Administrator’s Guide

iPlanet™ Calendar Server

Release 5.1

January 2002

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.

CETTE DOCUMENTATION EST FOURNIE “EN L'ÉTAT”, ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

List of Tables

List of Figures

About This Guide

This guide explains how to administer iPlanet™ Calendar Server and its accompanying software components. Calendar Server provides a scalable, Web-based solution for centralized calendaring and scheduling for enterprises and service providers. Calendar Server supports personal calendars as well as group and resource scheduling. Topics in this chapter include:

- Who Should Read This Guide
- What You Need to Know
- How This Guide is Organized
- Document Conventions Used in This Guide
- Where to Find Related Information

Who Should Read This Guide

This guide is intended for administrators and support specialists who are responsible for administering and configuring iPlanet Calendar Server at their 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 (Solaris™ Operating Evironment, other UNIX® system, or Windows NT system)
- Lightweight Directory Access Protocol (LDAP), if you plan to use an LDAP directory server to store user information

How This Guide is Organized

Table 1 Organization of the iPlanet Calendar Server Administrator's Guide

Chapter or Appendix	Description
About This Guide (this chapter)	Describes the audience, requirements, organization, document conventions, and related information.
Chapter 1, "Introduction to iPlanet Calendar Server"	Provides a high-level overview of iPlanet Calendar Server, including the components, architecture, interfaces, and protocols.
Chapter 2, "Managing Calendar Server Users and Calendars"	Describes how to deploy and manage Calendar Server users and calendars.
Chapter 3, "Managing the Calendar Server"	Describes the general Calendar Server tasks such as starting and stopping services and configuring the server.
Chapter 4, "Managing Calendar Server Access Control"	Describes how the Calendar Server uses Access Control Lists (ACLs) to determine the access control for calendars, calendar properties, and calendar components.
Chapter 5, "Managing Calendar Server Databases"	Describes how to administer and maintain the Calendar Server databases and data.
Chapter 6, "Backing Up and Restoring Calendar Server Data"	Describes how to back up and restore Calendar Server data.
Chapter 7, "Calendar Server Command-Line Utilities"	Describes the Calendar Server command-line utilities, including the commands, options, values, syntax, usage rules, and examples.
Chapter 8, "Calendar Server Configuration"	Describes the parameters you can configure for the Calendar Server.
Appendix A, "Monitoring Tools"	Lists system tools you can use to monitor the Calendar Server environment.
Appendix B, "Time Zones"	Describes how Calendar Server processes time zones and how to add a new time zone table or modify the time zone tables supplied.
Glossary	Describes Calendar Server terms.
Index	

Document Conventions Used in This Guide

This guide uses conventions for the Solaris operating environment. If you are running the Calendar Server on another UNIX system or a Windows NT system, use the file path equivalents for your operating system whenever Solaris file paths are shown.

Monospaced Font

The `Monospaced Font` typeface is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

Italicized Font

The *Italicized Font* typeface represents text that you enter using information unique to your installation. It is used for server directory paths and names. For example, in this guide you will see directory path references in the form:

server-root`/opt/cal/bin`

In these situations, *server-root* represents the directory where you installed the Calendar Server. For example:

- The default installation directory 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 directory is `/opt/iPlanet/CalendarServer5/cal`, where *server-root*`/cal` represents:

  `/opt/iPlanet/CalendarServer5/cal`

- On Windows NT, the default installation directory is `c:\Program Files\iPlanet\CalendarServer5\cal`, where *server-root*`/cal` represents:

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

Square Brackets [ ]

Square (or straight) brackets [ ] enclose optional parameters. For example, the syntax for the `csdb` utility `check` command is: `csdb check [`*dbdir*`]`

The *dbdir* parameter is optional. If you omit *dbdir*, the `csdb` utility uses the current directory specified in the `ics.conf` file.

Vertical Bar (|)

A vertical bar (|) separates alternatives in a horizontal list. For example, the syntax for the `csdb` utility `create` or `delete` command uses the vertical bar to specify various options you can use:

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

Command-Line Prompts

Command-line prompts (such as % for a C-Shell or $ for a Korn or Bourne shell) are not shown in the examples in this guide. The actual prompt you will see depends on the operating system you are using. Unless specifically noted, enter each command as it appears in this document.

Where to Find Related Information

iPlanet Calendar Server includes documentation for administrators, developers, and end users. In addition to this guide, the following Calendar Server documents are available in PDF and HTML formats on the iPlanet documentation Web site:

- *iPlanet Calendar Server Installation Guide*
- *iPlanet Calendar Server Programmer's Manual*
- *iPlanet Calendar Server Release Notes*
- *iPlanet Messaging and Collaboration Event Notification Service Manual*

For end users, online help is available from Calendar Express.

To view all of the Calendar Server documentation, see the following Web site:

`http://docs.iplanet.com/docs/manuals/calendar.html`

Where to Find This Guide Online

This guide is available online in PDF and HTML formats at:

`http://docs.iplanet.com/docs/manuals/calendar.html`

Chapter 1

Introduction to iPlanet Calendar Server

iPlanet Calendar Server is a scalable, Web-based solution for centralized calendaring and scheduling for enterprises and service providers. The Calendar Server supports personal and group calendars as well calendars for resources such as conference rooms and equipment.

This chapter contains these sections:

- Calendar Server Administrators
- Calendar Server Users
- Calendar Server Data
- Calendar Server Architecture
- Calendar Server Access Control
- Calendar Server APIs and SDKs
- Single Sign-on (SSO)
- Calendar Server Deployment Configurations

Calendar Server Administrators

- Calendar Server Administrator (calmaster)
- Calendar Server User and Group (UNIX only)
- root (UNIX only)
- Windows NT Administrator

Calendar Server Administrator (calmaster)

The Calendar Server Administrator is the user name and associated password that can manage the Calendar Server. This user has administration privileges for the Calendar Server but not for the directory server. The default user ID is `calmaster`, but you can specify a different user during installation, if you wish. After installation you can specify a different user in `service.admin.calmaster.userid` in the `ics.conf` file.

The user ID you specify for the Calendar Server Administrator must be a valid user account in your directory server. If the Calendar Server Administrator user account does not exist in the directory server during installation, you must add it after installation. For example, if you accept the default `calmaster`, a user named `calmaster` must exist in your directory server.

Table 1-1 describes the Calendar Server Administrator configuration parameters in the `ics.conf` file.

Table 1-1 Calendar Server Administrator Configuration Parameters

Parameter	Description
`service.admin.calmaster.userid`	User ID of the person designated as the Calendar Server administrator. You must provide this required value during Calendar Server installation. The default is `"calmaster"`.
`service.admin.calmaster.cred`	Password of the user ID specified as the Calendar Server administrator. You must provide this required value during installation.
`caldb.calmaster`	Email address of the Calendar Server administrator. The default is `"root@localhost"`.
`service.admin.calmaster.overrides.accesscontrol`	Indicates whether the Calendar Server administrator can override access control.The default is `"no"`.

Table 1-1 Calendar Server Administrator Configuration Parameters *(Continued)*

Parameter	**Description**
`service.admin.calmaster.wcap.allowgetmodifyuserprefs`	Indicates whether the Calendar Server administrator can get and set user preferences using WCAP commands.The default is `"no"`.
`service.admin.ldap.enable`	Enables LDAP for user authentication of the user specified in `service.admin.calmaster.userid`. The default is `"yes"`.

Calendar Server User and Group (UNIX only)

On Solaris and other UNIX systems, these accounts are the UNIX user and group identity under which Calendar Server runs. iPlanet recommends that you use the default values, `icsuser` and `icsgroup`, which are automatically created by the installation program if they do not exist. The `icsuser` and `icsgroup` values are stored in the `local.serveruid` and `local.servergid` parameters, respectively, in the `ics.conf` file.

root (UNIX only)

On Solaris and other UNIX systems, you must log in as (or become) `root` (user ID = 0) to install, re-install, or upgrade the Calendar Server. To manage the Calendar Server using the command-line utilities, you must log in as (or become) `root` or an administrator such as `icsuser`.

Windows NT Administrator

To install and manage the Calendar Server on Windows NT systems, you must log in as an administrator with full administrator rights to the system.

Calendar Server Users

- Creation of Calendar Server Users
- Authentication of Calendar Server Users

Creation of Calendar Server Users

Calendar Server users are created either manually or automatically:

- Manually — A administrator can add users to the directory server using the directory server tools and then create the users' default calendars using the Calendar Server `cscal` utility. If a user doesn't already exist in the directory server, an administrator can create both the user and the calendar at the same time using the Calendar Server `csuser` utility.
- Automatically — If a user already exists in the directory server, the Calendar Server automatically creates a default calendar the first time the user logs in. The Calendar Server uses the user's user ID for the calendar ID (`calid`) of default calendar (unless a calendar by that name already exists).

  For example, suppose `TChang` exists in the directory server but is not yet enabled for calendaring (that is, does not have a default calendar). When `TChang` logs into the Calendar Server for the first time, it automatically enables `TChang` for calendaring and creates a default calendar with the `calid` `TChang`.

Authentication of Calendar Server Users

The Calendar Server stores and manages calendars, calendar properties, access control information, events, todos (tasks), and alarms. The Calendar Server, however, requires a directory service such as an LDAP server for user authentication and for the storage and retrieval of user preferences.

The Calendar Server default installation supports users defined and maintained in an LDAP directory, such as Netscape Directory Server. To allow access for users defined in a non-LDAP directory server, the Calendar Server also supports Calendar Server API (CSAPI) plug-ins.

For more information, see "Provisioning New Calendar Server Users," on page 42.

Calendar Server Data

- Calendar Server Data Format
- Calendar Server Format Encoding
- Calendar Groups
- Calendar Server Event Feeds
- Calendar Server Data Exchange and Alarms
- Calendar Server User Preferences

For information about access control for data, see “Calendar Server Access Control,” on page 35.

Calendar Server Data Format

The Calendar Server data format is modeled after RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar). The Calendar Server stores and manages calendars, calendar properties, access control information, events, todos (tasks), and alarms. The Calendar Server, however, does not manage user information. It requires a directory service to perform operations such as user authentication and the storage and retrieval of user preferences.

Calendar Server Format Encoding

The Calendar Server supports the following format encodings:

- SHTML (.shtml)—the default
- XML (.xml)—WCAP only
- iCalendar (.ical)—WCAP only

You can add other formats by developing your own XSL translations for the Calendar Express views and dialogs. Or by using CSAPI, you can develop a translator DLL or shared library for the WCAP protocol.

For details about CSAPI, see the *iPlanet Calendar Server Programmer’s Manual*.

Calendar Groups

A calendar group is a named list of individual calendars. Group calendars allow multiple calendars to be combined into a single calendar for display purposes. A user can, for example, have a calendar group made up of a private calendar, department calendar, and a company holidays calendar.

Users can create or subscribe to calendar groups and then view and modify the calendar group (subject to access control). They can then view the calendar group, rather than having to select a list of calendars each time they want to view them side-by-side or invite the calendar owners to an event.

Calendar Server Event Feeds

The Calendar Server supports event feeds, which allows users to import and export calendar data in either iCalendar (`.ics`) or XML (`.xml`) formats. End users can import and export data using Calendar Express. For information, see the Calendar Express online help.

Users can also subscribe to event or group calendars such as holiday schedules, convention center schedules, concert schedules, or any other schedule of events that might be of interest.

Calendar Server administrators can also import and export calendar data using the csimport and csexport command-line utilities.

Calendar Server Data Exchange and Alarms

Calendars can be referenced as links that can be embedded in email messages and Web pages. Users can then click a link to view a calendar, and as long as the calendar allows read access, are not required to log into the Calendar Server. For example, the following link specifies a resource room named `Auditorium`:

```
http://calendar.sesta.com:8080/?calid=Auditorium
```

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

Calendar Server User Preferences

The Calendar Server customizes the display of calendar information for each user according to attributes called user preferences. User preferences (as opposed to calendar preferences) refer to the user interface representation of calendar information. User preferences include such things as email address, user name, and preferred colors to use when rendering calendar information. For a list of preferences, see the `get_userprefs` and `set_userprefs` commands in the *iPlanet Calendar Server Programmer's Manual*.

Calendar Server Architecture

- Calendar Server Internal Subsystems
- Calendar Server Services
- Basic Calendar Server Configurations

For descriptions of more complex Calendar Server configurations, see "Calendar Server Deployment Configurations" on page 39.

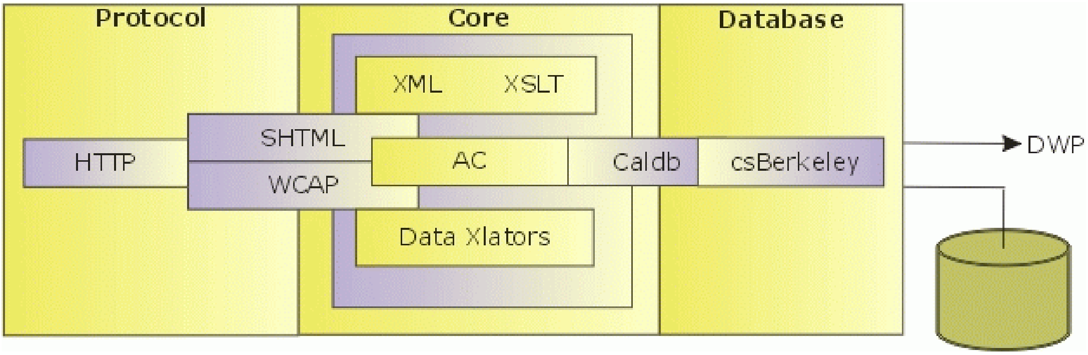
Calendar Server Internal Subsystems

The Calendar Server includes a collection of shared libraries that make up the following internal subsystems:

- Protocol Subsystem
- Core Subsystem
- Database Subsystem

Figure 1-1 shows the logical flow through these subsystems.

Figure 1-1 Calendar Server Internal Subsystems Logical Flow



Protocol Subsystem

Commands and requests enter through the HTTP protocol layer. This is a minimal HTTP server implementation, streamlined to support calendar requests.

Clients use either SHTML or Web Calendar Access Protocol (WCAP) to submit requests:

- SHTML is based on XML and XSLT specifications, which generate 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. This design results in fewer interactions between the client and server, which reduces the network traffic.

- WCAP is an open protocol that can perform all server commands (except for specific administrative commands). WCAP can be used by clients that need raw, unformatted calendar information, but it can also be used to obtain a JavaScript based user interface. WCAP commands (that is, commands that use the `.wcap` extension) can also request output as XML or iCalendar wrapped in HTML.

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

Core Subsystem

The Core subsystem includes the Access Control subsystem, the UI Generator subsystem (either SHTML using XML and XSLT or WCAP using data translators), Caldb subsystem, and any CSAPI plug-ins. The Core subsystem processes calendar requests and generates the desired UI output. The Core subsystem also handles user authentication including:

- Calendar Server API (CSAPI) authentication
- Proxy Authentication SDK (authSDK)
- Single Sign-on (SSO) authentication

Database Subsystem

The Database subsystem uses the Berkeley DB from Sleepycat Software (the database API is not public). The Database subsystem stores and retrieves calendar data to and from the database, including events, todos (tasks), and alarms. Calendar data is based on iCalendar format, and the schema used for Calendar Server data is a superset of the iCalendar standard. The Database subsystem returns data in a low-level format, and the Core UI generator (either SHTML or WCAP) then translates the low-level data into the desired output.

For a distributed calendar store, the Calendar Server use the Database Wire Protocol (DWP) to provide a networking capability. For more information, see “Distributed Database Service - csdwpd,” on page 31.

For information about managing the database, see Chapter 5, “Managing Calendar Server Databases,” which describes how to manage databases and calendar data using the csdb utility.

Calendar Server Services

- Administration Service - csadmind
- HTTP Service - cshttpd
- Notification Service - csnotifyd
- Event Notification Service (ENS) - enpd
- Distributed Database Service - csdwpd

These services run as processes (or daemons on UNIX systems) on a single machine or on multiple machines, depending on your configuration.

Administration Service - csadmind

The `csadmind` service is required for each instance of the Calendar Server. It provides a single point of authentication for administering the Calendar Server and provides most of the administration tools such as commands to start or stop a service, list or log out users, create or delete users or resources, fetch or store calendars, and fetch or reset counters. The `csadmind` service also manages alarm notifications, group scheduling requests, database checkpointing, and deadlock detection, as well as disk usage and server response monitoring.

HTTP Service - cshttpd

Since the Calendar Server uses HTTP as its primary transport, the `cshttpd` service listens for HTTP commands. The `cshttpd` service receives user commands and returns data to the caller, depending on the format of the incoming command:

- For a command received with the default `.shtml` extension, `cshttpd` returns data formatted in HTML.
- For a command received with the`.wcap` extension, `cshttpd` returns data formatted as calendar data in standard RFC2445 iCalendar format (`text/calendar`), XML format (`text/xml`), or JavaScript embedded in HTML (`text/js`).

Notification Service - csnotifyd

The `csnotifyd` service sends notifications of events and todos (tasks) using the Event Notification Service (ENS) as the broker for events. `csnotifyd` also subscribes to alarm events. When an alarm event occurs, `csnotifyd` sends an SMTP message reminder to the recipients.

Event Notification Service (ENS) - enpd

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

Distributed Database Service - csdwpd

The csdwpd service allows you to link multiple servers together within the same Calendar Server system to form a distributed calendar store. csdwpd can run in the background on any server where the Calendar Server is installed. It then accepts requests that follow the Database Wire Protocol (DWP) for calendar information. DWP is an internal protocol used to provide networking capability for the Calendar Server database.

The csdwpd service should be run only on a server that has a local calendar database and must provide network access to its calendar data from other Calendar Server installations.

You should use DWP only on a relatively fast network. If the network connection between the various databases is slow, DWP can degrade overall system performance.

Basic Calendar Server Configurations

The Calendar Server can be configured to fit the different needs of an organization. For example, it can run as a stand-alone server, or it can be configured with multiple instances, with the various Calendar Server services duplicated or split between the instances.

This section describes these basic Calendar Server configurations:

- Minimal Calendar Server Configuration
- Scalable Calendar Server Configuration

For descriptions of more complex Calendar Server configurations, see "Calendar Server Deployment Configurations" on page 39.

