform an action on one of the following database types: • `caldb` (calendar) • `sessdb` (session) • `statdb` (statistics) **Note:** If `-t` is not specified all databases are assumed.

csexport

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

- `calendar` exports a specified calendar.

- `version` displays the version number of the utility currently installed.

Syntax

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

Requirements:

- Must be run locally on the machine where iPlanet Calendar Server is installed.
- The iPlanet Calendar Server can be running or stopped.
- On Unix systems, must be run while logged in as the user and group under which the iPlanet 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 iCalendar (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-12 describes the commands available for the `csexport` utility.

Table A-12 `csexport` Utility Commands

Command	Description
`calendar` *`outputfile`*	Export the calendar to the specified output file. The data format of the file is determined by the specified file name extension: `.ics` for iCalendar (text/calendar) `.xml` for XML (text/xml)
`version`	Display the version of the utility.

Table A-13 describes the `csexport` utility command options.

Table A-13 csexport Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-c` *`CALID`*	The CALID of the calendar to export. This option is required with the `calendar` command. There is no default.

csimport

The `csimport` utility lets you import a calendar from a file in iCalendar 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 inputfile
```

Requirements:

- Must be run locally on the machine where iPlanet Calendar Server is installed.
- The iPlanet Calendar Server can be running or stopped.
- On Unix systems, must be run while logged in as the user and group under which the iPlanet 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 iCalendar (text/calendar file) format.

```
csimport -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-14 describes the commands available for the `csimport` utility.

Table A-14 `csimport` Utility Commands

Command	Description
`calendar` *`inputfile`*	Import the calendar from the specified input file. The data format of the file is determined by the file name extension: • `.ics` for iCalendar (text/calendar) • `.xml` for XML (text/xml)
`version`	Display the version of the utility.

Table A-15 describes the `csimport` utility command options.

Table A-15 `csimport` Utility Command Options

Option	Specifies...
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-c` *`CALID`*	The CALID to of the calendar to import. This option is required with the `calendar` command. **Note:** If the specified CALID already exits, the imported data is merged with the current calendar. There is no default.

csplugin

The `csplugin` utility lets you manage the CSAPI plug-ins configured for your iPlanet Calendar Server installation. It provides the following commands:

- `activate` loads and starts a specified plug-in.
- `deactivate` shut downs and disables the specified plug-in type and plug-in name. (For descriptions of the supported plug-in types, see the “-t” option in Table A-17.)
- `list` displays all supported plug-ins.
- `version` displays the version number of the utility currently installed.

Syntax

```
csplugin [-q|-v] [-r] -t ac|attr|auth|locate|lookup|xlate activate|deactivate plugin
```

```
csplugin [-q|-v] list
```

Requirements:

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

Examples

```
csplugin -v list
```

The above command list details about all the supported plug-ins, including the type, name and the activation status of each plug-in configured for use with this server instance.

```
csplugin activate -t lookup mylookup
```

The above command loads and enables the `lookup` type plug-in with the filename `mylookup`.

```
csplugin deactivate -t lookup mylookup -r
```

The above command disables the `lookup` type plug-in with the filename `mylookup` and then deletes it from the `plugin` directory.

Table A-16 describes the commands available for the `csplugin` utility.

Table A-16 `csplugin` Utility Commands

Command	Description
`activate -t type` *`name`*	Load and enable the specified plug-in type and plug-in name. (For descriptions of the supported plug-in types, see the "`-t`" option in Table A-17.)
`deactivate -t type` *`name`*	Shut down and disable the specified plug-in type and plug-in name. (For descriptions of the supported plug-in types, see the "`-t`" option in Table A-17.)
`list`	List all the supported plug-in types, names, and activation status. (For descriptions of the supported plug-in types, see the "`-t`" option in Table A-17.)
`version`	Display the version of the utility.

Table A-17 describes the `csplugin` utility command options.

Table A-17 `csplugin` Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-r`	When used with the `activate` command, physically copies the plug-in into the iPlanet Calendar Server `plugin` directory. When used with the `deactivate` command, deletes the plug-in from the `plugin` directory.

Table A-17 `csplugin` Utility Command Options *(Continued)*

Option	Specifies…
`-t` *`type`*	Specifies one of the following supported types of plug-ins: • `ac` — augments or overrides the default group scheduling access control mechanism. • `attr` — augments or overrides the mechanism for storing and retrieving user attributes. • `auth` — augments or overrides the login authentication mechanism. • `locate` — retrieves a calendar ID for the specified qualified URL. • `lookup` — augments or overrides the default calendar lookup mechanism. • `xlate` — augments or overrides the format translation of incoming and outgoing data..

csresource

The `csresource` utility lets you manage calendar resources objects stored in the LDAP server and in the iPlanet Calendar Server calendar database. It provides the following commands:

- `create` adds a new resource for a specified calid
- `delete` removes a resource or all resources
- `disable` disables a resource or all resources
- `enable` enables a resource or all resources
- `list` displays a single resource or a list of all resources

NOTE The `csresource` utility is available only for calendars associated with a resource and will return an error if issued against a user’s calendar.

Calendar Ids are case sensitive. For example, if you type `ROOM1` and the calendar ID is actually `room1,` you will not find the right calendar. This distinction differs from email addresses which are not case sensitive, such as `jsmith`@siroe.com which is equivalent to JSMITH@SIROE.COM.

Syntax

```
csresource [-q|-v] [-a ACEs] [-b BASEDN] -c CALID [-d description] [-k yes|no] [-o owner] [-y otherowners] create name

csresource [-q|-v] [-b BASEDN] delete|disable|enable|list [name]
```

Requirements:

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

Examples

```
csresource list
```

The above command displays a list of all resource calendars and their LDAP attributes.

```
csresource -c room1 create meetingroom1
```

The above command creates a resource calendar with the calendar ID `room1` and the viewable name (LDAP `cn` attribute) `meetingroom1`.

```
csresource delete meetinggroom1
```

The above command deletes the resource calendar with the viewable name `meetinggroom1`.

```
csresource disable meetinggroom1
```

The above command disables the resource calendar with the viewable name `meetinggroom1`.

```
csresource -b enable meetinggroom1
```

The above example enables the calendar with the resource calendar with the viewable name `meetinggroom1` (users are allowed to read or write to it using the user interface) and allows double-booking.

```
csresource -a "@^a^rw^g" -y userB modify meetinggroom1
```

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

```
csresource -v list meetinggroom1
```

The above command displays the LDAP attributes of the resource calendar with the viewable name `meetinggroom1`.

Table A-18 describes the commands available for the `csresource` utility.

Table A-18 `csresource` Utility Commands

Command	Description
`create` *`name`*	Create a new resource for a specified calendar ID. If the *name* contains a space, it must be enclosed in quotation marks (" ").
`delete [`*`name`*`]`	Delete a resource or, if no resource *name* is specified, delete all resources. If the *name* contains a space, it must be enclosed in quotation marks (" ").
`enable [`*`name`*`]`	Enable a resource or, if no resource *name* is specified, enable all resources. all resources. If the *name* contains a space, it must be enclosed in quotation marks (" ").
`disable [`*`name`*`]`	Disable a resource or, if no resource *name* is specified, disable all resources. If the *name* contains a space, it must be enclosed in quotation marks (" ").
`list [`*`name`*`]`	Display a single resource calendar or, if no resource *name* is specified, list of all resource calendars. If the *name* contains a space, it must be enclosed in quotation marks (" ").

Table A-19 describes the `csresource` utility command options.