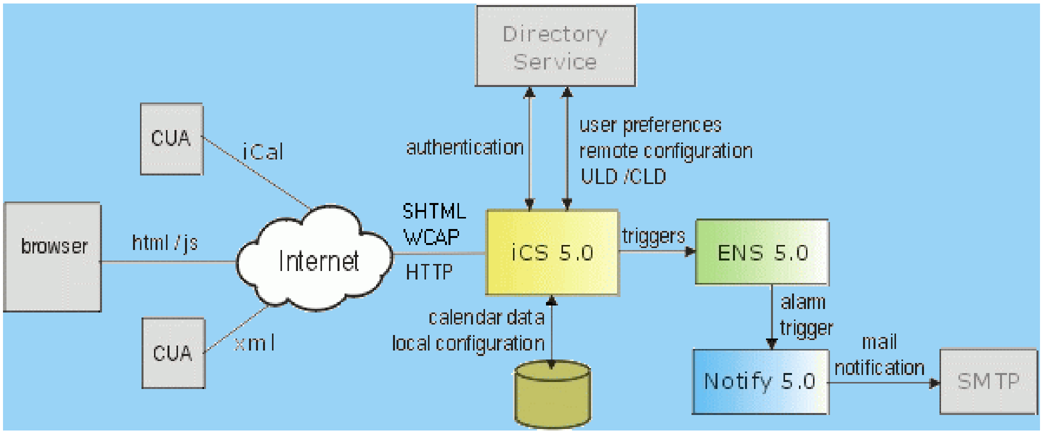
Minimal Calendar Server Configuration

Figure 1-2 shows a minimal Calendar Server configuration, including a:

- Single Calendar Server instance with support for event notifications. This configuration consists of the required Administration Service (`csadmind`), the HTTP service (`cshttpd`) to handle incoming SHTML and WCAP requests, and the Event Notification Services (ENS), `enpd` and `csnotifyd`.
- Directory service such as an LDAP server.

In Figure 1-2, CLD is the Calendar Lookup Database. CUA is a Calendar User Agent, which is an application that a calendar client uses to access the Calendar Server. iCal refers to RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar). The calendar database is local, so the Database Wire Protocol (DWP) service is not required.

Figure 1-2 Minimal Calendar Server Configuration



Scalable Calendar Server Configuration

The Calendar Server is scalable both vertically and horizontally. The Calendar Server can run in multiple processors on a single machine or on multiple machines. The Calendar Server consists of the `cshttpd`, `csadmind`, `csdwpd`, `csnotifyd`, and enpd. services, which can be run in different configurations to allow you great flexibility and scalability.

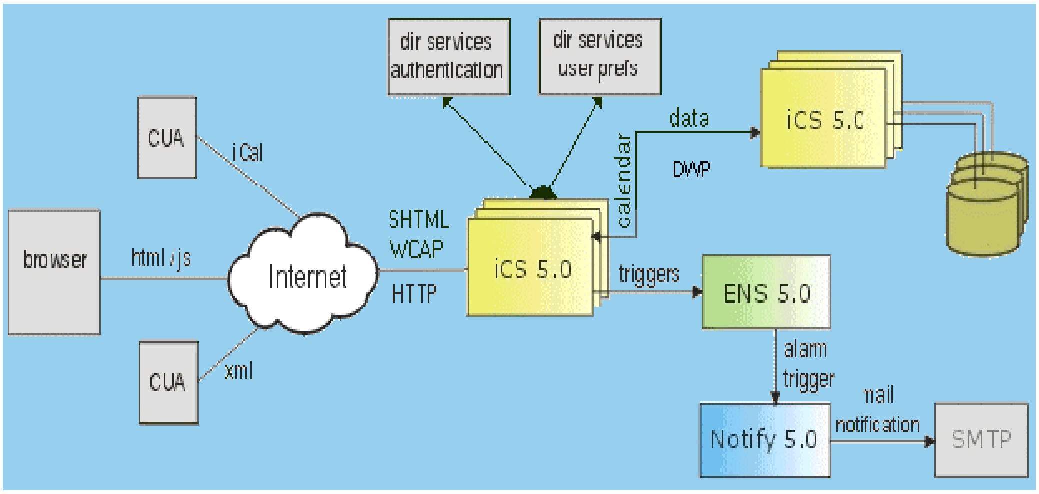
The Calendar Server supports horizontal scalability, which is spreading an installation over several machines. To achieve horizontal scalability, you can install various instances of the Calendar Server across your machines. The basic requirements for each system are:

- Each Calendar Server instance must have an Administration (`csadmind`) service. You can, however, install the Event Notification Services (ENS) `enpd` and `csnotifyd` on a separate instance without the `csadmind` service.
- All other Calendar Server services must be installed at least once, unless you have a single-instance installation using a local database. In this installation, the DWP (`csdwpd`) service is not required because the calendar database is local.

Figure 1-3 shows three Calendar Server HTTP front-end services using three Calendar Server database services. All six of these instances can run on separate machines.

This configuration uses the Database Wire Protocol (DWP), which is an internal protocol that provides networking capability for the Calendar Server database. DWP uses HTTP as its base, with an HTTP POST or GET command with a single binary MIME part that contains serialized binary database information. For more information, see “Distributed Database Service - csdwpd” on page 31.

Figure 1-3 Scalable Calendar Server Configuration



Calendar Server Access Control

The Calendar Server uses access control lists (ACLs) to determine the access control for calendars, calendar properties, and calendar components such as events and todos (tasks). An ACL consists of one or more access control entries (ACEs), which are strings that collectively apply to the same calendar or component. Each ACE in an ACL must be separated by a semicolon. For example:

- `jsmith^c^wd^g` consists of a single ACE.
- `@@o^a^r^g;@@o^c^wdeic^g;@^a^sf^g` consists of three ACEs.

Calendar Server access control features are:

- Calendar properties – Only the primary owner of a calendar or an administrator can delete a calendar or change the calendar’s properties, including its ACLs.
- Group scheduling – When an organizer schedules a group event, the Calendar Server uses ACLs to determine the access to the invitees’ calendars. For example, the organizer of an event might be able to view only the free/busy time for an attendee and not the attendee’s entire calendar.
- Other owners – The primary owner of a calendar can specify other owners of the calendar to act in their behalf. For example, a primary owner such as a vice president can designate an administrative assistant to schedule meetings and invite attendees, or to cancel or accept invitations to meetings scheduled by others.

End users set the access control for their calendars using Calendar Express. Calendar Server administrators can set access control configuration parameters in the `ics.conf` file and use the cscal, csresource, and csuser command-line utilities.

For more information, see Chapter 4, “Managing Calendar Server Access Control.”

Calendar Server APIs and SDKs

- Calendar Server API (CSAPI)
- Event Notification Service (ENS) API
- Proxy Authentication SDK (authSDK)
- Web Calendar Access Protocol (WCAP)

For detailed information about these APIs, see the *iPlanet Calendar Server Programmer's Manual*.

Calendar Server API (CSAPI)

The Calendar Server API (CSAPI) allows a programmer to customize functional areas of the Calendar Server such as user login authentication, access control, and calendar lookup.

For example, by default the Calendar Server uses entries in an LDAP server to authenticate users and to store user preferences. If you have another mechanism that is not based on LDAP, you can use CSAPI to implement your existing authentication and directory services and override the default Calendar Server mechanisms.

Event Notification Service (ENS) API

The Event Notification Service (ENS) is an alarm dispatcher that detects events on an alarm queue and sends notifications of these events to its subscribers. The ENS API allows programmers to modify publish-and-subscribe functions used by the Calendar Server to perform functions such as subscribe to events, unsubscribe to events, and notify a subscriber of events. The ENS APIs consists of these specific APIs:

- Published API
- Subscriber API
- Publish and Subscribe Dispatcher API

Proxy Authentication SDK (authSDK)

The Calendar Server provides the authSDK for user authentication. With authSDK, you can integrate an existing portal service with the Calendar Server, thus allowing users to access various applications without requiring re-authentication.

A connection established between the Calendar Server and the authSDK forms a trusted relationship. If a user logs in and successfully authenticates to the authSDK, the Calendar Server accepts the certificate generated by the proxy for its functions.

The authSDK consists of the following functions packaged in a DLL / shared-object library (`libicsexp10`) and a header (file, `expapi.h`):

- `CEXP_GenerateLoginURL` generates a URL with the valid session ID.
- `CEXP_GetVersion` generates the version ID string.
- `CEXP_Init` initializes the SDK.
- `CEXP_SetHttpPort` allows you to specify the port over which you will contact the Calendar Server.
- `CEXP_Shutdown` performs all shutdown procedures, including freeing memory and shutting down connections.

Web Calendar Access Protocol (WCAP)

iPlanet Calendar Server 5.x supports WCAP 2.0, a high-level, command-based protocol that allows communication with clients. WCAP commands, which use the `.wcap` extension, allow clients to get, modify, and delete calendar components, user preferences, calendar properties, and other calendar information such as time zones. WCAP elements such as times, strings, and parameters follow RFC2445, RFC2446, and RFC2447 specifications (unless otherwise noted).

WCAP returns output calendar data in an HTTP message in the following formats:

- Standard RFC2445 iCalendar format (`text/calendar`)
- XML format (`text/xml`)
- JavaScript embedded in HTML (`text/js`).

Using WCAP commands, a Calendar Server administrator who logs in using `login.wcap` has the following capabilities:

- To override the access control of WCAP commands

  The administrator can use WCAP commands to read (fetch), alter (store), or delete other user's calendars. For an administrator to have this privilege, the following parameter in the `ics.conf` file must be set to "yes":

  ```
  service.admin.calmaster.overrides.accesscontrol="yes"
  ```

- To retrieve and modify user preferences for any user

  The administrator can use `get_userprefs.wcap` and `set_userprefs.wcap` to retrieve and modify any user's preferences. For an administrator to have this privilege, the following parameter in the `ics.conf` file must be set to "yes":

  ```
  service.admin.calmaster.wcap.allowmodifyuserprefs="yes"
  ```

For more information, see the *iPlanet Calendar Server Programmer's Manual*.

Single Sign-on (SSO)

Single Sign-on (SSO) allows a user to authenticate once and then use multiple applications. For example, a user can log into Calendar Express and then use Messenger Express without authenticating again.

SSO is independent from other authentication mechanisms, session management, and resource access control. With SSO, applications form circles of trust that share cookies and accept each others' user authentication. Each verification authority stores a cookie that is understood by the other applications' verification routines. Each application can also have its own verification interface, if necessary.

SSO has the following requirements:

- The client browser must accept cookies.
- Each application must implement the verification protocol.
- All trusted applications must be in the same domain.
- To switch to a different identity, the user must restart the browser, because each browser session can support only one user ID.

To enable SSO between applications, you must configure each application. For information about configuring Calendar Server, see "Configuring Single Sign-on (SSO)," on page 61.

Calendar Server Deployment Configurations

In addition to the Minimal Calendar Server Configuration (Figure 1-2) and the Scalable Calendar Server Configuration (Figure 1-3), two other Calendar Server configurations are:

- Network Front End, Single Database Back End
- Multiple Front Ends, Multiple Database Back Ends

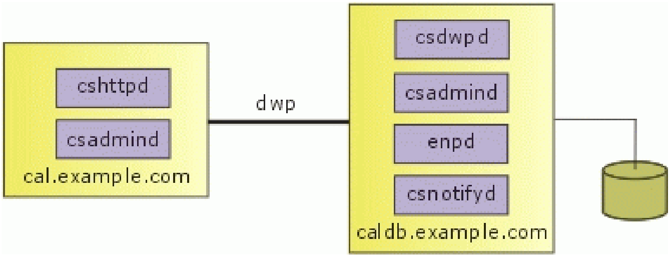
(Other Calendar Server configurations are also possible, depending on the requirements of the specific site.)

Network Front End, Single Database Back End

serviceFigure 1-4 shows a configuration where client browsers and applications connect to the Calendar Server through the HTTP (cshttp) service at the front end (cal.example.com). All requests for calendar data are then routed to the back end (caldb.example.com) using the DWP (csdwpd) service.

The front end requires only the HTTP (cshttp) and Administration (csadmind) services, because it does not perform any database processing. The back end does not require cshttpd, but it does require the Administration (csadmind) and DWP (csdwpd) services and the Event Notification Services (ENS), enpd and csnotifyd.

Figure 1-4 Network Front End, Database Back End



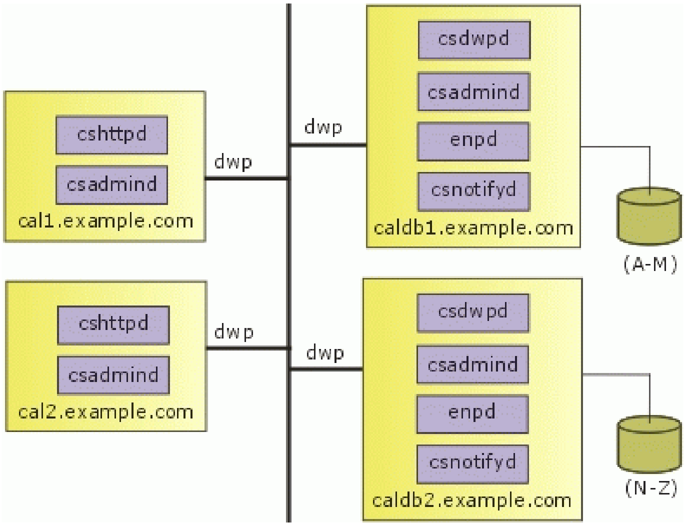
Multiple Front Ends, Multiple Database Back Ends

Figure 1-5 shows a configuration where clients are routed to one of the front-end HTTP (cshttp) services by a user-supplied mechanism. The session ID returned at login is valid only on the host system where the login occurred. All requests for this session ID must be routed to the same host, or the user is forced to log in again.

In this example, the database has been distributed, 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 the mapping between calendar IDs and the server where the calendar is located. (The default CSAPI implementation provided by the Calendar Server uses an algorithm to associate a calendar ID with a server name. See "Calendar Lookup Database Plug-in," on page 43 for more information.)

Each front end requires the HTTP (cshttp) and Administration (csadmind) services. All requests for calendar data are routed to the appropriate back end using the DWP (csdwpd) service. Each back end requires the Administration (csadmind) and DWP (csdwpd) services and the Event Notification Services (ENS), enpd and csnotifyd.

Figure 1-5 Multiple Front Ends, Multiple Database Back Ends



Chapter 2

Managing Calendar Server Users and Calendars

This chapter describes how to use the Calendar Server command-line utilities to provision and manage users and calendars, including both user calendars and resource calendars.

This chapter contains these sections:

- Provisioning New Calendar Server Users
- Managing Calendar Server Users
- Managing User Calendars
- Creating and Managing Resource Calendars

To provision and manage Calendar Server users and calendars, use the following command line-utilities:

- The csuser utility manages calendar user information stored in an LDAP directory server and the calendar database.
- The cscal utility manages calendars and their properties in the calendar database.
- The csresource utility manages calendars for resources such as conference rooms or equipment stored in the LDAP server and the calendar database.

To run the command-line utilities, you must log in as a user who has administrator rights to the system where the Calendar Server is running. For more information, see Chapter 7, “Calendar Server Command-Line Utilities.”

Provisioning New Calendar Server Users

This section provides the following information about provisioning new Calendar Server users:

- Directory Server Requirements
- Calendar Identifiers (calids)
- Checking if a User is Enabled for Calendaring
- Provisioning a New User
- Creating a New Calendar

Directory Server Requirements

The Calendar Server requires that a calendar user be stored in a directory server. The Calendar Server then uses the directory server for user authentication and for the storage and retrieval of user preferences.

The Calendar Server default installation supports users defined in an LDAP directory, such as the Netscape Directory Server. If your users are already stored in an LDAP directory, you can simply upgrade your directory server to Netscape Directory Server 4.12 or later, which supports the schema extensions that allow users to access the Calendar Server.

For information about installing and configuring Netscape Directory Server, see:

`http://docs.iplanet.com/docs/manuals/directory.html`

You can also modify your directory schema manually to allow your users to access the Calendar Server data. For information, see the *iPlanet Calendar Server Installation Guide*.

Calendar Identifiers (calids)

Each calendar in the Calendar Server database is identified by a unique calendar identifier (ID) or `calid`. The format for a `calid` is:

```
userid[:calendar]
```

where `userid` is the user ID and `calendar` is the calendar name.

Calendar IDs are case sensitive. For example, `JSMITH` is not equivalent to `jsmith`. (This distinction differs from email addresses, which are not case sensitive. For example, `jsmith@sesta.com` is equivalent to `JSMITH@SESTA.COM`.)

A `calid` cannot contain spaces but can include the following characters:

- Alphabetic (a-z, A-Z) and numeric (0-9) characters
- Special characters: period (.), underscore (_), hyphen or dash (-), at sign (@), apostrophe (‘), percent sign (%), slash (/), or exclamation point (!)

Examples of calendar IDs are `jsmith`, `jsmith:new-cal`, and `jsmith:private_calendar`.

Because the user ID is part of the `calid`, the user ID cannot contain spaces (for example, `j smith`). A user with a user ID that contains a space can log into the Calendar Server, but if the user then tries to create an event or task, the Calendar Server returns a Bad Request error.

Calendar Lookup Database Plug-in

If the calendar database is distributed over several servers (for example, see Figure 1-5), the Calendar Server uses the Calendar Lookup Database plug-in to determine the physical location of a calendar. The Database Wire Protocol (DWP) calls the Calendar Lookup service to fully qualify a calendar ID (`calid`). From the returned URL, DWP can then determine the location of the calendar, along with its access protocol.

The `caldb.cld.type` parameter in the `ics.conf` file specifies the type of Calendar Lookup Database plug-in to use:

- `local` (the default) specifies that all calendars are stored on the local machine where the Calendar Server is running, and no plug-in is loaded.
- `algorithmic` loads a plug-in that uses a regular expression to determine the server where a specified calendar ID is stored. This expression is specified with the parameter:

  `caldb.cld.server.[`*hostname*`].regexpr = "`*expression*`"`

  where *hostname* specifies the name of the server and *expression* identifies the calendar on the server. For more information, see “Configuring Database Wire Protocol (DWP),” on page 65.
- `directory` loads a plug-in that uses an LDAP directory schema entry to determine the physical machine where a specified calendar ID is stored.