Table A-19 `csresource` Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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 `csresource` Utility Command Options *(Continued)*

Option	Specifies…
`-a` *`ACEs`*	Access Control Entries (ACE) for a specified calendar which determines who can access a calendar associated with a resource object (such as a conference room) for group scheduling and what types of permissions they have, such as create, delete, read, and write privileges. An ACE string, called an Access Control List (ACL), must be enclosed in quotation marks (`""`). Default is taken from the setting `resource.default.acl` defined in the `ics.conf` file. For details on the ACE format, see "Access Control," on page 81.
`-b` *`BASEDN`*	The LDAP base DN (distinguished name) to be used for the specified resource. The default is taken from the setting `local.ugldapbasedn` defined in the `ics.conf` file.
`-c` *`CALID`*	The icsCalendar attribute. This option is required with the `create` command and there is no default.
`-d` *`description`*	Description (a viewable comment about the purpose of the calendar). The default is no description.
`-k yes|no`	If double booking is allowed for a calendar associated with a resource object such as a conference room. If `yes`, the resource calendar can have more than one event scheduled for the same time slot. .Default is taken from the setting `resource.allow.doublebook` defined in the `ics.conf` file.
`-o` *`owner`*	Primary owner. Default is taken from the setting `service.admin.calmaster.userid` defined in the `ics.conf` file.
`-y` *`otherowners`*	Other owners. Multiple owners must be enclosed in quotation marks (`""`) and separated by spaces. The default is no other owners.
`version [`*`name`*`]`	Display the version of the utility.

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.

NOTE The version file `caldb.conf` located in the specified backup directory shows the version number of the database that was backed up.

Do not attempt to restore data backed up using iPlanet Calendar Server version 2.x because the utilities provided in versions 2.x and 5.0 are incompatible and doing so could result in data loss.

Syntax

```
csrestore [-v|-q] [-f] database inputdir
```

or

```
csrestore [-v|-q] -c CALID calendar inputfile
```

or

```
csrestore [-v|-q] -a USERID [-b BASEDN] defcal inputfile
```

Requirements:

- Must be run locally on the machine where iPlanet Calendar Server is installed.
- The iPlanet 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 iPlanet 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 iCalendar (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 iCalendar (text/calendar file) format using the csbackup or csexport utility.

Table A-20 describes the commands available for the csrestore utility.

Table A-20 csrestore Utility Commands

Command	Description
database *inputdir*	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.
calendar *inputfile*	Restore the specified CALID from the specified input file. The data format of the file is determined by the file name extension: .ics for iCalendar (text/calendar). .xml for XML (text/xml). If the specified CALID already exists, the calendar's data is cleared before it is restored.
defcal *inputfile*	Restore 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 iCalendar (text/calendar). .xml for XML (text/xml).
Version	Display the version of the utility.

Table A-21 describes the csrestore utility command options.

Table A-21 csrestore Utility Command Options

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

Table A-21 `csrestore` Utility Command Options *(Continued)*

Option	Specifies...
`-q`	Quiet mode which: • 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.
`-a` *`USERID`*	The USERID to restore. This option is required with the `defcal` command. There is no default.
`-b` *`BASEDN`*	The LDAP base DN (distinguished name) to be used for the specified USERID. The default is taken from the setting `local.ugldapbasedn` defined in the `ics.conf` file.
`-f`	To force any existing database files to be deleted.
`-c` *`CALID`*	The CALID (calendar ID) to restore. This option is required with the `calendar` command. There is no default.

csschedule

The `csschedul` utility lets you manage schedule entries stored in the Group Scheduling Engine (GSE) queue. It provides the following commands:

- `list` displays entries held in the Group Scheduling Engine queue requested by a specifed calendar ID.
- `delete` removes an entry from the Group Scheduling Engine queue requested by a specifed calendar ID.
- `version` displays the version number of the utility currently installed.

Syntax

```
csschedule [-q|-v] [-c count] [-e endtime] [-s starttime] [-t scheduletime -o offset] [-u
uid] list [CALID]
```

or

```
csschedule [-q|-v] [-t scheduletime -o offset -u UID -n sequencenumber -r RID] list [CALID]
```

or

```
csschedule [-q|-v] [-t scheduletime -o offset -u UID -n sequencenumber -r RID] delete CALID
```

or

```
csschedule [-q|-v] [-s starttime] [-e endtime] delete [CALID]
```

Requirements:

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

Examples

```
csschedule -v list
```

The above command lists, in detail, all entries stored in the GSE queue.

```
csschedule -c 10 list
```

The above command lists the first ten entries stored in the GSE queue.

```
csschedule -s 20001231T103045Z  -e 20001231T113045Z list
```

The above command lists the entries in the GSE queue in the scheduled between 10:30:45 to 11:30:45 on 12/31/2000.

```
csschedule -v -t 20001231T103045Z -o 2 -u 1111 -r 0 -n 0 list calA
```

The above command lists the entry in the GSE queue for calendar `calA` that is scheduled at 10:30:45, with an offset number of `2` at the time 10:30:45, on 12/31/2000 with the unique identifier `1111`, the recurrence ID `0`, and the sequence number `0`.

```
csschedule -v -t 20001231T103045Z -o 1 -u 1111 -r 0 -n 0 delete calA
```

The above command deletes this one entry in the GSE queue for calendar `calA` at 13:30:45, the first offset at time 13:30:45, on 12/31/2000, with the unique identifier 1111, the recurrence ID 0, and the sequence number 0.

```
csschedule -v -s 20001231T103045Z -e 20001231T163045Z delete
```

The above command deletes entries in the GSE that are scheduled between 10:30:45 and 16:30:45 on 12/31/2000.

```
csschedule -v delete
```

The above command deletes all entries in the GSE queue.

Table A-22 describes the commands available for the `csschedule` utility.

Table A-22 `csschedule` Utility Commands

Command	Description
`list`	Display entries held in the Group Scheduling Engine queue requested by a specified calendar ID.
`delete`	Delete an entry from the Group Scheduling Engine queue requested by a specified calendar ID.
`version`	Display the version of the utility.

Table A-23 describes the `csschedule` utility command options.

Table A-23 `csschedule` Utility Command Options

Option	Specifies…
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-c` *`count`*	The number of GSE queue entries to list. For example, enter `10` if you want to examine ten entries in the queue.
`-e` *`endtime`*	The ending time of the entry in the GSE queue where `0` means to the end of time. The default is `0`.
`-n` *`sequencenumber`*	The sequence number of the event or task in the queue.
`-o` *`offset`*	An offset number for a schedule time. The offset number uniquely identifies an entry in the GSE queue when there is more than one entry scheduled at the same time.
`-r` *`RID`*	The recurrence ID (RID) of the event or task (todo). An RID is a semicolon delimited list of strings that identify each occurrence of a recurring event or todo.

Table A-23 `csschedule` Utility Command Options *(Continued)*

Option	Specifies…
`-s` *`starttime`*	The starting time of the entry in the GSE queue where `0` means from the beginning of time. The default is `0`.
`-t` *`scheduletime`*	A schedule time, for example: `20001231T103045Z`
`-u` *`UID`*	The unique identifier (UID) of an entry in the GSE queue.

csstart

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

- `check` determines if all iPlanet Calendar Server services or a specified service is running.
- `list` displays all the iPlanet Calendar Server services or a specified service.
- `service` starts all iPlanet Calendar Server services or a specified service.
- `version` displays the version number of the utility currently installed.

Syntax

```
csstart [-q|-v] check|list [servicename]
```

or

```
csstart [-q|-v] [-f] service [servicename]
```

Requirements:

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

Examples

```
csstart service
```

The above command starts all iPlanet Calendar Server services using the default ports and in the default start order.

```
csstart service http
```

The above command starts the local iPlanet Calendar Server HTTP service.

```
csstart check
```

The above command checks if all local iPlanet Calendar Server services are started.

```
csstart list
```

The above command lists all local iPlanet Calendar Server services that are started.

```
csstart service
```

The above command starts all local iPlanet Calendar Server services.