Checking if a User is Enabled for Calendaring

To determine if a specific user exists in your directory server and is enabled to access Calendar Server data, use the `csuser` utility `check` command.

For example, to check if `JSmith` is enabled for calendaring:

```
csuser check JSmith
```

If the `check` command indicates that a user does not exist in your LDAP directory server, you must create a directory server entry for the user.

Provisioning a New User

The `csuser` utility can create an entry in the LDAP directory server and enable a user for calendaring. In the current release, however, iPlanet recommends that you provision user LDAP attributes using a directory service utility or a customized user provisioning tool.

For example, if you are using Netscape Directory Server, use the `ldapsearch` and `ldapmodify` utilities. For information about these utilities, see the Netscape Directory Server documentation on the following web site:

```
http://docs.iplanet.com/docs/manuals/
```

If the user already exists in your LDAP directory, a new calendar can be created for the user either manually or automatically:

- **Manually -** Use the `cscal` utility `create` command to create a new calendar for the user. See “Creating a New Calendar,” on page 45.
- **Automatically -** The first time a new user logs into the Calendar Server, the Calendar Server automatically creates a new default calendar for the user. The Calendar Server uses the user’s user ID for the calendar ID (`calid`) of the new calendar, unless a calendar by that name already exists.

  In the current release, the Calendar Server always creates a default calendar for a new user the first time the user logs into the Calendar Server, and you cannot disable this functionality.

  For example, if JSmith logs into the Calendar Server for the first time, the Calendar Server automatically creates a default calendar with `JSmith` as the `calid`.

  The Calendar Server also assigns the prefix `JSmith` to the `calid` of each subsequent calendar that JSmith creates. For example, if JSmith later creates a new calendar named `meetings`, the `calid` is `JSmith:meetings`.

If a user sends a request to another user who has not been enabled for calendaring (that is, the user does not have a default calendar), the Calendar Server returns the `Calendar not found` error to the user sending the request.

Creating a New Calendar

To create a new calendar, use the `cscal` utility `create` command. The user (user ID) must already exist in the directory server.

For example, to create a new calendar with the calendar ID (`calid`) `JSmith`:

```
cscal -o JSmith -n JohnSmithCalendar create JSmith
```

where:

- `-o JSmith` specifies the primary owner of the new calendar.
- `-n JohnSmithCalendar` specifies the viewable name for the new calendar.
- The default access control settings are defined by `calstore.calendar.default.acl` in the `ics.conf` file.

To create a calendar with the viewable name `Hobbies` that is owned by `JSmith` and uses the default access control settings for group scheduling:

```
cscal -n Hobbies -o JSmith create Personal
```

where:

- `-n Hobbies` specifies the viewable name of the calendar.
- `-o JSmith` specifies the user ID of the primary owner.
- `Personal` is used as the second part of the calendar ID (`calid`). For example: `JSmith:Personal`

The following example creates a new calendar similar to the previous example, but it also associates the calendar with the category named `sports`, enables double booking, and specifies `RJones` as another owner:

```
cscal -n Hobbies -o JSmith -g sports -k yes -y RJones create Personal
```

where:

- `-g sports` associates the calendar with a category named `sports`.
- `-y RJones` specifies another owner of the calendar.
- `-k yes` enables double booking.

The following example creates a calendar similar to the previous example, but it also sets specific access control settings for group scheduling:

```
cscal -n Hobbies -o JSmith -a "@@o^a^sfr^g" create Personal
```

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

Managing Calendar Server Users

After your users are provisioned, use the csuser utility to perform the following administrative tasks:

- Displaying User Information
- Disabling and Enabling a User
- Deleting a User
- Resetting a User's Attributes

Displaying User Information

To list all calendar users or to display the calendar attributes of a specified user, use the `csuser` utility `list` command.

For example, to display all users enabled for calendaring:

```
csuser list
```

To display all of the calendar attributes of a single user such as JSmith:

```
csuser -v list JSmith
```

Disabling and Enabling a User

To prevent a user from logging into iPlanet Calendar Server, use the `csuser` utility `disable` command. The `disable` command prohibits a user from accessing calendar data, but it does not remove the user's information from the directory server or the Calendar Server database.

For example, to disable JSmith from accessing the Calendar Server:

```
csuser disable JSmith
```

This command prevents JSmith from logging into the Calendar Server to access calendar data, but JSmith's data is not deleted from the calendar database. However, if JSmith is currently logged into the Calendar Server, JSmith retains access to calendar data until logging off.

To enable a user to access the Calendar Server and optionally to assign specific configuration settings such as a default calendar, use the `csuser` utility `enable` command.

For example, to enable JSmith to access (log into the Calendar Server) and to assign JSmith a default calendar:

```
csuser jsmith enable JSmith
```

Deleting a User

To delete a Calendar Server user, use the csuser utility `delete` command.

CAUTION The `csuser` utility `delete` command removes all of the user's Calendar Server information from the LDAP server. You can recover Calendar Server database information if the calendar database has been backed up. For more information, see Chapter 6, "Backing Up and Restoring Calendar Server Data."

However, you can recover the LDAP server information only if you have specifically backed it up.

For example, to delete JSmith from the Calendar Server:

```
csuser delete JSmith
```

Resetting a User's Attributes

To restore the default settings of all calendar LDAP attributes for a specific user, use the `csuser` utility `reset` command.

For example, to reset all calendar attributes of `JSmith` to the default configuration settings:

```
csuser reset JSmith
```

Managing User Calendars

After your user calendars are created, use the cscal utility to perform the following administrative tasks:

- Displaying Calendars
- Deleting a Calendar
- Disabling and Enabling a Calendar
- Modifying Calendar Properties
- Removing Properties From a Calendar
- Recovering a "Lost" Calendar

Displaying Calendars

To display all calendars, all calendars owned by a user, or the properties of a specific calendar, use the `cscal` utility `list` command.

For example, to list all calendars in the calendar database:

```
cscal list
```

To list all calendars owned by JSmith:

```
cscal -o JSmith list
```

To list all the properties of a calendar with the calendar ID `JSmith:meetings`:

```
cscal -v list JSmith:meetings
```

Deleting a Calendar

End users can unsubscribe from a calendar through Calendar Express, but an end user cannot delete a calendar from the Calendar Server database. Deleting a calendar must be done by an administrator who has administrative rights to the system.

To delete one or more calendars from the Calendar Server, use the `cscal` utility `delete` command. This utility deletes the calendar, but it does not delete the user from the directory server.

CAUTION The `delete` command removes all of the calendar information from the calendar database and cannot be undone. After you delete a calendar, you can recover the calendar data only if it was backed up. For more information, see Chapter 6, “Backing Up and Restoring Calendar Server Data.”

The `cscal` utility lets you delete a single calendar or multiple calendars.

For example, to delete a specific calendar with the calendar ID `JSmith:meetings`:

```
cscal delete JSmith:meetings
```

To delete all calendars whose primary owner is `JSmith`:

```
cscal -o JSmith delete
```

Disabling and Enabling a Calendar

To prevent users from accessing a calendar, use the `cscal` utility `disable` command. The `disable` command prohibits users from accessing the calendar, but it does not remove the information from the calendar database.

For example, to prevent users from accessing `JSmith:meetings`:

```
cscal disable JSmith:meetings
```

To enable a calendar to allow users to access the calendar, use the `cscal` utility `enable` command. For example, to enable calendar `JSmith:meetings` using the default configuration settings:

```
cscal enable JSmith:meetings
```

To enable the calendar `JSmith:meetings` but not allow double booking:

```
cscal -k no enable JSmith:meetings
```

Modifying Calendar Properties

To modify the properties of a calendar, use the `cscal` utility `modify` command.

For example, to change the group scheduling access control settings of `AllAdmins` and specify `RJones` as another owner:

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

where:

- `-a "@@o^c^wd^g"` grants owners write and delete access to the components (events and tasks) of `AllAdmins`.
- `-y RJones` specifies this user ID as another owner.

Removing Properties From a Calendar

To remove a property value from a calendar, use the `cscal` utility `modify` command and specify the option with two double quotes (`""`) as the value for the option.

For example, to remove a description from `JSmith:meetings`:

```
cscal -d "" modify JSmith:meetings
```

To remove all categories from `JSmith:meetings`:

```
cscal -g "" modify JSmith:meetings
```

To remove “other owners” from `JSmith:meetings`:

```
cscal -y "" modify JSmith:meetings
```

Recovering a “Lost” Calendar

If a user’s default calendar does not appear in the Calendar Express View tab or Calendars tab but still exists in the database, you can recover the calendar by updating the user’s LDAP entry with these attributes:

- `icsCalendar:`*default_calid*
- `icsSubscribed:`*default_calid*

where *default_calid* is the user’s default calendar ID (`calid`).

Creating and Managing Resource Calendars

A resource calendar is associated with a resource such as a meeting room or equipment such as a notebook computer or overhead projector.

To create and manage resource calendars, use the csresource utility. To run `csresource`, you must log in as a user who has administrator rights to the system where the Calendar Server is running.

This section describes how to create and manage resource calendars, including:

- Setting Resource Calendar Configuration Parameters
- Creating a Resource Calendar
- Displaying Resource Calendars and Attributes
- Modifying a Resource Calendar
- Disabling and Enabling a Resource Calendar
- Deleting a Resource Calendar
- Linking to a Resource Calendar

Setting Resource Calendar Configuration Parameters

Table 2-1 lists the resource calendar configuration parameters in the `ics.conf` file.

Table 2-1 Resource Calendar Configuration Parameters in `ics.conf`

Parameter	Description
`resource.default.acl`	This parameter determines the default access control permissions used when a resource calendar is created. The default permissions are specified by the following Access Control List (ACL): `"@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g"` This ACL grants all calendar users read, schedule, and free/busy access to the calendar, including both components and properties. To change the permissions for a resource, use the `-a` option when you create the calendar using the `csresource` utility `create` command.

Table 2-1 Resource Calendar Configuration Parameters in `ics.conf` *(Continued)*

Parameter	Description
`resource.allow.doublebook`	This parameter determines if a resource calendar allows double-booking. Double-booking allows a resource calendar to have more than one event scheduled for the same time. The default is"`no`" — Do not allow double-booking. To allow double-booking for a resource calendar, use the `-k` option when you create the calendar using the `csresource` utility `create` command.

The default values shown in Table 2-1 apply to new resource calendars, but you can change these default values by editing the `ics.conf` file. For more information, see "Editing the ics.conf Configuration File," on page 155.

Creating a Resource Calendar

The Calendar Server does not automatically create resource calendars, so you must use the `csresource` utility `create` command to manually create each resource calendar required at your site. This command creates an entry in the LDAP directory server and calendar database for the new calendar. Several considerations for creating calendars are:

- By default, the Calendar Server does not allow double booking for a resource calendar (`resource.allow.doublebook` parameter). This default prevents scheduling conflicts for resources such as rooms and equipment. However, if you want to allow double booking for a resource calendar, set the `csresource` `-k` option to "yes" when you create the calendar.
- To control who can schedule a specific resource, consider limiting the users who have write access to the resource calendar. For example, you might want to allow only certain users to schedule meeting rooms or reserve equipment.

For example, to create a resource calendar with the calendar ID `aud100`, viewable name `Auditorium` (LDAP `cn` attribute), and the default settings shown in Table 2-1:

```
csresource -c aud100 create Auditorium
```

The following command performs the same action as the previous example, but the `-k` option allows double booking on the calendar, the `-o` option specifies `bkamdar` as the owner of the calendar, and the `-y` option specifies `jsmith` as another owner:

```
csresource -c aud100 -k yes -o bkamdar -y jsmith create Auditorium
```

If you do not specify an owner for a resource calendar, the value is taken from the service.admin.calmaster.userid parameter in the ics.conf file.

Displaying Resource Calendars and Attributes

To display a resource calendar, use the csresource utility list command.

For example, to display a list of all Calendar Server resource calendars and their corresponding LDAP attributes:

```
csresource list
```

To display a list of all LDAP attributes for a specific resource calendar named Auditorium:

```
csresource -v list Auditorium
```

Modifying a Resource Calendar

To modify a resource calendar, use the cscal utility modify command (csresource does not have a modify command).

For example, to set the owner as tchang and add another owner named mwong to the resource calendar named Auditorium:

```
cscal -o tchang -y mwong modify aud100
```

In this example, the cscal utility requires the calid (aud100) rather than the calendar name (Auditorium).

Disabling and Enabling a Resource Calendar

You might need to disable a resource calendar to prevent users from scheduling events. For example, a conference room might be unavailable during remodeling, or an overhead project might be out for repair.

To disable or enable a resource calendar, use the `csresource` utility `enable` or `disable` command.

For example, to disable the resource calendar named `Auditorium`:

```
csresource disable Auditorium
```

Then, to enable the resource calendar later:

```
csresource enable Auditorium
```

Deleting a Resource Calendar

To delete a resource calendar, use the `csresource` utility `delete` command.

For example, to delete the Auditorium resource calendar:

```
csresource delete Auditorium
```

The Calendar Server displays the following message:

```
Do you really want to delete this resource (y/n)?
```

Enter “y” to delete the calendar or “n” to cancel the operation.

If you enter “y”, the Calendar Server deletes the calendar and displays a message that it has been deleted.

Linking to a Resource Calendar

You can link to a resource calendar in email messages and web pages using a URL. Users can use the URL to anonymously view the resource calendar without having to log in to the Calendar Server (provided the calendar allows read access).

For example, the following link specifies a resource calendar for an overhead projector with the calendar ID (`calid`) `overhead_projector10`:

```
http://calendar.sesta.com:8080/?calid=overhead_projector10
```

Chapter 3

Managing the Calendar Server

This chapter describes how to manage and configure iPlanet Calendar Server.

This chapter contains these sections:

- Starting and Stopping the Calendar Server
- Configuring Calendar Server Timeout Values
- Configuring Single Sign-on (SSO)
- Configuring Database Wire Protocol (DWP)
- Managing LDAP Attributes
- Managing the Group Scheduling Engine (GSE) Queue
- Monitoring the Calendar Server
- Pinging the Calendar Server
- Refreshing the Calendar Server Configuration

You manage the Calendar Server by running command-line utilities and by editing the `ics.conf` configuration file.

To run the command-line utilities, you must log in as a user who has administrative rights to the system where the Calendar Server is running.

For more information, see Chapter 7, “Calendar Server Command-Line Utilities” and Chapter 8, “Calendar Server Configuration.”

Starting and Stopping the Calendar Server

You can start and stop the Calendar Server as follows:

- On Solaris, other UNIX, and Windows NT systems, use the `start-cal` and `stop-cal` commands. See "Using the start-cal and stop-cal Commands," on page 56.
- On Windows NT systems, you can also use the Control Panel Services. See "Using the Windows NT Control Panel," on page 57.

NOTE The Calendar Server provides the `csstart` and `csstop` utilities only to provide compatibility with earlier releases. iPlanet recommends that you use the `start-cal` and `stop-cal` commands to start and stop the Calendar Server.

Using the start-cal and stop-cal Commands

The `start-cal` and `stop-cal` utilities are located in the *server-root*`/cal/bin` directory. You must run these utilities on the local machine where the Calendar Server is installed. For problems that might occur, see "Troubleshooting the start-cal and stop-cal Commands," on page 57.

The `start-cal` command starts the Calendar Server services in the following order:

1. `enpd` — Event Notification Service (ENS)
2. `csnotifyd` — Notification Service
3. `csadmind` — Administration Service
4. `csdwpd` — Distributed Database Service (started only with a remote Calendar Server database configuration)
5. `cshttpd` — HTTP Service

For a description of these services, see "Calendar Server Services," on page 30.

To start the Calendar Server using the `start-cal` command:

1. Log in as a user who has administrative rights to the system.
2. Change to the *server-root*`/cal/bin` directory. For example on Solaris systems:

   ```
   cd /opt/SUNWics5/cal/bin
   ```

3. Start the Calendar Server:

   ```
   ./start-cal
   ```

To stop the Calendar Server using the `stop-cal` command:

1. Log in as a user who has administrative rights to the system where the Calendar Server is running.
2. Change to the *server-root*`/cal/bin` directory. For example on Solaris systems:

   ```
   cd /opt/SUNWics5/cal/bin
   ```

3. Stop the Calendar Server:

   ```
   ./stop-cal
   ```

Using the Windows NT Control Panel

On Windows NT systems, use the Services dialog box from the Control Panel.

To start and stop the Calendar Server using the Windows NT Control Panel:

1. Log in as a user who has administrative rights to the Windows NT system.
2. Display the Services dialog box from the Control Panel:

   ```
   Start>Settings>Control Panel>Services
   ```

3. Under Service, click the specific Calendar Server service (Admin, DWP, HTTP, ENS, or Notification) and then click Start or Stop.

For more information, refer to the Windows NT online Help.

Troubleshooting the start-cal and stop-cal Commands

When you are starting and stopping the Calendar Server, the following problems might occur:

- The `start-cal` command does not start all of the Calendar Server processes. For example, `start-cal` might start the `enpd`, `csnotifyd`, and `csadmind` processes but not `cshttpd`. In this situation, you must stop all of the Calendar Server processes before you try to restart the Calendar Server.

- The `stop-cal` command does not stop all of the Calendar Server processes. For example, `stop-cal` might stop the `cshttpd` parent process but not any `cshttpd` child processes. In this situation, you must stop the remaining Calendar Server processes.

To stop Calendar Server processes on a Windows NT system:

1. Log in as a user who has administrative rights to the system where the Calendar Server is running.

2. Use the Task Manager to identify and then stop any remaining Calendar Server processes.

To stop Calendar Server processes on Solaris and other UNIX systems:

1. Log in as a user who has administrative rights to the system where the Calendar Server is running.

2. Determine the process ID (PID) of the remaining Calendar Server processes by entering a `ps` command for each service:

   `ps -elf | grep` *cs-process*

   where *cs-process* is `enpd`, `csnotifyd`, `csdwpd`, `csadmind`, or `cshttpd`. For example:

   ```
   ps -elf | grep cshttpd
   ```

3. Using the PID of each process that is still running, enter a `pkill -15` command to kill the process. For example:

   ```
   pkill -15 9875
   ```

4. Enter each `ps` command again to make sure that all Calendar Server processes are stopped.

   If a Calendar Server process is still running, enter a `pkill -9` command to kill it. For example:

   ```
   pkill -9 9875
   ```

CAUTION After you have stopped all of the Calendar Server processes and before you restart the Calendar Server, consider running the `csdb` utility `check` command to check for any calendar database corruption that might have occurred.