Table A-24 describes the commands available for the `csstart` utility.

Table A-24 `cscstart` Utility Commands

Command	Description
`check [`*`servicename`*`]`	Check if a specified iPlanet Calendar Server service is running, or check if all Calendar Server services are running if a service name is not specified, where *`servicename`* can be one of the following: • `ens` — a generic event registration and notification service that can be shared by other iPlanet servers • `notify` — Calendar Server notification daemon • `admin` — Calendar Server administration daemon (installation required on every server machine) • `dwp` — Calendar Server dataBase daemon (only started with remote database configuration) • `http` — Calendar Server daemon
`list [`*`servicename`*`]`	List all iPlanet Calendar Server services (or a specified service) and their session IDs.

Table A-24 `cscstart` Utility Commands *(Continued)*

Command	Description
`service [`*`servicename`*`]`	Start a specified iPlanet Calendar Server service or all of its services if no service name is specified. The iPlanet Calendar Server 5.0 services should be started in the following order: 1.`ens` — a generic event registration and notification service that can be shared by other iPlanet servers 2.`notify` — Calendar Server notification daemon 3.`admin` — Calendar Server administration daemon (installation required on every server machine) 4.`dwp` — Calendar Server dataBase daemon (only started with remote database configuration) 5.`http` — Calendar Server daemon For more information on iPlanet Calendar Server services, see the following sections: •“Calendar Server Services,” on page 17. •“Starting and Stopping the Server,” on page 80. •“Services Configuration Settings in ics.conf,” on page 40.
`version`	Display the version of the utility.

Table A-25 describes the `csstart` utility command options.

Table A-25 `csstart` Utility Command Options

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

Table A-25 `csstart` Utility Command Options *(Continued)*

Option	Specifies…
`-f`	To force a specified iPlanet Calendar Server service (or all currently running iPlanet Calendar Server services if a service is not specified) to: 1. Stop (similar to a `kill -9` command on Unix or stopping a service from the Windows NT Task Manager if previous attempts have failed). 2. Cleanup any database problems. 3. Start all services. .**Note:** It is recommended that you use the `stop-cal` and `start-cal` utilities instead to stop and start the Calendar Server.
`-q`	Quiet mode which: • 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.

csstats

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

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

NOTE For more information on counters, see “Counters” on page 73.

Syntax

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

Requirements:

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

- On Unix systems, must be run while logged in as the user and group under which the iPlanet 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-26 describes the commands available for the `csstats` utility.

Table A-26 `csstats` Utility Commands

Command	Description
`list [`*`subsystem`*`]`	List counter statistics about a specified iPlanet Calendar Server subsystem or, if one is not specified, display basic information about the available subsystems which are: • `alarm` — monitoring of services alarm notifications • `auth` — login authentication • `db` — calendar database • `disk` — disk usage monitoring • `gse` — Group Scheduling Engine • `http` — HTTP transport • `response` — server response times • `sess` — server session status • `wcap` — Web Calendar Access Protocol
`Version`	Display the version of the utility.

Table A-27 describes the `csstats` utility command options.

Table A-27 `csstats` Utility Command Options

Option	Specifies...
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-i` *`iterations`*	The number of times to repeat statistical lookups. Default is 1.
`-r` *`registry`*	The name and location of the file that stores counter statistics. The default is: *`server-root`*`/cal/bin/counter/counter`. For example: `/opt/SUNWics5/cal/bin/counter/counter`
`-s` *`delay`*	The amount of time (in seconds) to wait before displaying each statistical lookup. The default is 1 second.

csstop

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

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

Syntax

```
csstop [-q|-v] check | list [servicename]
```

or

```
csstop [-q|-v] [-f] service [servicename]
```

Requirements:

- Can be run on the local machine where iPlanet Calendar Server is installed or from a remote machine.
- The iPlanet Calendar Server must be running.