For information about the `check` command, see “Checking and Rebuilding a Calendar Database,” on page 91.

Configuring Calendar Server Timeout Values

- Configuring Timeout Values for csadmind
- Configuring HTTP Timeout Values for End Users

For information about editing `ics.conf` parameters, see "Editing the ics.conf Configuration File," on page 155.

Configuring Timeout Values for csadmind

Table 3-1 describes the Calendar Server timeout parameters in the `ics.conf` file used by the Administration (`csadmin`) service.

Table 3-1 HTTP Timeout Values for the Administration Service (`csadmin`)

Parameter	Description
`service.admin.idletimeout`	Specifies the number of seconds the `csadmind` service waits before timing out an idle HTTP connection. The default is 120 seconds (2 minutes).
`service.admin.resourcetimeout`	Specifies the number of seconds the `csadmind` service waits before timing out an HTTP session for a resource calendar. The default is 900 seconds (15 minutes).
`service.admin.sessiontimeout`	Specifies the number of seconds the `csadmind` service waits before timing out an HTTP session. The default is 1800 seconds (30 minutes).

Configuring HTTP Timeout Values for End Users

Table 3-2 describes the Calendar Server HTTP timeout parameters in the `ics.conf` file that apply to end users.

Table 3-2 HTTP Timeout Values in `ics.conf` for End Users (`cshttpd` Service)

Parameter	Description
`service.http.idletimeout`	Specifies the number of seconds the `cshttpd` service waits before timing out an idle HTTP connection. The default is 120 seconds (2 minutes).
`service.http.resourcetimeout`	Specifies the number of seconds the `cshttpd` service waits before timing out an HTTP session for a resource calendar. The default is 900 seconds (15 minutes).
`service.http.sessiontimeout`	Specifies the number of seconds the `cshttpd` service waits before timing out an HTTP session. The default is 1800 seconds (30 minutes).

Configuring Single Sign-on (SSO)

Single Sign-on (SSO) allows a user to authenticate once and then use multiple trusted applications without having to authenticate again. For example, a user can log in to Messenger Express and then use Calendar Express without authenticating again, if SSO is enabled for both applications.

When configuring SSO, consider these points:

- Each trusted application must be configured for SSO.
- SSO does not work correctly if the `default.html` page is in your browser's cache. Before using SSO, be sure to reload the `default.html` page in your browser. For example, in Netscape Navigator, hold down the Shift key and then click Reload.
- SSO works only for bare URLs. For example, SSO works for `http://servername` but not for URLs such as `http://servername/command.shtml?view`.

The following example shows an SSO configuration for the Calendar Server (Calendar Express) and Messaging Server (Messenger Express) for the `sesta.com` domain.

To configure Single Sign-on (SSO):

1. Log in as a user with administrator privileges.
2. Stop the Calendar Server and Messaging Server.
3. Edit the Calendar Server `ics.conf` file as described in Table 3-3. (For a description of the Calendar Server SSO configuration parameters, see "Single Sign-on (SSO) Configuration," on page 175.)

Table 3-3 Calendar Server Configuration for SSO

Parameter	Description
`sso.enable = "1"`	This parameter must be set to`"1"` (the default) to enable SSO. `"0"` disables SSO.
`sso.appid = "ics50"`	This parameter specifies the unique application ID for the specific Calendar Server installation. Each trusted application must also have a unique application ID. The default is `"ics50"`.

Table 3-3 Calendar Server Configuration for SSO *(Continued)*

Parameter	**Description**
`sso.appprefix = "ssogrp1"`	This parameter specifies the prefix value to be used for formatting SSO cookies. The same value must be used by all trusted applications, because only SSO cookies with this prefix will be recognized by the Calendar Server. The default is `"ssogrp1"`.
`sso.cookiedomain = ".sesta.com"`	This parameter causes the browser to send a cookie only to servers in the specified domain. The value must begin with a period (.)
`sso.singlesignoff = "true"`	A value of `"true"` (the default) clears all SSO cookies on the client with prefix values matching the value configured in `sso.appprefix` when the client logs out.
`sso.userdomain = "sesta.com"`	This parameter sets the domain used as part of the user's SSO authentication.

`sso.`*appid*`.url = "`*verifyurl*`"`
For example:
`sso.ics50.url = "http://sesta.com:8883/VerifySSO?"`
`sso.msg50.url = "http://sesta.com:8882/VerifySSO?"`
	This parameter sets the verify URL values for peer SSO hosts for the Calendar Server configuration. One parameter is required for each trusted peer SSO host. The parameter includes the:
	• Application ID (*appid*) identifies each peer SSO host whose SSO cookies are to be honored
	• Verify URL (`"`*verifyurl*`"`) includes the host URL, host port number, and `VerifySSO?` (including the ending ?).
	In this example, the Calendar Server application ID is `ics50`, the host URL is `sesta.com`, and the port is 8883.
	The Messenger Express application ID is `msg50`, the host URL is `sesta.com`, and the port is 8882.

4. Use `configutil` to set the Messaging Server configuration parameters as shown in Table 3-4. You do not have to enclose these parameters in double quotation marks (").

Table 3-4 Messaging Server Configuration for SSO

Parameter	Description
`local.webmail.sso.enable = 1`	This parameter must be set to a non-zero value to enable SSO.
`local.webmail.sso.prefix = ssogrp1`	This parameter specifies a prefix used when formatting SSO cookies set by the HTTP server.
`local.webmail.sso.id = msg50`	This parameter specifies the unique application ID (`msg50`) for the Messaging Server. Each trusted application must also have a unique application ID.
`local.webmail.sso.cookiedomain = .sesta.com`	This parameter specifies the cookie domain value of all SSO cookies set by the HTTP server.
`local.webmail.sso.singlesignoff = 1`	A non-zero value clears all SSO cookies on the client with prefix values matching the value configured in `local.webmail.sso.prefix` when the client logs out.

`sso.`*appid*`.url = "`*verifyurl*`"`
For example:
`local.sso.ics50.verifyurl = http://sesta.com:8883/VerifySSO?`
`local.sso.msg50.verifyurl = http://sesta.com:8882/VerifySSO?`
	This parameter sets the verify URL values for peer SSO hosts for the Messaging Server configuration. One parameter is required for each trusted peer SSO host. The parameter includes the:
	• Application ID (*appid*) identifies each peer SSO host whose SSO cookies are to be honored
	• Verify URL ("*verifyurl*") includes the host URL, host port number, and `VerifySSO?` (including the ending `?`).
	In this example, the Messaging Server application ID is `msg50`, the host URL is `sesta.com`, and the port is 8882.
	The Calendar Server application ID is `ics50`, the host URL is `sesta.com`, and the port is 8883.

5. Restart the Calendar Server and Messaging Server to update the configurations.

   For more information, see “Starting and Stopping the Calendar Server,” on page 56. For information about the Messaging Server, see the *iPlanet Messaging Server Administrator’s Guide*.

Configuring Database Wire Protocol (DWP)

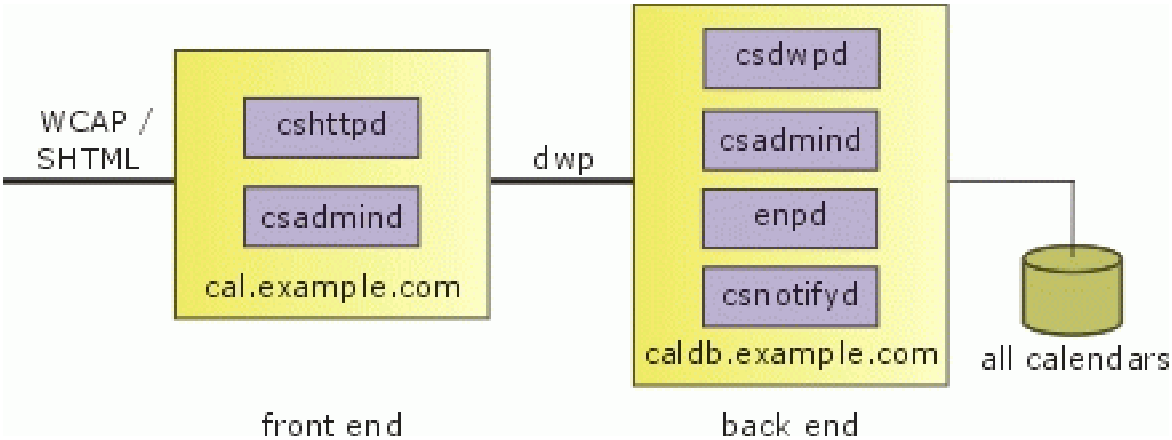
DWP is a proprietary protocol used by the Calendar Server to perform calendar database operations over a network. DWP uses HTTP as its transport mechanism and includes a subset of the calendar database API.

If the calendar database resides on the local server, the Calendar Server Database subsystem uses the `calid` to access a calendar in the database. However, if the calendar database is located over a network (such as, on a back-end server), the Calendar Server must use the Calendar Lookup Database plug-in to determine the network address of the server where a calendar actually resides. The request is then forwarded to the DWP (`csdwpd`) service on the other server for processing.

Front-End Machine and a Back-End Server

Figure 3-1 shows a Calendar Server configuration with a single front-end machine and a back-end server. To use DWP, the front-end machine and back-end server must be running the same operating system and the same version of the Calendar Server (such as 5.1).

Figure 3-1 Calendar Server Configuration for a Front-End Machine and a Back-End Server



In the configuration shown in Figure 3-1, the front-end machine performs these functions:

- Handles client requests for calendar data residing over a network
- Makes requests to the back-end server for calendar data
- Transforms the calendar data first into an XML document tree
- Applies an XSL file to the XML document tree to produce HTML
- Sends the HTML back to the client

The back-end server performs these functions:

- Handles all calendar database requests
- Processes the group scheduling engine (GSE) request queue
- Monitors the reminder queue
- Sends reminders using the `ens` and `csnotifyd` services

To configure DWP, you must set `ics.conf` parameters on both the front-end machine and back-end servers.

To Configure DWP Parameters on a Front-End Machine:

1. On the front-end machine, log in as a user with administrator privileges.
2. Change to the *server-root*`/cal/bin` directory where the Calendar Server command-line utilities are located and stop the Calendar Server using the `stop-cal` command.
3. Change to the *server-root*`/cal/bin/config/` directory and edit the `ics.conf` parameters shown in Table 3-5. (To configure multiple front-end machines that point to a single back-end server, see "Multiple Front-End Machines" on page 69.)

Table 3-5 DWP Configuration Parameters for a Front-End Machine

Parameter	Description
`service.http.enable = "yes"`	Set to "yes" (which is the default) to indicate that the local configuration requires the `cshttpd` service.
`csapi.plugin.calendarlookup = "y"`	Set to "y" (yes) to cause the CSAPI subsystem to load the Calendar Lookup Database plug-in.

Table 3-5 DWP Configuration Parameters for a Front-End Machine *(Continued)*

Parameter	Description
`csapi.plugin.calendarlookup.name = "*"`	This parameter specifies the plug-in name. In the current release, Set to `"*"` (which is the default) to cause the CSAPI subsystem to load the Calendar Lookup Database plug-in.
`caldb.cld.type = "algorithmic"`	This parameter specifies the Calendar Lookup Database plug-in to use. The default is `"local"`, but for a successful connection to the *hostname* server, set to `"algorithmic"`.
`caldb.dwp.server.`*hostname*`.ip = "`*hostname*`"`	This parameter specifies the name (network address) of the back-end server where the DWP service is running. It is read by all services that use DWP, and at startup, each service tries to establish contact with the DWP service. For example, if the server name is `sesta`, the parameter is: `caldb.dwp.server.sesta.ip = "sesta"`
`caldb.dwp.server.`*hostname*`.port = "9779"`	This parameter specifies the port on which the DWP (`csdwpd`) service listens. The default is `"9779"`. It is used with the `caldb.dwp.server.`*hostname*`.ip` value. The *hostname* part must be the same as in the `caldb.dwp.server.`*hostname*`.ip`. For example: `caldb.dwp.server.sesta.port = "9779"`
`caldb.cld.server.`*hostname*`.regexpr = "`*expression*`"`	If `caldb.cld.type` is `"algorithmic"`, this parameter specifies the expression used by the Calendar Lookup Database plug-in to determine the physical server where a specified calendar ID is stored. For example: `caldb.cld.server.sesta.regexpr = "^[^\n]"` matches all calendar IDs on the `sesta` server.

4. Change to the *server-root*`/cal/bin` directory where the Calendar Server command-line utilities are located and restart the Calendar Server using the `start-cal` command.

   On the front-end machine, the `cshttpd` and `csadmind` services are required.

To Configure DWP Parameters on a Back-End Server:

1. On the back-end server, log in as a user with administrator privileges.
2. Change to the *server-root*`/cal/bin` directory where the Calendar Server command-line utilities are located and stop the Calendar Server using the `stop-cal` command.

3. Change to the *server-root*/`cal/bin/config/` directory and edit the `ics.conf` parameters shown in Table 3-6.

Table 3-6 DWP Configuration Parameters for a Single Back-End Server

Parameter	Description
`service.dwp.enable = "yes"`	Enables the DWP (`csdwpd`) service. Causes the Calendar Server to start the `csdwpd` service when starting other services. Also, causes `csdwpd` service to listen on the port specified in `service.dwp.port`. Default is “no” and must be reset to “yes”.
`service.dwp.port = "9779"`	Specifies the port on which the `csdwpd` service listens. Default is “9779”.

`service.notify.enable = "yes"`
`service.admin.enable = "yes"`
`service.ens.enable = "yes"`
	Enables each service on the back-end server. Must be set to “yes” (which is the default) to enable respective service.

4. Change to the *server-root*/`cal/bin` directory where the Calendar Server command-line utilities are located and restart the Calendar Server using the `start-cal` command.

   On the back-end server, the `csdwpd`, `csadmind`, `enpd`, and `csnotifyd` services are required.

You can now access the Calendar Server database on the back-end server from the front-end machine using Calendar Express. Here’s how it works:

When the `cshttpd` service starts on the front-end machine, it initializes the Database subsystem. It loads the `caldb.dwp.server.`*host*`.ip` and `caldb.dwp.server.`*host*`.port` parameters from the `ics.conf` file and attempts to contact the back-end server using the host IP address and port values. If the contact is successful, the `cshttpd` service creates a pool of connections with the `csdwpd` service on the back-end server that are to used exclusively for DWP transactions.

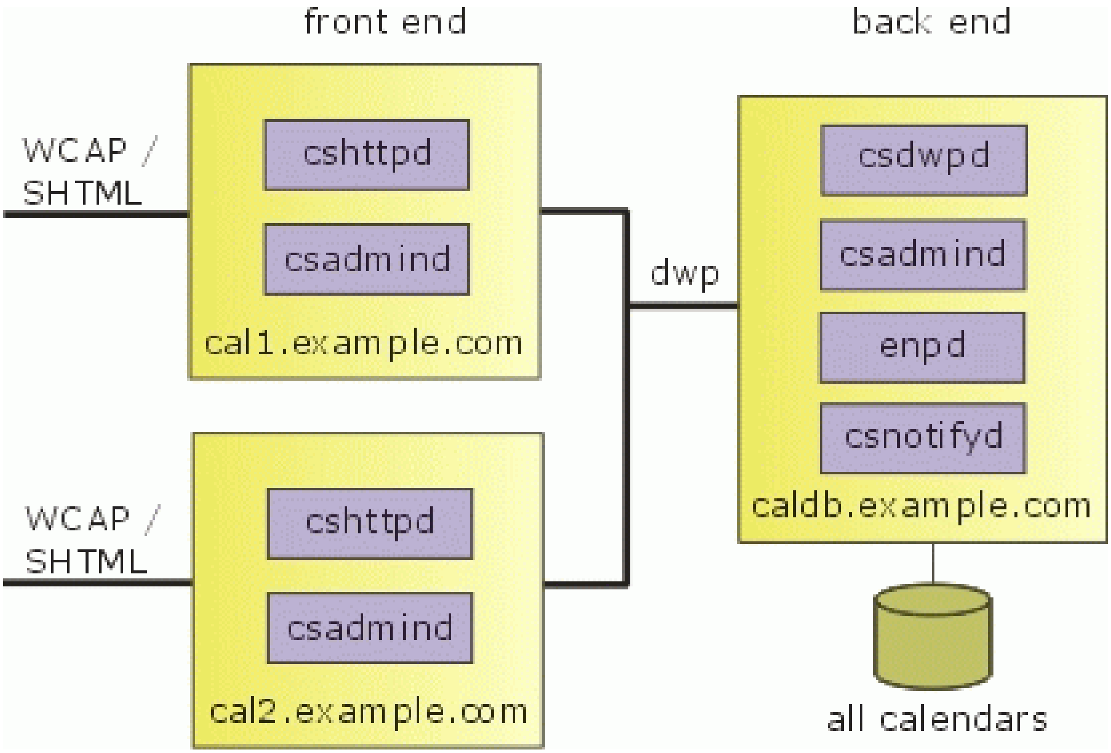
Initially, the size of this connection pool is set to the value of the `caldb.dwp.initconns`. parameter, but the pool can grow to the maximum value specified by the `caldb.dwp.maxcons` parameter. Each connection in the pool is an HTTP/1.1 persistent connection, and if any connection fails, an error message is written to the log file.

If a DWP connection between the front-end machine and back-end server is broken (for example, the back-end server is restarted), the front-end machine tries to reconnect with the back-end server. Requests for calendar data fail while the DWP connection is broken, and data is not available until the connection is restored.

Multiple Front-End Machines

Figure 3-2 shows a Calendar Server configuration with multiple front-end machines and a single back-end server. The configuration parameters for both front-end machines (cal1.example.com and cal2.example.com) are identical and are described in “To Configure DWP Parameters on a Front-End Machine:” on page 66. You can add other front-end machines as well to distribute the front-end load.

Figure 3-2 Calendar Server Configuration for Multiple Front-End Machines and a Back-End Server



Managing LDAP Attributes

To manage the LDAP attributes used by Calendar Server, use the csattribute utility.

Listing LDAP Attributes

To list LDAP attributes for a user or resource, use the `csattribute` utility `add` command. For example, to list the LDAP attributes for the user `TChang`:

```
csattribute list TChang
```

Adding an LDAP Attribute

To add an attribute to the LDAP server, use the `csattribute` utility `add` command. For example, to add the LDAP attribute `icsCalendar` with the value `Conference_Schedule` to the user `TChang`:

```
csattribute -a icsCalendar=Conference_Schedule add TChang
```

Deleting an LDAP Attribute

To delete an attribute to the LDAP server, use the `csattribute` utility `delete` command. For example, to delete the LDAP attribute `icsCalendar` from `TChang`:

```
csattribute -a icsCalendar delete TChang
```

Managing the Group Scheduling Engine (GSE) Queue

Group scheduling allows a Calendar Server user to create an event such as a meeting and then invite other attendees. By using the free/busy lookup feature, the user can determine when an invitee is actually available for an event.

If an attendee is on the same Calendar Server, the event is scheduled in the attendee’s calendar. If an attendee is not on the same Calendar Server, the invitation is sent via email. The attendee can then accept or decline the invitation.

Calendar Server users can also compare a group schedule by viewing attendees' calendars side-by-side.

To manage entries in the GSE queue, use the csschedule utility. You must run csschedule on the local machine where the Calendar Server is installed.

Listing Entries in the GSE Queue

To list entries in the GSE queue, use the csschedule utility list command. For example, to list all entries in the GSE queue:

```
csschedule list
```

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

```
csschedule -c 10 list
```

To list all entries in the GSE queue for a calendar with the calid Holiday_Schedule:

```
csschedule -v list Holiday_Schedule
```

Deleting Entries in the GSE Queue

To delete entries in the GSE queue, use the csschedule utility delete command. For example, to delete all entries in the GSE queue:

```
csschedule -v delete
```

To delete the entry in the GSE queue for calendar calA with the first schedule time of 13:30:45 on 11/30/2001, an offset number of 1, the unique identifier 1111, the recurrence ID 0, and the sequence number 0:

```
csschedule -v -t 20011130T133045Z -o 1 -u 1111 -r 0 -n 0 delete calA
```

Monitoring the Calendar Server

To monitor Calendar Server activity, use the csstats and cstool utilities. This section describes the following tasks:

- Listing Counter Statistics
- Monitoring the Calendar Server Log Files

Listing Counter Statistics

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

- 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 more information about Calendar Server counter statistics, see “Counters Configuration (counter.conf) File” on page 187.

To list statistical information, use the `csstats` utility `list` command. For example, to display basic information about the counter objects and the types available:

```
csstats list
```

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

```
csstats list http
```

To list statistics about the `wcapstat` counter object every 10 seconds for one hour:

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

Monitoring the Calendar Server Log Files

Each Calendar Server service writes status information to its own log file. Each log file is named after its associated service name, as shown in Table 3-7:

Table 3-7 Calendar Server Log Files

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

Calendar Server log files are stored in the default log directory:

- On Solaris systems:

  `/var/opt/SUNWics5/logs`
- On UNIX systems other than Solaris:

  `/var/opt/iPlanet/CalendarServer5/logs`
- On Windows NT systems:

  `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
```

Log Event Severity Levels

The Calendar Server provides eight levels of severity for events reported to the log files as described in Table 3-8.

Table 3-8 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 condition.
WARNING	Warning condition.
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 timestamp, 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 the Calendar Server reports to the log files by modifying certain configuration settings in the `ics.conf` file. For information, see "Calendar Log Information Configuration" on page 164.

You should inspect the log files on a regular basis for EMERGENCY, ALERT, CRITICAL, ERROR, and WARNING level errors and, if found, examine the events 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 might be asked to provide the log files for help in resolving problems.

Pinging the Calendar Server

To verify that a Calendar Server service is listening on a specified port number, use the cstool utility `ping` command. Pinging a service does not verify that a service is actually running but indicates if it can accept a socket connection.

The Calendar Server service options are:

- `http` — HTTP Service (`cshttpd`)
- `admin` — Administration Service (`csadmind`)

NOTE In the current release, you cannot ping the Distributed Database Service (`csdwpd`), Event Notification Service (`enpd`), or Notification Service (`csnotifyd`).

To run `cstool`, the Calendar Server must be running.

For example, to ping the machine with the host name `calserver` to see if the `cshttpd` service is listening on port 80:

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

By default, `cstool` waits 120 seconds for a response; however, you can change by value by using the `-t` *timeout* option.

Refreshing the Calendar Server Configuration

To force a Calendar Server service to refresh its configuration, use the cstool utility `refresh` command. If no service is specified, the command refreshes the configuration of all Calendar Server services.

To run `cstool`, the Calendar Server must be running.

For example, to force a local Calendar Server to refresh configurations for all services:

```
cstool refresh
```

Chapter 4

Managing Calendar Server Access Control

iPlanet Calendar Server uses Access Control Lists (ACLs) to determine the access control for calendars, calendar properties, and calendar components such as events and todos (tasks).

This chapter contains these sections:

- Access Control by Users
- Access Control Lists (ACLs)
- Public and Private Events and Tasks Filter
- Configuration Parameters for Access Control
- Command-Line Utilities for Access Control

Access Control by Users

The Calendar Server considers the following users when determining access to calendars, calendar properties, and calendar components:

- Primary calendar owners

  Primary calendar owners have full access to their own calendars. The Calendar Server does not perform any access control checks for primary owners accessing their own calendars.

- Administrators and superusers

  An administrator such as `icsuser` or `calmaster`, or a superuser such as `root`, is not subject to access control restrictions and can perform any operation on a calendar or calendar component. For more information, see "Calendar Server Administrators," on page 22.

- Other calendar owners

  Primary calendar owners can designate other owners for their calendars. The other owner can then act on behalf of the primary owner to schedule, delete, modify, accept, or decline events or todos (tasks) for a calendar.

- `anonymous` user

  The special calendar ID (`calid`) `anonymous` can access the Calendar Server using any password, if `service.http.allowanonymouslogin` in the `ics.conf` file is set to "yes" (which is the default). The `anonymous` user is not associated with any particular domain.You can change the `calid` for the `anonymous` user by editing the `calstore.anonymous.calid` parameter.

  You can also view a calendar anonymously if the calendar's permissions allow read access for everybody. For example, the following link allows users to anonymously view the calendar with the `calid` `tchang:meetings` (if the calendar's permissions allow read access for everybody):

  ```
  http://calendar.sesta.com:8080/?calid=tchang:meetings
  ```

  An `anonymous` user can view, print and search for public events and tasks on the calendar but cannot perform any other operations.

  For information about viewing a resource calendar anonymously, see "Linking to a Resource Calendar" on page 54.

Access Control Lists (ACLs)

The Calendar Server uses access control lists (ACLs) to determine access control for calendars, calendar properties, and calendar components such as events and todos (tasks). An ACL consists of one or more access control entries (ACEs), which are strings that collectively apply to the same calendar or component Each ACE in an ACL must be separated by a semicolon. For example:

- `jsmith^c^wd^g` consists of a single ACE.
- `@@o^a^r^g;@@o^c^wdeic^g;@^a^sf^g` consists of three ACEs.

An ACE consists of the following elements, with each element separated by a caret (`^`):

- Who - The individual, user, domain, or type of user who the ACE applies to.
- What - The target being accessed, such as a calendar or a calendar component such as an event, todo (task), or calendar property.
- How - The type of access control rights permitted, such as read, write, or delete.
- Grant - A specific access control right that is either granted or denied.

For example, in the ACE `jsmith^c^wd^g`:

- `jsmith` is the Who element, indicating who the ACE applies to.
- `c` is the What element, indicating what is being accessed (only the calendar components).
- `wd` is the How element, indicating which access rights are to be granted or denied (write and delete).
- `g` is the Grant element, indicating that the specified access rights, write and delete, for the calendar components are granted to `jsmith`.

Who

The Who element is the principal value for an ACE and indicates who the ACE applies to, such an individual user, domain, or specific type of user.

Who is also called the Universal Principal Name (UPN). The UPN for a user is the user's login name combined with the user's domain. For example, user `bill` in domain `sesta.com` has the UPN `bill@sesta.com`.

Table 4-1 shows the Who formats used in Calendar Server ACEs.

Table 4-1 Who Formats for Access Control Entry (ACE) Strings

Format	Description
user	Refers to a specific user. For example: `jsmith`.
user@domain	Refers to a specific user at a specific domain. For example: `jsmith@sesta.com`.
@domain	Refers to any user at the specified domain. For example: `@sesta.com` specifies `jsmith@sesta.com`, `sally@sesta.com`, and anyone else at `sesta.com`. Use this format to grant or deny access to an entire domain of users.
`@`	Refers to all users.
`@@{p|o|n}`	Refers to owners for the calendar: • `@@p` – primary owner only • `@@o` – all owners, including the primary owner • `@@n` – not an owner

What

The What element specifies the target being accessed, such as a calendar, calendar component (event or task), or calendar property.

Table 4-2 shows the What target values used in Calendar Server ACEs.

Table 4-2 What Values for Access Control Entry (ACE) Strings

Value	Description
c	Specifies calendar components such as events and tasks
p	Specifies calendar properties such as name, description, owners, and so forth
a	Specifies an entire calendar (all), including both components and properties

How

The How element specifies the type of access control rights permitted, such as read, write, or delete.

Table 4-3 shows the How types of access control rights used in Calendar Server ACEs.

Table 4-3 How Types for Access Control Entry (ACE) Strings

Type	Description
r	Read access.
w	Write access, including adding new items and modifying existing items.
d	Delete access.
s	Schedule (invite) access. Requests can be made, replies will be accepted, and other iTIP scheduling interactions will be honored.
f	Free/busy (availability) access only. Free/busy access means that a user can see scheduled time on a calendar, but is not allowed to see the event details. Instead, only the words “Not Available” appear by a scheduled time block. Blocks of time without any scheduled events are listed with the word “Available” next to them.
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.

Table 4-3 How Types for Access Control Entry (ACE) Strings *(Continued)*

Type	Description
`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.
`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.

Grant

The Grant element specifies whether to grant or deny access for a specified access type, such as `d` (delete) or `r` (read).

Table 4-4 shows the Grant attribute values used in Calendar Server ACEs.

Table 4-4 Grant Values for Access Control Entry (ACE) Strings

Value	Description
`g`	Grant the specific access control right.
`d`	Deny the specific access control right.

Examples of ACEs

The following examples show the use of ACEs:

- Grant the user ID `jsmith` read access to the entire calendar, including both components and properties:

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

- Grant `jsmith` write and delete access to components only:

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

- Grant all users in the sesta.com domain privileges to schedule, availability, and read access to components only:

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

- Grant other owners write and delete access to components only:

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

- Deny jsmith all access to calendar data:

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

- Grant all owners read, schedule, and availability access to the entire calendar, including both components and properties:

  ```
  @@o^a^rsf^g
  ```

- Grant read access to all users:

  ```
  @^a^r^g
  ```

Placing ACEs in an ACL

When the Calendar Server reads an ACL, it uses the first ACE it encounters that either grants or denies access to the target. Thus, the ordering of an ACL is significant, and ACE strings should be ordered such that the more specific ones appear before the more general ones.

For example, suppose the first ACE in an ACL for the calendar jsmith:sports grants read access to all owners and user bjones is one of the owners. Then, the Calendar Server encounters a second ACE that denies bjones read access to this calendar. In this case, the Calendar 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.

Configuration Parameters for Access Control

Table 4-5 describes the configuration parameters in the `ics.conf` file that the Calendar Server uses for access control. For more information see Chapter 8, “Calendar Server Configuration.”

Table 4-5 Access Control Configuration Parameters

Parameter	Description
`calstore.calendar.default.acl`	Specifies the default access control settings used when a user creates a calendar. The default is: `"@@o^a^r^g;@@o^c^wdeic^g;@^a^fs^g;@^c^^g;@^p^r^g"`
`calstore.calendar.owner.acl`	Specifies the default access control settings for owners of a calendar. The default is: `"@@o^a^rsf^g;@@o^c^wdeic^g"`.
`resource.default.acl`	Specifies the default access control settings used when a resource calendar is created. The default is: `"@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g"`

Public and Private Events and Tasks Filter

When creating a new event or task, a user can specify whether the event or task is Public, Private, or Time and Date Only (confidential):

- Public—Anyone with read permission to the user’s calendar can view the event or task.
- Private—Only owners of the calendar can view the event or task.
- Time and Date Only (confidential)—Owners of the calendar can view the event or task. Other users with read permission to the calendar can see only “Untitled Event” on the calendar, and the title is not an active link.

The `calstore.filterprivateevents` determines whether the Calendar Server filters (recognizes) Private and Time and Date Only (confidential) events and tasks. By default this parameter is set to “yes”. If you set `calstore.filterprivateevents` to “no”, the Calendar Server treats Private and Time and Date Only events and tasks as if they are Public.

Command-Line Utilities for Access Control

The Calendar Server provides the following command-line utilities to allow you to set or modify ACLs for access control:

Table 4-6 Command-Line Utilities for Access Control

Utility	Description
cscal	Use the `create` and `modify` commands with the `-a` option to set ACLs for specific user's calendars.
csresource	Use the `create` command with the `-a` option to set ACLs for resource calendars for resources such as conference rooms or equipment.
csuser	Use the `csuser` utility with the `-a` option to manage calendar user information stored in an LDAP directory server and the calendar database.

Chapter 5

Managing Calendar Server Databases

This chapter describes how to manage iPlanet Calendar Server databases and calendar data using the csdb utility.

This chapter contains these sections:

- Specifying a Target Database
- Viewing Calendar Database Status
- Recovering a Damaged Database
- Deleting a Calendar Database
- Importing and Exporting Calendar Data
- Checking and Rebuilding a Calendar Database

To run the csdb utility, you must log in as a user who has administrative rights to the system where the Calendar Server is running. For more information, see Chapter 7, "Calendar Server Command-Line Utilities."

For information about backing up and restoring a calendar database, see Chapter 6, "Backing Up and Restoring Calendar Server Data."

Specifying a Target Database

The `csdb` utility `-t` option allows you to specify a target database:

- `-t caldb` — calendar database
- `-t sessdb` — session database
- `-t statdb` — statistics database

If you do not include the `-t` option, `csdb` operates on all three databases, except for the `check` and `rebuild` commands, which operate only on the calendar database.

Table 5-1describes the calendar database (`caldb`) files:

Table 5-1 Calendar Database Files

File	Description
`ics50calprops.db`	Calendar properties for all calendars. Includes the calendar ID (`calid`), calendar name, Access Control List (ACL), owner, and more.
.`ics50events.db`	Events for all calendars.
`ics50todos.db`	Todos (tasks) for all calendars.
`ics50alarms.db`	Alarms for all events and todos (tasks).
`ics50gse.db`	Queue of scheduling requests for the group scheduling engine (GSE).
`ics50journals.db`	Journals for all calendars. Journals are not implemented in the current release of Calendar Server.
`ics50caldb.conf`	Database version identifier.

Viewing Calendar Database Status

To view the status of a calendar database, use the `csdb` utility `list` command. The Calendar Server can be either running or stopped.

Use the `-t` option to specify the target database (`caldb`, `sessdb`, or `statdb`); otherwise, `csdb` operates on all three databases.

For example, to view database status and statistics for all databases:

```
csdb list
```

To view information about the calendar database in the current directory in verbose mode:

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

Deleting a Calendar Database

To delete a calendar database, use the csdb utility delete command. The Calendar Server must be stopped.

Use the -t option to specify the target database (caldb, sessdb, or statdb); otherwise, csdb deletes all three databases.

For example, to delete the calendar database:

```
csdb -t caldb delete
```

The csdb utility issues a warning before deleting the database.

Importing and Exporting Calendar Data

To export and import calendar data to and from a file, use the csexport and csimport utilities. The calendar data can be in either iCalendar (.ics) or XML (.xml) format.

You must run csexport and csimport locally on the machine where your Calendar Server is installed. The Calendar Server can be either running or stopped.

Exporting Calendar Data

To export calendar data to a file, use csexport. 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 calendar ID (calid) JSmithcal in iCalendar (text/calendar MIME) format to a file named jsmith.ics:

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

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

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

Importing Calendar Data

To import calendar data from a file previously saved using the csexport utility, use csimport. 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 calendar ID (calid) JSmithcal from the file jsmith.ics that was saved in iCalendar (text/calendar MIME) format:

```
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:

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

If the specified calendar ID (calid) already exists, its data is cleared before the new data is imported.

Recovering a Damaged Database

To recover a damaged session or statistics database, use the csdb utility recover command. Use the -t option to specify the target database: sessdb (session) or statdb (statistics).

Before you run the recover command, stop the Calendar Server using the stop-cal command.

For example, to recover a damaged session database in the current directory:

```
csdb -t sessdb recover
```

NOTE To recover a damaged calendar database (caldb), use the csdb utility check and rebuild commands instead of recover.

For more information, see “Checking and Rebuilding a Calendar Database,” on page 91.

Checking and Rebuilding a Calendar Database

The `csdb` utility commands includes the following commands to check and, if necessary, to rebuild a calendar database (`caldb`):

- The `check` command scans a calendar database to determine if any corruption has occurred and reports the results in its output. (The `check` command does not check for corruption in the alarm or group scheduling engine (GSE) database.)
- The `rebuild` command also scans a calendar database to determine if any corruption has occurred and if necessary, generates a rebuilt calendar database (`.db` files).

The `csdb` utility also incudes the `recover` command to recover a damaged session or statistics database. If your calendar database is corrupted, use `check` and `rebuild` instead of `recover`.

CAUTION Before you run these commands, backup your calendar database using the `csbackup` utility (or another backup utility).

Checking a Calendar Database for Corruption

The `check` command scans a calendar database and checks calendar properties (calprops) events and todos (tasks) for corruption. If the `check` command finds an inconsistency that cannot be resolved, it reports the situation in its output.