Examples

```
csstop service
```

The above command stops all local iPlanet Calendar Server services.

```
csstop service http
```

The above command stops the local iPlanet Calendar Server HTTP service.

```
csstop check
```

The above command checks if all local iPlanet Calendar Server services are stopped.

```
csstop list
```

The above command lists all local iPlanet Calendar Server services that are started.

Table A-28 describes the commands available for the `csstop` utility.

Table A-28 `csstop` Utility Commands

Command	Description
`check [`*`servicename`*`]`	Check if a specified iPlanet Calendar Server service is running or check if all of its services are running if no service name is specified. Check if a specified iPlanet Calendar Server service is running (or check if all Calendar Server services are running if a service name is not specified) where *`servicename`* can be one of the following: • `ens` — a generic event registration and notification service that can be shared by other iPlanet servers • `notify` — Calendar Server notification daemon • `admin` — Calendar Server administration daemon (installation required on every server machine) • `dwp` — Calendar Server dataBase daemon (only started with remote database configuration) • `http` — Calendar Server daemon
`list [`*`servicename`*`]`	List all iPlanet Calendar Server services (or a specified service) and their session IDs.

Table A-28 csstop Utility Commands *(Continued)*

Command	Description
service [*servicename*]	Stop a specified iPlanet Calendar Server service or all of its services if no service name is specified. The iPlanet Calendar Server 5.0 services should be started in the following order and stopped in the reverse order as follows: 1.ens — a generic event registration and notification service that can be shared by other iPlanet servers 2.notify — Calendar Server notification daemon 3.admin — Calendar Server administration daemon (installation required on every server machine) 4.dwp — Calendar Server database daemon (only started with remote database configuration) 5.http — Calendar Server daemon For more information on iPlanet Calendar Server services, see the following sections: •“Calendar Server Services,” on page 17. •“Starting and Stopping the Server,” on page 80. •“Services Configuration Settings in ics.conf,” on page 40.
version	Display the version of the utility.

Table A-29 describes the csstop utility command options.

Table A-29 csstop Utility Command Options

Option	Specifies
-v	Verbose mode which displays all available information about the command being performed. Default is off.
-q	Quiet mode which: • 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-29 `csstop` Utility Command Options *(Continued)*

Option	Specifies
`-f`	To force an iPlanet Calendar Server service to stop (similar to a `kill -9` command on Unix or stopping a service from the Windows NT Task Manager) if previous attempts have failed. This option is available only with the `service` command. **Note:** It is recommended that you use the `stop-cal` and `start-cal` utilities instead to stop and start the Calendar Server.

cstool

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

- `ping` a Calendar Server service to verify that it is listening on a specified port.
- `refresh` a Calendar Server service to force it to refresh its configuration settings.
- `version` lets you display the version number of the utility currently installed.

Syntax

```
cstool [-q|-v] [-h host] [-p port] [-t timeout] ping [servicename]
```

or

```
cstool [-q|-v] [-h host] refresh [servicename]
```

Requirements:

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

Examples

```
cstool ping http
```

The above command pings the `cshttpd` service.

```
cstool -p 80 -h calserver -p 80 ping http
```

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

```
cstool refresh
```

The above command forces a local iPlanet Calendar Server to refresh the `cshttpd` service’s configuration.

Table A-30 describes the commands available for the `cstool` utility.

Table A-30 `cstool` Utility Commands

Command	Description
`ping [`*`http`*`]`	Ping the iPlanet Calendar Server http service.
`refresh [`*`hostname`*`]`	Force iPlanet Calendar Server to refresh the configuration of a specified service or if no service is specified, refresh the configuration of all iPlanet Calendar Server services. The iPlanet Calendar Server service options are: • admin (`csdamind` service) • dwp (`csdwpd` service) • ens (`enpd` service) • http (`cshttpd` service) • notify (`csnotifyd` service)
`version`	Display the version of the utility.

Table A-31 describes the `cstool` utility command options.