You should run the `check` command regularly to check your calendar database for inconsistencies. For example, consider running `check` after each backup of your database. However, if you already know that your calendar database is corrupted, you don't need to run the `check` command. Go ahead and rebuild your corrupted database.

To check a calendar database for corruption:

1. Log in as a user who has administration rights to the system where the Calendar Server is installed.
2. The Calendar Server can be either running or stopped; however, if possible, stop the Calendar Server.
3. Make a copy of your calendar database, if you haven't already done so. Copy only the six database (`.db`) files. You don't need to copy any share (`*.share`) or log (`log.*`) files.

4. Change to the *server-root*/`cal/bin/` directory. For example, on Solaris, enter:

   ```
   cd /opt/SUNWics5/cal/bin
   ```

5. Run the `check` command on the copy of your calendar database:

   ```
   ./csdb check dbdir > /tmp/check.out 2>&1
   ```

   If you don't specify *dbdir*, `check` uses the database in the current directory.

   The `check` command can generate a lot of information, so consider redirecting all output, including `stdout` and `stderr`, to a file (as shown in the example).

6. When `check` has finished, review the output file. If your database is corrupted, run the `rebuild` command.

Rebuilding a Corrupted Calendar Database

The `rebuild` command scans a calendar database and checks the calendar properties (calprops) events and todos (tasks) for corruption. If the `rebuild` command finds an inconsistency, it generates a rebuilt calendar database (`.db` files) in the *server-root*/`cal/bin/rebuild_db/` directory.

NOTE The `rebuild` command rebuilds all databases except the group scheduling engine (GSE) database. To determine if the GSE database has any entries, run the `csschedule -v list` command and then let the GSE finish processing the entries before you run the `rebuild` command.

To rebuild a corrupted calendar database:

1. Log in as a user who has administration rights to the system where the Calendar Server is installed.

2. Stop the Calendar Server.

3. Make a copy of your calendar database, if you haven't already done so. Copy only the six database (`.db`) files. You don't need to copy any share (`*.share`) or log (`log.*`) files.

4. Change to the *server-root*/`cal/bin/` directory. For example, on Solaris, enter:

   ```
   cd /opt/SUNWics5/cal/bin
   ```

5. Run the `rebuild` command on the copy of your calendar database:

   ```
   ./csdb rebuild dbdir > /tmp/rebuild.out 2>&1
   ```

   If you don't specify a directory, `rebuild` uses the database in the current directory.

   The `rebuild` command can generate a lot of information, so consider redirecting all output, including `stdout` and `stderr`, to a file (as shown in the example).

NOTE Always rebuild your calendar database using the latest backup copy.

However, if you have experienced a significant loss of data and you have periodically backed up your database and have more than one copy available, rebuild from the latest copy to the oldest one. (The only drawback is that calendar components that were deleted will reappear in the rebuilt database.)

For example, if you have three sets of backup calendar database files in directories `db_0601`, `db_0615`, and `db_0629`, run the `rebuild` command in the following sequence:

```
./csdb rebuild db_0629
./csdb rebuild db_0615
./csdb rebuild db_0601
```

The `rebuild` command then writes the rebuilt database to the *server-root*`/cal/bin/rebuild_db/` directory.

---

6. When `rebuild` has finished, review the output in the `rebuild.out` file. If the rebuild was successful, the last line in the `rebuild.out` file should be:

   ```
   Calendar database has been rebuilt
   ```

7. After you have verified that rebuild was successful in the previous step, copy the rebuilt database (`.db`) files from the *server-root*`/cal/bin/rebuild_db/` directory to your production database.

8. If any share (`*.share`) or log (`log.*`) files are left from the corrupted database, delete them or move them to another directory.

9. Restart the Calendar Server.

Chapter 6

Backing Up and Restoring Calendar Server Data

To back up and restore Calendar Server data, use these command-line utilities:

- `csbackup` backs up the calendar database, a specified calendar, or a user's default calendar.
- `csrestore` restores the calendar database, individual calendars, or a user's default calendar that was saved using `csbackup`.

This chapter contains these sections:

- Backing Up Calendar Server Data
- Restoring Calendar Server Data
- Using Solstice Backup™ or Legato Networker®

CAUTION The Calendar Server 2.x and 5.x `backup` and `restore` utilities are not compatible. Do not try to restore calendar data backed up by the Calendar Server 2.x `backup` utility, because data loss can occur.

If you need to migrate 2.x calendar data to 5.x, use the `ics2migrate` migration utility, which is described in the *iPlanet Calendar Server Installation Guide*.

Backing Up Calendar Server Data

The csbackup utility can back up the calendar database, a specified calendar, or a user’s default calendar. This section describes:

- Backing Up the Calendar Database to a Directory
- Backing Up a Specific Calendar to a File
- Backing Up a User’s Default Calendar to a File

Backing Up the Calendar Database to a Directory

To backup the calendar database to a target backup directory, use the csbackup utility database command. For example, to back up the calendar database to a directory named backupdir:

```
csbackup -f database backupdir
```

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

NOTE Currently, csbackup utility fails if the target backup directory already exists and you do not specify the -f option. For example, the following command fails if backupdir exists, even if the directory is empty:

```
csbackup database backupdir
```

Therefore, if you specify a target backup directory that already exists, include the -f option when you run the csbackup utility.

Backing Up a Specific Calendar to a File

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

To backup a calendar to a backup file in iCalendar or XML format, use the csbackup utility calendar command. The file-name extension (.ics or .xml) of the backup file indicates the format.

For example, to backup the calendar JSmithcal in iCalendar format (text/calendar MIME) to the file jsmith.ics in the backupdir directory:

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

Or, to backup the calendar JSmithcal in XML format (text/XML) to the file jsmith.xml in the bcakupdir directory:

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

Backing Up a User’s Default Calendar to a File

To back up a user’s default calendar to a text file in iCalendar or XML format, use the csbackup utility defcal command. The file-name extension (.ics or .xml) that you specify for the output file determines which format is used.

For example, to back up calendar user JSmith’s default calendar in iCalendar (text/calendar MIME) format to a file named jsmith.ics:

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

Or, to back up calendar user JSmith’s default calendar in XML (text/xml MIME) format to a file named jsmith.xml:

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

Restoring Calendar Server Data

The csrestore utility restores the calendar database, individual calendars, or a user's default calendar that was saved using csbackup. You must run the csrestore utility on the local machine where the Calendar Server is installed, and you must first stop the Calendar Server. (The Calendar Server can be running, however, when you backup the database.)

This section describes:

- Restoring the Calendar Database
- Restoring a Calendar From a Backup Directory
- Restoring a Calendar From a File
- Restoring a User's Default Calendar

Restoring the Calendar Database

To restore a calendar database that was saved to a backup directory using the csbackup utility, use the csrestore utility database command.

For example, to restore the calendar database that was saved to a backup directory named backupdir:

```
csrestore database backupdir
```

Restoring a Calendar From a Backup Directory

To restore a specific calendar from a database that was saved to a backup directory using the csbackup utility, use the csrestore utility database command with the -c option.

For example, to restore the calendar JSmithcal from the backup database directory backupdir:

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

Restoring a Calendar From a File

To restore a specific calendar that was saved to a backup file using the csbackup utility, use the csrestore utility calendar command with the -c option. The file-name extension (.ics or .xml) of the backup file indicates the format in which the calendar 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:

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

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

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

Restoring a User's Default Calendar

To restore a a user's default calendar that was saved to a backup file using the csbackup utility, use the csrestore utility defcal command. The file-name extension (.ics or .xml) of the backup file indicates the format in which the calendar 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:

```
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:

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

Using Solstice Backup™ or Legato Networker®

You can also use either Solstice Backup or the Legato Networker to back up and restore Calendar Server data. Solstice Backup and the Legato Networker are identical. The instructions in this section apply to both products. Before attempting to backup the Calendar Server, however, see the Solstice Backup or Legato Networker documentation.

For Solstice Backup documentation, see `http://docs.sun.com`.

This section describes:

- Backing Up Calendar Server Data Using Solstice Backup or Legato Networker
- Restoring Calendar Server Data Using Solstice or Legato Software

If you do not have Solstice Backup or Legato Networker, use the Calendar Server `csbackup` and `csrestore` utilities.

Backing Up Calendar Server Data Using Solstice Backup or Legato Networker

To backup Calendar Server data:

1. Run the `csbackup` utility to back up the calendar database or specified calendars using the `-l` option.

   For more information, see "csbackup" on page 109.

   The backup procedure creates a backup directory under the current directory. This directory is not the actual directory where 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 Backup or Legato Networker.

   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 by the 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. iPlanet recommends that you set up the backup program's `backup` command to run the Calendar Server `csbackup` command-line utility before the running the backup program's `save` command to achieve an automated backup process.

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

Restoring Calendar Server Data Using Solstice or Legato Software

To restore Calendar Server data:

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

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

2. 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 7

Calendar Server Command-Line Utilities

iPlanet Calendar Server provides a set of command-line administration utilities that can be invoked from batch, shell, and scripting programs such as Perl. If needed, these utilities use default values from the `ics.conf` configuration file.

The command-line utilities are located in the *server-root*`/cal/bin` directory.

This chapter provides the following information:

- Running the Command-Line Utilities
    - Syntax for Command-Line Utilities
    - Usage Rules for Command-Line Utilities
- Descriptions of Command-Line Utilities (table)

    csattribute
    csbackup
    cscal
    cscomponents
    csdb
    csexport
    csplugin
    csresource
    csrestore
    csschedule
    csstart
    csstats
    csstop
    cstool
    csuser

Running the Command-Line Utilities

On UNIX systems, run the command-line utilities while logged in as the user and group under which the iPlanet Calendar Server is running (defaults are `icsuser` and `icsgroup`) that was specified during installation, or as `root`. To run the command-line utilities on Windows NT systems, the user must have full administrator privileges.

In most cases, you must change to the directory where the utilities 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 systems

  `c:\ProgramFiles\iPlanet\CalendarServer5\cal\bin`

Syntax for Command-Line Utilities

The 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` or `csuser`.

`option` determines which action the command performs. Options are in lowercase and preceded by a hyphen (-), such as `-d`. An option enclosed in brackets (`[]`) is optional. If indicated, of two or more options can be used at the same time.

`value` further qualifies the action specified by `option`, such as a description used with the `-d` option. A value enclosed in brackets (`[]`) indicates it is optional. Values that include spaces must be enclosed in quotation marks (" "). Multiple values must be enclosed in quotation marks (`""`), and each value must be 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 performs such as `list` or `create`. 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 a calendar ID or user ID.

Usage Rules for Command-Line Utilities

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

- If you do not specify a command, the utility lists all options and commands along with examples.
- If you do not specify a required password, the utility prompts for it.
- The `-v` (verbose) and `-q` (quiet) options are available for each utility.
- If a command is dangerous (that is, one that can result in data loss), the utility prompts for confirmation before executing the command. Examples of dangerous commands are `cscal`, which can delete a calendar, and `csuser`, which can delete a user. The `-q` (quiet) option, however, disables confirmation prompting.
- The `version` command is available for each utility.

Descriptions of Command-Line Utilities

Table 7-1 iPlanet Calendar Server Command-Line Utilities Summary

Utility	Description
`csattribute`	Manages the LDAP attributes of a calendar user or resource.
`csbackup`	Backs up individual calendars, users, and the calendar database.
`cscal`	Manages calendars and their properties.
`cscomponents`	Manages calendar components: events and tasks (todos).
`csdb`	Manages the calendar database.
`csexport`	Exports a calendar in iCalendar (`.ics`) or XML (`.xml`) format.
`csimport`	Imports a calendar in iCalendar (`.ics`) or XML (`.xml`) format.
`csplugin`	Views, enables, or disables configured Calendar Server API (CSAPI) plug-ins.
`csresource`	Manages calendar resources such as conference rooms and equipment.
`csrestore`	Restores individual calendars, users, and the calendar database.
`csschedule`	Manages scheduling entries in the Group Scheduling Engine (GSE) queue.
`csstart`	Starts the Calendar Server.
`csstats`	Displays counters in a Calendar Server.
`csstop`	Stops the Calendar Server.
`cstool`	Pings a running Calendar Server instance or causes the Calendar Server to refresh its configuration.
`csuser`	Manages calendar users.

csattribute

The `csattribute` utility manages the Calendar Server attributes in the LDAP server. Commands are:

- `add` an LDAP attribute and value to a specified target (user or resource object).
- `list` the attributes of a target object.
- `delete` an attribute from a target.

Requirements

- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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

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

csattribute [-q | -v] [-t resource | user] list target
```

Table 7-2 describes the commands available for `csattribute`.

Table 7-2 `csattribute` Utility Commands

Command	Description
`add` *target*	Adds an LDAP attribute and value to a specified target (user or resource object).
`list` *target*	Lists the attributes of a target object.
`delete` *target*	Deletes an attribute from a target.
`version`	Displays the version of the utility.

Table 7-2 describes the `csattribute` utility command options.

Table 7-3 `csattribute` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-a` *attribute* =*value* or `-a` *attribute* [=*value*]	An LDAP attribute and value: • *attribute* is required when using the `-a` option. • *value* is required when the `-a` option is used with the `add` command, but it is optional when the `-a` option is used with the `delete` and `list` commands.
`-t user | resource`	The type of target (`user` or `resource` object). Default is `user`.

Examples

- Add the LDAP attribute `icsCalendar` with the value `tchang` to the user ID `TChang`:

  ```
  csattribute -a icsCalendar=tchang add TChang
  ```

- Delete the LDAP attribute `icsCalendar` from `TChang`:

  ```
  csattribute -a icsCalendar delete TChang
  ```

- Display the attributes of `TChang`:

  ```
  csattribute list TChang
  ```

csbackup

The `csbackup` utility backs up the calendar database, a specified calendar, or a user's default calendar. Commands are:

- `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.

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

For information about `csrestore`, see "csrestore," on page 132.

Requirements

- The Calendar Server can be running or stopped.
- You must run the utility locally on the machine where the Calendar Server is installed.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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


csbackup [-q|-v] -c calid calendar target


csbackup [-q|-v] -a userid [-b basedn] defcal target
```

Table 7-4 describes the commands available for `csbackup`.

Table 7-4 csbackup Utility Commands

Command	Description
`database` *target*	Backs up the calendar database to the specified target database directory. By default, the target database directory is: *server-root*`/cal/bin/`*target-directory* If you specify only the target database directory, do not include the slash (/) before the directory name. For example: `csbackup database backupdir`
`calendar` *calid target*	Backs up the specified calendar ID to the specified target output file. The data format of the file is assumed by the file extension, `.ics` for text/calendar or `.xml` for text/xml.
`defcal` *userid target*	Backs up the default calendar of the specified user ID 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`	Displays the version of the utility.

Table 7-5 describes the `csbackup` utility command options.

Table 7-5 `csbackup` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-a` *userid*	The user ID of the calendar user to backup. This option is required for the `defcal` command. There is no default.

Table 7-5 `csbackup` Utility Command Options *(Continued)*

Option	Description
`-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 occur. For example, if you specify a base DN of `ou=people, o=sesta.com`, all LDAP search operations executed by the Calendar Server examine only the `ou=people` subtree in the `o=sesta.com` directory tree.
`-c` *calid*	The calendar ID to backup. This option is required with the `calendar` command. There is no default. For more information, see "Calendar Identifiers (calids)," on page 42.
`-f`	To force any existing backup files to be deleted. In the current release, you must include the `-f` option if the backup target directory already exists, even if the directory is empty.
`-l`	To prepare the backup file for use with the Solstice™ Backup™ or the Legato Networker™ backup programs. For more information, see Chapter 6, "Backing Up and Restoring Calendar Server Data."

Examples

- Backup the calendar database to a directory named `backupdir`:

```
csbackup database backupdir
```

- Backup the calendar with the calendar ID `tchang` to the file `tchang.ics as text/calendar`:

```
csbackup -c tchang calendar tchang.ics
```

- Backup the default calendar for `tchang` to the file `tchang.xml as text/xml`:

```
csbackup -a tchang defcal tchang.xml
```

cscal

The `cscal` utility manages calendars and their properties. Commands are:

- `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

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

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


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


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

Table 7-6 describes the commands available for the `cscal` utility.

Table 7-6 `cscal` Utility Commands

Command	Description
`create` *calid*	Creates the calendar specified by *calid*.
`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*`]`	Enables the calendar specified as *calid*.
	If the `-o owner` option is specified, enables all calendars whose primary owner is the specified owner.
`disable [`*calid*`]`	Disables the calendar specified as *calid*.
	If the `-o owner` option is specified, disables all calendars whose primary owner is the specified owner.
`list [`*calid*`]`	Lists 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*	Modifies the properties of the calendar specified by *calid.*
`reset [`*calid*`]`	Resets the properties of the calendar specified by *calid* to the default configuration settings.
`version`	Displays the version of the utility.

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

Table 7-7 cscal Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.

Table 7-7 cscal Utility Command Options *(Continued)*

Option	Description
`-a [`*aces*`]`	Access Control Entries (ACEs) for a specified calendar. ACEs determine who can access a calendar for group scheduling and the types of permissions they have, such as create, delete, read, and write privileges. An ACE string or Access Control List (ACL), must be enclosed in quotation marks (`""`). The default is the `calstore.calendar.default.acl` parameter in the `ics.conf` file. For details about the ACE format, see Chapter 4, "Managing Calendar Server Access Control."
`-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 calendar ID (`calid`), which is usually the same as the user ID.
`-y` *otherowners*	Other calendar owners. Multiple owners must be enclosed in quotation marks(`""`) and separated by spaces. The default is no other owners.

Examples

- Create the calendar with the calendar ID `tchang` with `TChang` as the primary owner with the visible name `Public_Calendar` using the default access control settings (as defined by `calstore.calendar.default.acl` in the `ics.conf` file):

```
cscal -o TChang -n Public_Calendar create tchang
```

- Modify calendar chang so that anyone has read and write access, it is associated with the category *sports*, and it is co-owned by JSmith:

  ```
  cscal -a "@^a^rw^g" -g sports -y JSmith modify tchang
  ```

- Disable the calendar with the calendar ID tchang (users will not be allowed to read, write to, or locate it using the user interface):

  ```
  cscal disable tchang
  ```

- Enable the calendar with the calendar ID tchang (users are allowed to read or write to it using the user interface), but it does not allow double-booking:

  ```
  cscal -k no enable tchang
  ```

- List the properties of tchang:

  ```
  cscal list tchang
  ```

- List all the properties of tchang:

  ```
  cscal -v list tchang
  ```

- List all the calendars in the database:

  ```
  cscal list
  ```

- Reset the calendar with the calendar ID tchang to the default configuration settings:

  ```
  cscal reset tchang
  ```

- Remove a description from the calendar with the calendar ID tchang:

  ```
  cscal -d "" modify tchang
  ```

- Remove all categories from the calendar with the calendar ID tchang:

  ```
  cscal -g "" modify tchang
  ```

- Remove other owners from the calendar with the calendar ID tchang:

  ```
  cscal -y "" modify tchang
  ```

- Delete tchang from the calendar database:

  ```
  cscal delete tchang
  ```

- Delete all calendars from the calendar database whose primary owner is TChang:

  ```
  cscal -o TChang delete
  ```

cscomponents

The cscomponents utility manages calendar components: events and tasks (todos). Commands are:

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

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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

Table 7-8 describes the commands available for the cscomponent utility.

Table 7-8 cscomponent Utility Commands

Command	Description
delete *calid*	Deletes events and tasks in the calendar with the specified calendar ID.
list *calid*	Lists events and tasks in the calendar with the specified calendar ID.
version	Prints the version of the utility to the screen.

Table 7-9 describes the `cscomponent` utility command options.

Table 7-9 `cscomponent` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress 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.

Examples

- Delete all 2000 events in the calendar with the calendar ID `tchang`:

  ```
  cscomponents -s 20000101T000000Z -e 20001231T000000Z delete
  tchang
  ```

- List all events and tasks with details in the calendar with the calendar ID `tchang`:

  ```
  cscomponents -v list tchang
  ```

csdb

The csdb utility manages the calendar databases (calendar, session, and statistics). Commands are:

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

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server must be stopped for the create, delete, or rebuild commands.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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

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

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

csdb check|rebuild [dbdir]
```

Table 7-10 describes the commands available for the `csdb` utility.

Table 7-10 `csdb` Utility Commands

Command	Description
`create [`*dbdir*`]`	Creates the databases in the specified database directory. If a database directory is not specified, the current directory is used. If a database does not exist when the server is started, the Calendar Server creates one automatically.
`delete [`*dbdir*`]`	Deletes the databases in the specified database directory. If a database directory is not specified, the current directory is used. A database cannot be deleted while it is open (when the Calendar Server is running).
`list [`*dbdir*`]`	Lists information about the databases in the specified database directory. If a database directory is not specified, the current directory is used.
`recover [`*dbdir*`]`	Attempts to recover damaged databases in the specified database directory. If a database directory is not specified, the current directory is used.
`check [`*dbdir*`]`	Scans a calendar database in the specified database directory to determine if any corruption has occurred and reports the results in its output. If a database directory is not specified, the current directory is used.
`rebuild [`*dbdir*`]`	Scans a calendar database in the specified database directory to determine if any corruption has occurred and if so, generates a rebuilt calendar database (`.db` files). If a database directory is not specified, the current directory is used.
`version`	Displays the version of the utility.

Table 7-11 describes the `csdb` utility command options.

Table 7-11 `csdb` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.

Table 7-11 `csdb` Utility Command Options *(Continued)*

Option	Description
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-f`	Force the recovery of the calendar database.
`-t caldb|sessdb|statdb`	Specifies the target database: • `caldb` (calendar) • `sessdb` (session) • `statdb` (statistics) **Note:** If `-t` is not specified, `csdb` operates on all databases, except for the `check` and `rebuilt` commands, which operate only on `caldb` (calendar).

Examples

- Create new, unpopulated databases in the current directory:

  ```
  csdb -t caldb create
  ```

- Delete the databases in the current directory:

  ```
  csdb -t caldb delete
  ```

- List information about the calendar database in the current directory:

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

- Attempt to recover all damaged databases in the current directory:

  ```
  csdb recover
  ```

- List information about the sessions database in the current directory:

  ```
  csdb -t sessdb list
  ```

- Attempt to recover a damaged statistics database in the current directory:

  ```
  csdb -t statdb recover
  ```

csexport

The `csexport` utility exports a calendar to a file in iCalendar (`.ics`) or XML (`.xml`) format. Commands are:

- `calendar` exports a specified calendar.
- `version` displays the version number of the utility currently installed.

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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

Table 7-12 describes the commands available for the `csexport` utility.

Table 7-12 `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 7-13 describes the `csexport` utility command options.

Table 7-13 csexport Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-c` *calid*	The calendar ID of the calendar to export. This option is required with the `calendar` command. There is no default.

Examples

- Export the calendar with the calendar ID `tchang` in iCalendar (text/calendar) format to a file named `tchang.ics`:

  ```
  csexport -c tchang calendar tchang.ics
  ```

- Exports the calendar with the calendar ID `tchang` in XML (text/xml) format to a file named `tchang.xml`:

  ```
  csexport -c tchang calendar tchang.xml
  ```

csimport

The `csimport` utility imports a calendar from a file in iCalendar (`ics`) or XML format that was saved with the `csexport` utility. Commands are:

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

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

```
csimport [-v|-q] -c calid calendar inputfile
```

Table 7-14 describes the commands available for the `csimport` utility.

Table 7-14 `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 7-15 describes the `csimport` utility command options.

Table 7-15 `csimport` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-c` *calid*	The calendar ID of the calendar to import. This option is required with the `calendar` command. If the specified calendar ID already exits, the imported data is merged with the current calendar. There is no default. For more information, see "Calendar Identifiers (calids)," on page 42.

Examples

- Import the calendar with the calendar ID `tchang` from the file `tchang.ics` and expect iCalendar (text/calendar file) format:

```
csimport -c tchang calendar tchang.ics
```

- Import the calendar with the calendar ID `tchang` from the file `tchang.xml` and expect XML (text/xml file) format:

```
csimport -c tchang calendar tchang.xml
```

csplugin

The `csplugin` manages CSAPI plug-ins configured for your Calendar Server installation. Commands are:

- `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.

Requirements

- Must be run on the local machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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


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

Table 7-16 describes the commands available for the `csplugin` utility.

Table 7-16 `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 7-17.)
`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 7-17.)

Table 7-16 `csplugin` Utility Commands *(Continued)*

Command	Description
`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 7-17.)
`version`	Display the version of the utility.

Table 7-17 describes the `csplugin` utility command options.

Table 7-17 `csplugin` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-r`	When used with the `activate` command, physically copies the plug-in into the Calendar Server `plugin` directory. When used with the `deactivate` command, deletes the plug-in from the `plugin` directory.
`-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..

Examples

- 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 -v list
  ```

- Load and enable the `lookup` type plug-in with the file named `mylookup`:

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

- Disable the `lookup` type plug-in with the file named `mylookup` and then delete it from the `plugin` directory:

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

csresource

The csresource utility manages calendars for resources such as conference rooms or equipment stored in the LDAP server and the Calendar Server database. (The csresource utility is available only for calendars associated with a resource and returns an error if issued against a user's calendar.) Commands are:

- create adds a new resource for a specified calendar ID (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

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

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]
```

Table 7-18 describes the commands available for the csresource utility.

Table 7-18 csresource Utility Commands

Command	Description
create *name*	Create a new resource for a specified calendar ID.
delete [*name*]	Delete a resource or, if no resource *name* is specified, delete all resources.

Table 7-18 `csresource` Utility Commands *(Continued)*

Command	Description
`enable [`*name*`]`	Enable a resource or, if no resource *name* is specified, enable all resources.
`disable [`*name*`]`	Disable a resource or, if no resource *name* is specified, disable all resources.
`list [`*name*`]`	Display a single resource calendar or, if no resource *name* is specified, list of all resource calendars.

NOTE If the *name* contains a space in any of the above commands, it must be enclosed in quotation marks (`"  "`).

Table 7-19 describes the `csresource` utility command options.

Table 7-19 `csresource` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-a [`*aces]*	Access Control Entries (ACEs) for the specified calendar. ACEs determine who can access a calendar for group scheduling and the types of permissions they have, such as create, delete, read, and write privileges. An ACE string or Access Control List (ACL), must be enclosed in quotation marks (`""`). The default is the `resource.default.acl` parameter in the `ics.conf` file. For information about the ACE format, see Chapter 4, “Managing Calendar Server Access Control.

Table 7-19 `csresource` Utility Command Options *(Continued)*

Option	Description
`-b` [*basedn]*	LDAP base DN (distinguished name) to be used for the specified resource. Default is taken from the `local.ugldapbasedn` setting in the `ics.conf` file.
`-c` *calid*	The `icsCalendar` attribute. This option is required with the `create` command. For more information, see "Calendar Identifiers (calids)," on page 42.
`-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 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 `resource.allow.doublebook` setting in the `ics.conf` file.
`-o` *owner*	Primary owner. Default is taken from the setting `service.admin.calmaster.userid` 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.

Examples

- Display a list of all resource calendars and their LDAP attributes.

  ```
  csresource -v list
  ```

- Create a resource calendar with the calendar ID (`calid`) `room100` and the viewable name (LDAP `cn` attribute) `MeetingRoom100`:

  ```
  csresource -c room100 create MeetingRoom100
  ```

- Display the LDAP attributes of the resource calendar with the viewable name `MeetingRoom100`:

  ```
  csresource -v list MeetingRoom100
  ```

- Disable the resource calendar with the viewable name `MeetingRoom100`:

  ```
  csresource disable MeetingRoom100
  ```

- Enable the calendar with the resource calendar with the viewable name MeetingRoom100 and allow double-booking:

  ```
  csresource -k yes enable MeetingRoom100
  ```

- Delete the resource calendar with the viewable name MeetingRoom100:

  ```
  csresource delete MeetingRoom100
  ```

csrestore

The csrestore utility restores the calendar database, a specified calendar, or a user's default calendar that was saved using csbackup or csexport. Commands are:

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

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

CAUTION Calendar Server 5.x csrestore is not compatible with the 2.x version of csrestore. Do not try to restore data that was backed up using Calendar Server 2.x csrestore because data loss can occur.

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- If you are restoring the calendar database, the Calendar Server must be stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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


csrestore [-v|-q] -c calid calendar inputfile


csrestore [-v|-q] -a userid [-b basedn] defcal inputfile
```

Table 7-20 describes the commands available for the `csrestore` utility.

Table 7-20 `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 operation overwrites all previous contents of the current calendar database.
`calendar` *inputfile*	Restore the specified calendar ID 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 calendar ID already exists, the calendar's data is cleared before it is restored.
`defcal` *inputfile*	Restore the default calendar of the specified user ID 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 7-21 describes the `csrestore` utility command options.

Table 7-21 `csrestore` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-a` *userid*	The user ID to restore. This option is required with the `defcal` command. There is no default.

Table 7-21 csrestore Utility Command Options *(Continued)*

Option	Description
-b *basedn*	The LDAP base DN (distinguished name) to be used for the specified user ID. 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 calendar ID to restore. This option is required with the calendar command. There is no default. For more information, see "Calendar Identifiers (calids)," on page 42.

Examples

- Restore the calendar database stored in the directory backupdir that was previously saved using csbackup:

  ```
  csrestore database backupdir
  ```

- Restore the calendar with the calendar ID tchang from the file tchang.ics located in the directory backupdir that was previously saved in iCalendar (text/calendar file) format using csbackup or csexport:

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

- Restore tchang from the calendar database in backupdir that was previously saved using csbackup:

  ```
  csrestore -c tchang calendar backupdir
  ```

- Restore the default calendar owned by TChang from the file TChang.ics located in the directory backupdir that was previously saved in iCalendar (text/calendar file) format using csbackup or csexport:

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

csschedule

The `csschedule` utility manages schedule entries stored in the Group Scheduling Engine (GSE) queue. Commands are:

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

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server must be stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

```
csschedule [-q|-v] [-c count] [-e endtime] [-s starttime]


  [-t scheduletime -o offset] [-u uid] list [calid]


csschedule [-q|-v] [-t scheduletime -o offset -u uid -n sequencenumber
  -r rid] list [calid]


csschedule [-q|-v] [-t scheduletime -o offset -u uid -n sequencenumber
  -r rid] delete [calid]


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

Table 7-22 describes the commands available for the `csschedule` utility.

Table 7-22 `csschedule` Utility Commands

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

Table 7-23 describes the `csschedule` utility command options.

Table 7-23 `csschedule` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-c` *count*	The number of GSE queue entries to list. For example, specify 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 todo. An RID is a semicolon delimited list of strings that identify each occurrence of a recurring event or todo.
`-s` *starttime*	The starting time of the entry in the GSE queue where `0` means from the beginning of time. The default is `0`.

Table 7-23 `csschedule` Utility Command Options *(Continued)*

Option	Description
`-t` *scheduletime*	A schedule time, for example: `20001231T103045Z`
`-u` *uid*	The unique identifier (UID) of an entry in the GSE queue.

Examples

- List in detail all entries stored in the GSE queue:

  ```
  csschedule -v list
  ```

- List the first ten entries stored in the GSE queue:

  ```
  csschedule -c 10 list
  ```

- List the entries in the GSE queue scheduled between 10:30:45 to 11:30:45 on 12/31/2000:

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

- List the entry in the GSE queue for calendar `tchang` 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 2 -u 1111 -r 0 -n 0 list
  tchang
  ```

- Delete the entry in the GSE queue for calendar `tchang` 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 -t 20001231T103045Z -o 1 -u 1111 -r 0 -n 0 delete
  tchang
  ```

- Delete entries in the GSE that are scheduled between 10:30:45 and 16:30:45 on 12/31/2000:

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

- Delete all entries in the GSE queue:

  ```
  csschedule -v delete
  ```

csstart

NOTE The Calendar Server provides the `csstart` and `csstop` utilities only to provide compatibility with earlier releases.

iPlanet recommends that you use the `start-cal` and `stop-cal` commands to start and stop the Calendar Server. For more information, see "Starting and Stopping the Calendar Server," on page 56.

The `csstart` utility starts the Calendar Server. Commands are:

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

On Windows NT systems, you can also start the Calendar Server using the Services dialog box accessed from the Control Panel.

Requirements

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

Syntax

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


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

Table 7-24 describes the commands available for the `csstart` utility.

Table 7-24 `cscstart` Utility Commands

Command	Description
`check	list` [*servicename*]	Check if a specified Calendar Server service is running, or check if all Calendar Server services are running if a service name is not specified Or list all Calendar Server services (or a specified service) and their session IDs. *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 service • `admin` — Calendar Server administration service (required on every server machine) • `dwp` — Calendar Server Database service (started only with remote database configuration) • `http` — Calendar Server service
`service` [*servicename*]	Start a specified Calendar Server service or all of its services if no service name is specified. The Calendar Server 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 service 3.`admin` — Calendar Server administration service (required on every server machine) 4.`dwp` — Calendar Server database service (started only with remote database configuration) 5.`http` — Calendar Server service For more information about Calendar Server services, see the following sections: •"Calendar Server Access Control," on page 35. •"Starting and Stopping the Calendar Server," on page 56." •"Services Configuration," on page 165.
`version`	Display the version of the utility.

Table 7-25 describes the `csstart` utility command options.

Table 7-25 `csstart` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-f`	To force a specified Calendar Server service (or all currently running 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:** iPlanet recommends that you use the `stop-cal` and `start-cal` utilities instead to stop and start the Calendar Server.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.

Examples

- Start all local Calendar Server services using the default ports and in the default start order:

  ```
  csstart service
  ```

- Start the local Calendar Server HTTP service:

  ```
  csstart service http
  ```

- Check if all local Calendar Server services are started:

  ```
  csstart check
  ```

- List all local Calendar Server services that are started:

  ```
  csstart list
  ```

csstats

The `csstats` utility displays Calendar Server statistics. Commands are:

- `list` counter statistics about a specified Calendar Server subsystem.
- `version` displays the version number of the utility currently installed.

For more information about counters, see “Counters Configuration (counter.conf) File” on page 187.

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server can be running or stopped.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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

Table 7-26 describes the commands available for the `csstats` utility.

Table 7-26 `csstats` Utility Commands

Command	Description
`list [`*subsystem*`]`	List counter statistics about a specified Calendar Server subsystem or. If *subsystem* 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 (GSE) • `http` — HTTP transport • `response` — server response times • `sess` — server session status • `wcap` — Web Calendar Access Protocol
`Version`	Display the version of the utility.

Table 7-27 describes the `csstats` utility command options.

Table 7-27 `csstats` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress 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`

Table 7-27 `csstats` Utility Command Options

Option	Description
`-s` *delay*	The amount of time (in seconds) to wait before displaying each statistical lookup. The default is 1 second.

Examples

- Display basic information about counters and what types are available:

  ```
  csstats list
  ```

- List counter statistics about the HTTP service subsystem (`hpptstat`):

  ```
  csstats list http
  ```

- List counter statistics about the WCAP subsystem (`wcapstat`) every 10 seconds for one hour (3600 seconds):

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

csstop

NOTE The Calendar Server provides the `csstop` and `csstart` utilities only to provide compatibility with earlier releases.

iPlanet recommends that you use the `start-cal` and `stop-cal` commands to start and stop the Calendar Server. For more information, see "Starting and Stopping the Calendar Server," on page 56.

The `csstop` utility stops the Calendar Server. Commands are:

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

On Windows NT systems, you can also stop the Calendar Server using the Services dialog box accessed from the Control Panel.

Requirements

- You must run the utility locally on the machine where the Calendar Server is installed.
- The Calendar Server must be running.

Syntax

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


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

Table 7-28 describes the commands available for the `csstop` utility..

Table 7-28 `csstop` Utility Commands

Command	Description
`check | list` [*servicename*]	Check if a specified Calendar Server service is running or check if all services are running if a service name is not specified. Or list all Calendar Server services or a specified service and their session IDs. *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 service. • `admin` — Calendar Server administration service (admin is required on every server machine). • `dwp` — Calendar Server database service (started only with remote database configuration). • `http` — Calendar Server service.
`service` [*servicename*]	Stop a specified Calendar Server service or all of its services if no service name is specified. Calendar Server services should be started in the following order and stopped in the reverse order: 1.`ens` — a generic event registration and notification service that can be shared by other iPlanet servers. 2.`notify` — Calendar Server notification service. 3.`admin` — Calendar Server administration service (admin is required on every server machine). 4.`dwp` — Calendar Server dataBase service (only started with remote database configuration). 5.`http` — Calendar Server service. For more information about Calendar Server services, see the following sections: •“Calendar Server Access Control,” on page 35. •“Starting and Stopping the Calendar Server,” on page 56. •“Services Configuration,” on page 165.
`version`	Display the version of the utility.