Table A-31 `cstool` Utility Command Options

Option	Specifies...
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-h` *`host`*	Specifies host name of the machine on which iPlanet Calendar Server is running. The default value is set at installation and taken from the setting `local.hostname` defined in the `ics.conf` file. Use this option if you are accessing an iPlanet Calendar Server running on a remote machine.

Table A-31 `cstool` Utility Command Options *(Continued)*

Option	Specifies...
`-p` *`port`*	The port of the specified service of if no service is specified, use the default value of the port as defined in the `ics.conf` file.
`-t` *`timeout`*	The amount of time (in seconds) to wait for a response from the server. The default is 120 seconds.

csuser

The `csuser` utility lets you manage calendar user information stored in an LDAP directory server and in the iPlanet Calendar Server calendar database. 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 iPlanet Calendar Server.
- `enable` a user to log on to the iPlanet Calendar Server.
- `list` a user's calendar attributes.
- `reset` a user's calendar attributes to the default settings.

NOTE This tool is provided as a convenience, but it is recommended that administrators provision user attributes using a directory service or a customized user provisioning mechanism.

Syntax

```
csuser [-q|-v] [-a ACEs] [-b BASEDN] [-c CALID] -g givenname [-k yes|no] -s
surname -y userpassword create USERID
```

or

```
csuser [-q|-v] [-b BASEDN] check|delete|disable|enable|list|reset [USERID]
```

Requirements:

- The Calendar Server can be running or stopped.

- Must be run locally on the machine where iPlanet Calendar Server is installed.
- 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 iPlanet Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

NOTE User IDs are sometimes case sensitive and sometimes not case sensitive depending on the application. For example, in iPlanet Calendar Server 5.0, the user `jsmith` can log in as `JSMITH` or `jsmith`, but other users cannot invite `JSMITH` as an attendee, but instead must specify `jsmith`.

Calendar Ids are case sensitive. For example, if you type `ROOM1` and the calendar ID is actually `room1,` you will not find the right calendar. This distinction differs from email addresses which are not case sensitive, such as `j`smith@siroe.com which is equivalent to JSMITH@SIROE.COM.

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 iPlanet 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 iPlanet Calendar Server.

NOTE The command in this example prevents JSmith from being able to log in to the iPlanet 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 iPlanet Calendar Server, JSmith retains access to calendar data until he logs off.

```
csuser enable JSmith
```

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

```
csuser -v list JSmith
```

The above example lists all calendar attributes for JSmith.

```
csuser -v list "user*"
```

The above example lists all calendar USERIDs prefixed with the string: `user`

```
csuser reset JSmith
```

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

Table A-32 describes the commands available for the `csuser` utility.

Table A-32 `csuser` Utility Commands

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

Table A-33 describes the `csuser` utility command options.

Table A-33 `csuser` Utility Command Options

Option	Specifies...
`-v`	Verbose mode which displays all available information about the command being performed. Default is off.
`-q`	Quiet mode which: • 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.
`-a [`*`ACEs`*`]`	Access Control Entries (ACE) for a specified calendar which determines who can access a calendar for group scheduling and what types of permissions they have, such as create, delete, read, and write privileges. An ACE string, called an Access Control List (ACL), must be enclosed in quotation marks (`""`). Default is taken from the setting `calstore.calendar.default.acl` defined in the `ics.conf` file. For details on the ACE format, see “Access Control,” on page 81.
`-b` *`BASEDN`*	The LDAP base DN (distinguished name) of the specified USERID. The default is taken from the setting `local.ugldapbasedn` defined in the `ics.conf` file.
`-c` *`CALID`*	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.
`-g` *`givenname`*	The user's LDAP givenname (first name). This option is required. There is no default.
`-k yes|no`	If double booking is allowed for a user’s calendar. If `yes`, the user’s calendar can have more than one event scheduled for the same time slot. Default is taken from the setting `user.allow.doublebook` defined in the `ics.conf` file.
`-s` *`surname`*	The user's LDAP surname (last name). This option is required. There is no default.
`-y` *`userpassword`*	The calendar user's password. This option is required. There is no default.

Appendix B

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
`iostat`	Provides information about disk I/O and CPU usage.
`lsof`	Provides information about open file descriptors. Available in source from: :`ftp://vic.cc.purdue.edu/pub/tools/unix`.
`lslk`	Provides information about file system locks. Available in source from: :`ftp://vic.cc.purdue.edu/pub/tools/unix`
`netstat`	Provides statistics about network functions.
`nslookup`	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).
`ping`	Allows you to query the status of a remote host or network gateway.
`sar`	Unix SysV performance monitoring tool. Useful for gathering system information over a longer period of time to use in long term planning, for example.

Table B-1 General Unix tools *(Continued)*

Tool	Description
tcpdump	Public domain tools used in debugging and to monitor network traffic.
top	Provides quick, easy monitoring of processes and CPU activities. (This is a public domain tool that works on most Unix platforms.)
trace	Similar to truss on Solaris. Sometimes included by the vendor; otherwise, you can download this tool from an Internet site.
traceroute	Determines the path a packet takes throughout the Internet to reach its final destination.
vmstat	Provides statistics about process, virtual memory, disk, trap, and CPU activity.

Table B-2 System monitoring tools - Sun Solaris

Tool	Description
lockstat	Provides information on OS and application locking. Available on Solaris 2.6 only.
mpstat	Provides statistics about each processor on the system
pmap	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 /usr/proc/bin.)
proctool	Monitors processes and threads. (Available from Sun's web site.)
snoop	Monitors network traffic; indispensable when debugging low-level packets.
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 ncsize and ufs_ninode parameters and even has a mode to tune the operating system automatically.
truss	Provides information about which system calls a process makes.

Table B-3 System monitoring tools - HP-UX

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

Table B-4 System monitoring tools - SGI Irix

Tool	Description
`dkstat`	Similar to `iostat`. Provides information about disk I/O and CPU use.
`gmemusage (Irix 6.x)`	X windows tool for viewing information about virtual memory.
`netstat -C`	Provides real-time, full-screen data.
`osview`	Provides full-screen information; combines functionality of `vmstat`, `mpstat`, and `netstat`.
`par`	Similar to `truss` on Solaris. Provides information about system calls made by a process.
`Performance Copilot`	An SGI add-in package.

Appendix C

Time Zones

This appendix describes how iPlanet 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 iPlanet Calendar Server.

The file `timezones.ics` shown on page 166 contains the representation of the time zones supported by iPlanet Calendar Server. This file is located in the *`server-root`*`/cal/bin/data` directory (for example, `opt/SUNWics5/cal/bin/data`). This file contains 91 time zones.

At startup, the iPlanet Calendar Server reads the file `timezones.ics`, 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 Reference.*

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.

Time Zone Administration

This section describes how to:

- modify the iPlanet Calendar Server time zone list

- add a new time zone to iPlanet Calendar Server

To modify the iPlanet Calendar Server time zone list, an administrator must edit the file `timezones.ics` located in the *`server-root`*`/cal/bin/data` directory (for example, `opt/SUNWics5/cal/bin/data`). This file contains the iPlanet Calendar Server format of 91 time zones. (See iCalendar Specification for more details about the VTIMEZONE format).

Time zones are identified by the property `TZID`. For example, iPlanet 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 DAYLIGHT:

- 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 iPlanet Calendar Server

- Create TZID name not already included in the TZID list.
- Create a representation of the time zone.

- 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:

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

Customizing Time Zones in the iPlanet 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 iPlanet 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, that administrators should not modify the time zone list unless absolutely necessary.

iPlanet Calendar Server Time Zones Table

The iPlanet Calendar Server time zone table is defined in a plain text file called timezones.ics located in the *server-root*/cal/bin/data directory (for example, opt/SUNWics5/cal/bin/data). It includes 91 time zones based on the JDK (Java Development Kit) version 1.1.

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
```

Index