Table 7-29 describes the `csstop` utility command options.

Table 7-29 `csstop` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-f`	To force a 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.

Examples

- Stop all local Calendar Server services:

  ```
  csstop service
  ```

- Stop the local Calendar Server HTTP service:

  ```
  csstop service http
  ```

- Check if all local Calendar Server services are stopped:

  ```
  csstop check
  ```

- List all local Calendar Server services that are started:

  ```
  csstop list
  ```

cstool

The cstool utility pings a Calendar Server or forces it to refresh its configuration. Commands are:

- 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.

NOTE If you modify the settings for parameters in the ics.conf file, you must stop and then restart the Calendar Server for the changes to take effect. See "Editing the ics.conf Configuration File," on page 155 for more information.

Requirements

- The Calendar Server must be running.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

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


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

Table 7-30 describes the commands available for the `cstool` utility.

Table 7-30 `cstool` Utility Commands

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

Table 7-31 describes the `cstool` utility command options.

Table 7-31 `cstool` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-h` *host*	Specifies host name of the machine on which the Calendar Server is running. The default value is set at installation and taken from the `local.hostname` parameter in the `ics.conf` file. Use this option if you are accessing a Calendar Server running on a remote machine.
`-p` *port*	The port of the specified service, or 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.

Examples

- Ping the cshttpd service:

```
cstool ping http
```

- Ping the machine with the host name calserver to see if the Calendar Server cshttpd service is listening on port 80:

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

- Force a local Calendar Server to refresh all service's configurations:

```
cstool refresh
```

csuser

The `csuser` utility manages calendar user information stored in an LDAP directory server and the Calendar Server calendar database. Commands are:

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

NOTE In the current release, iPlanet recommends that you provision user LDAP attributes using a directory service utility or a customized user provisioning tool.

For example, if you are using Netscape Directory Server, use the `ldapsearch` and `ldapmodify` utilities. For information about these utilities, see the Netscape Directory Server documentation on the following web site:

`http://docs.iplanet.com/docs/manuals/`

Requirements

- The Calendar Server can be running or stopped.
- You must run the utility locally on the machine where the Calendar Server is installed.
- The LDAP server that stores calendar user information must be running.
- On UNIX systems, you must be logged in as the user and group under which the Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`. On Windows NT systems, you must be logged in as an administrator with full administrator rights to the system.

Syntax

```
csuser [-q|-v] [-a aces] [-b basedn] [-c calid] -g givenname
  [-k yes|no] [-l langcode] -s surname -y userpassword create userid


csuser [-q|-v] [-b basedn]
  check|delete|disable|enable|list|reset [userid]
```

Table 7-32 describes the commands available for the `csuser` utility.

Table 7-32 `csuser` Utility Commands

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

Table 7-33 describes the `csuser` utility command options.

Table 7-33 `csuser` Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.

Table 7-33 `csuser` Utility Command Options *(Continued)*

Option	Description
`-q`	Run in quiet mode: • Display no information if the operation is successful (errors, if they occur, are displayed). • Suppress confirmation prompting for dangerous commands. Default is off.
`-a [`*aces*`]`	Access Control Entries (ACEs) for a specified calendar. ACEs determine who can access a calendar for group scheduling and the types of permissions they have, such as create, delete, read, and write privileges. An ACE string or Access Control List (ACL), must be enclosed in quotation marks (`""`). The default is the `calstore.calendar.default.acl` parameter in the `ics.conf` file. For details about the ACE format, see "Services Configuration," on page 165.
`-b [`*basedn*`]`	The LDAP base DN (distinguished name) of the specified user ID. The default is taken from the `local.ugldapbasedn` parameter in the `ics.conf` file.
`-c [`*calid*`]`	The calendar ID of the default calendar to associate with the specified user ID. The default is the user ID. This command does not create a calendar. For more information, see "Calendar Identifiers (calids)," on page 42.
`-g` *givenname*	The user's LDAP given name (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.
`-l [`*langcode*`]`	Language code. Default is the value of `local.sitelanguage` in `ics.conf`.
`-s` *surname*	The user's LDAP surname (last name). This option is required. There is no default.

Table 7-33 `csuser` Utility Command Options *(Continued)*

Option	Description
`-y` *userpassword*	The calendar user's password. This option is required. There is no default.

Examples

- Check if the calendar user JSmith is enabled for calendaring (if the existing calendar user has access to calendar data for this Calendar Server):

```
csuser check JSmith
```

- Create an LDAP user with the user ID `JSmith` with the given name John, surname Smith, and the calendar ID `JSmithcal`:

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

- Delete the calendar user JSmith:

```
csuser delete JSmith
```

- Disable the calendar user JSmith from logging in to the Calendar Server:

```
csuser disable JSmith
```

NOTE This command prevents JSmith from logging into the Calendar Server to access calendar data, but JSmith's data is not deleted from the local calendar database. If JSmith is currently logged into the Calendar Server, JSmith retains access to calendar data until he logs off.

- Enable JSmith for calendaring (lets existing calendar user JSmith log in to the Calendar Server):

```
csuser enable JSmith
```

- List all calendar attributes for JSmith:

```
csuser -v list JSmith
```

- List all calendar user IDs prefixed with the string `user`:

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

- Reset all calendar attributes for JSmith to the default configuration settings:

```
csuser reset JSmith
```

Chapter 8

Calendar Server Configuration

iPlanet Calendar Server configuration parameters are stored in configuration files, including `ics.conf` and `counter.conf`.

This chapter provides the following information:

- Editing the ics.conf Configuration File
- Configuration Parameters (ics.conf) File
- Counters Configuration (counter.conf) File
- Notification Messages

Editing the ics.conf Configuration File

Calendar Server configuration parameters are stored in the following file:

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

The `ics.conf` file is a ASCII text file, with each line defining a parameter and its associated value(s). The parameters are initialized during Calendar Server installation. After installation, you can edit the file using a text editor.

CAUTION Modify the settings for parameters in the `ics.conf` only as described in iPlanet documentation or as directed by an iPlanet customer support representative.

To edit the `ics.conf` file:

1. Log in as a user who has administrator rights to the system where the Calendar Server is running.

2. Change to the *server-root*`/cal/bin/config/` directory where the `ics.conf` file is located.

3. Edit parameters in the `ics.conf` file using a text editor such as `vi` on Solaris and other UNIX systems or Notepad on Windows NT systems. Conventions for parameters are:

    - All parameters must be in lower case only.

    - A parameter and its associated value(s) must be separated by an equal sign (=), with spaces or tabs allowed before or after the equal sign. For example:

      ```
      service.http.idletimeout = "120"
      ```

    - A parameter value must be enclosed in double quotation marks ("). If a parameter allows multiple values, the entire value string must be enclosed in double quotation marks. For example:

      ```
      calstore.calendar.owner.acl = "@@o^a^rsf^g;@@o^c^wdeic^g"
      ```

    - A comment line begins with an exclamation point (!). Comment lines are for informational purposes only and are ignored by the Calendar Server.

      Some parameters are released as comments, beginning with either one or two exclamation points (! or !!). To use this type of parameter, you must remove the exclamation point(s), supply a value (if needed), and then restart the Calendar Server for the parameter to take effect.

      For example, to use `!!caldb.dwp.server.[`*hostname*`].ip`, you must remove the exclamation points (!!), supply a value for *hostname*, and then restart the Calendar Server.

    - If a parameter appears more than once, the value of the last parameter listed overrides the previous value.

4. After you make changes to parameters in the `ics.conf` file, stop and then restart the Calendar Server for the new configuration values to take effect. (If you prefer, you can also stop the Calendar Server before you edit the `ics.conf` file.)

   For more information, see “Starting and Stopping the Calendar Server,” on page 56.

Configuration Parameters (ics.conf) File

Configuration parameters in the `ics.conf` file include:

- Local Configuration
- Calendar Store Configuration
- Calendar Log Information Configuration
- Services Configuration
- Alarm Notification Configuration
- Calendar Lookup Database Configuration
- Single Sign-on (SSO) Configuration
- Group Scheduling Configuration
- Database Configuration
- Calendar Server API Configuration
- Event Notification Server (ENS) Configuration
- User Interface Configuration

Local Configuration

Table 8-1 Local Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`local.authldapbasedn`	`" "`	Base DN for LDAP authentication. If not specified, `local.ugldapbasedn` is used.
`local.authldaphost`	`"localhost"`	Host for LDAP authentication. If not specified, `local.ugldaphost` is used.
`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.lookupldap.search.minwildcardsize`	`"3"`	Specifies the minimum string size for wild-card searches in an attendee lookup search. Zero (0) means always do a wild-card search.
`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 the Calendar Server is installed.

Table 8-1 Local Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`local.installeddir`	`" "`	Directory path location where the Calendar Server is installed. The default is *server-root*`/cal`.
`local.instancedir`	`"."`	Directory path location where programs and data for this instance of the Calendar Server are installed. The default is *server-root*`/cal/bin`.
`local.plugindir.path`	`" "`	Directory path location where CSAPI plug-ins for this instance of the 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 Calendar Server files, such as counters and logs.
`local.serveruid`	`"icsuser"`	UNIX User ID (UID) for Calendar Server files, such as counters and logs.
`local.sitelanguage`	`"en"`	Default language for this instance of the 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@sesta.com` if the value for this is setting is `"sesta.com"`.
`local.supportedlanguages`	`"en"`	User languages supported by this instance of the Calendar Server.
`local.ugldapbasedn`	`" "`	Base DN for LDAP user preferences. Must be specified and cannot be blank.
`local.ugldaphost`	`"localhost"`	Host name of machine that stores the LDAP user preferences.

Table 8-1 Local Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`local.ugldapicsextendeduserprefs`	`"ceColorSet, ceFontFace, ceFontSizeDelta, ceDateOrder, ceDateSeparator, ceClock, ceDayHead, ceDayTail, ceInterval, ceToolText, ceToolImage, ceDefaultAlarmStart, ceSingleCalendarTZID, ceAllCalendarTZIDs, ceDefaultAlarmEmail,ceNotifyEmail, ceNotifyEnable, ceDefaultView"`	Values for the Calendar Server 5.x 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 8-2 Calendar Store Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`calstore.anonymous.calid`	`"anonymous"`	Calendar ID (`calid`) used for anonymous logins.
`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. In the current release, the Calendar Server allows double booking of a calendar even if `resource.allow.doublebook` or `user.allow.doublebook` is set to `"no"`. To prevent double booking, run the `cscal` utility with the `-k no` option for each calendar.
`calstore.calendar.default.acl`	`"@@o^a^r^g;@@o^c^wdeic^g;@^a^fs^g;@^c^^g;@^p^r^g"`	Specifies 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 Chapter 4, "Managing Calendar Server Access Control. To specify Access Control Entries for one or more calendars using the command-line utilities, see "cscal," on page 112.
`calstore.calendar.owner.acl`	`"@@o^a^rsf^g;@@o^c^wdeic^g"`	Specifies the default access control settings for owners of a calendar. **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 8-2 Calendar Store Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`calstore.default.timezoneID`	`"America/New_York"`	Timezone ID to be used when importing files when: • A timezone ID is not supplied • A calendar timezone ID is not found • A user timezone ID is not found An invalid value causes the server to use to the GMT (Greenwich Mean Time) timezone.
`calstore.filterprivateevents`	`"yes"`	Specifies whether the Calendar Server filters (recognizes) Private and Time and Date Only (confidential) events and tasks. If `"no"`, the Calendar Server treats them the same as Public events and tasks.
`calstore.freebusy.include.defaultcalendar`	`"yes"`	Specifies whether a user's default calendar is included in user's free/busy calendar list.
`calstore.freebusy.remove.defaultcalendar`	`"no"`	Specifies whether a user's default calendar can be removed from user's free/busy calendar list.
`calstore.group.attendee.maxsize`	`"0"`	Maximum number of attendees allowed in an LDAP group when expanding an event. Value of `"0"` means to expand the group entirely.
`calstore.recurrence.bound`	`"60"`	Maximum number of events that can be created by a recurrence expansion.
`calstore.subscribed.include.defaultcalendar`	`"yes"`	Specifies whether a user's default calendar is included in the user's subscribed calendar list.
`calstore.subscribed.remove.defaultcalendar`	`"no"`	Specifies whether a user's default calendar can be removed from the user's subscribed calendar list.
`calstore.userlookup.maxsize`	`"200"`	Maximum number of results returned from LDAP lookup from user search. Value of `"0"` means no limit.
`calstore.virtualdomain.mode`	`"n"`	Specifies whether the Calendar Server is in limited virtual domain mode (`"y"` or `"n"`).

Table 8-2 Calendar Store Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
calstore.unqualifiedattendee.fmt1.type	"uid"	Specifies how the 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 the 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 the 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 8-3 Calendar Log Parameters in the ics.conf File

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 these levels (starting with the most severe): `CRITICAL`, ALERT, `ERROR`, `WARNING`, `NOTICE`, `INFORMATION`, and `DEBUG`. If you set to `CRITICAL`, the server logs the least amount of detail. If you want the server to log the most amount of detail, specify `DEBUG`. For example, if set to `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 attempts 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 8-4 Services Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`service.authcachesize`	`"10000"`	Maximum number of authenticated user IDs (UIDs) and passwords that the Calendar Server will maintain.
`service.authcachettl`	`"900"`	Number of seconds before a user UID and password are cached.
`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.
`resource.default.acl`	`"@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g"`	Specifies the default access control permissions used when a resource calendar is created.
`service.admin.alarm`	`"yes"`	Enable (`"yes"`) or disable (`"no"`) alarm notifications for administration tools.
`service.admin.calmaster.cred`	`" "`	Password of the user ID 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"`	User ID of the person designated as the Calendar Server administrator. This value is supplied at installation and is required by the installation program.
`service.admin.calmaster.overrides.accesscontrol`	`"no"`	Indicates whether the Calendar Server administrator can override access control.
`service.admin.calmaster.wcap.allowgetmodifyuserprefs`	`"no"`	Indicates whether the Calendar Server administrator can get and set user preferences using WCAP commands.
`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.

Table 8-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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.
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", enables 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	"20080"	Port for administration (csadmind) requests for Calendar Server.
service.admin.port.enable	"no"	Enable ("yes") or disable ("no") remote administration. **Note** Remote administration is planned for a future release, and the admin port is disabled by default.
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.sleeptime	"2"	Number of seconds to wait between checking for started, stopped, or ready calendar service.
service.admin.starttime	"300"	Number of seconds to wait for any calendar service to start.

Table 8-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`service.admin.stoptime`	`"300"`	Number of seconds to wait for any calendar service to stop.
`service.admin.stoptime.next`	`"60"`	Number of seconds to wait between sending stop commands to any calendar service.
`service.dnsresolveclient`	`"no"`	If `"yes"`, client IP addresses are checked against DNS if allowed HTTP access.
`service.http.admins`	`"calmaster"`	Space separated list of user IDs 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 host name 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@sesta.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.sesta.com"` would allow local HTTP access to anyone in the `sesta.com` domain. Multiple filters are separated by CR-LF (line feed).
`service.http.domainnotallowed`	`" "`	If specified and not `" "`, filter to not allow access based on TCP domains. For example, `"ALL: LOCAL.sesta.com"` would deny HTTP access to anyone in the `sesta.com` domain. Multiple filters must be separated by CR-LF (line-feed).

Table 8-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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 HTTP sessions in `cshttpd` service.
`service.http.maxthreads`	`"20"`	Maximum number of threads to service HTTP requests in `cshttpd` service.
`service.http.numprocesses`	`"1"`	Maximum number of processes to service HTTP requests.
`service.http.port`	`"80"`	Port for HTTP requests from Calendar Server users.
`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 an HTTP session in `cshttpd` service.
`service.http.sourceurl`	`" "`	Directory relative to executable where all URL references to files are stored.

Table 8-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.http.uidir.path	"html"	Directory that contains the default calendar client. If allowing only WCAP access, set to "".
service.http.renderhtml	"no"	Enables or disables HTML rendering of the contents of text fields in Calendar Express.
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.
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).

Table 8-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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).
`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 ID
`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 8-5 shows the Alarm Notification Server configuration parameters in `ics.conf`.

Table 8-5 Alarm Notification Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`alarm.diskstat.msgalarmdescription`	"percentage calendar partition diskspace available"	Description sent with insufficient disk space messages.
`alarm.diskstat.msgalarmstatinterval`	"3600"	Number of seconds between monitoring disk space.
`alarm.diskstat.msgalarmthreshold`	"10"	Percentage of available disk space that triggers sending a warning message.
`alarm.diskstat.msgalarmthresholddirection`	"-1"	Whether `alarm.diskavail.msgalarmthreshold` is above or below percentage. -1 is below and 1 is above.
`alarm.diskstat.msgalarmwarninginterval`	"24"	Number of hours between sending warning messages about insufficient disk space.
`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"	Whether `alarm.diskavail.msgalarmthreshold` is above or below the acceptable percentage. -1 is below and 1 is above.

Table 8-5 Alarm Notification Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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`	`""`	The default format used to send email alarms: `"From: %s\nTo: %s\nSubject: ALARM: %s of \"%s\" is %u\n\n%s\n"`
`alarm.responsestat.msgalarmdescription`	`"calendar service not responding"`	Description sent with no service response messages.
`alarm.responsestat.msgalarmstatinterval`	`"3600"`	Number of seconds between monitoring services.
`alarm.responsestat.msgalarmthreshold`	`"100"`	Only trigger sending a warning message if no service response.
`alarm.responsestat.msgalarmthresholddirection`	`"-1"`	Specifies whether `alarm.responsestat.msgalarmthreshold` is above or below percentage. -1 is below and 1 is above.
`alarm.responsestat.msgalarmwarninginterval`	`"24"`	Number of hours between sending warning messages about no service response sent out.

Calendar Store Configuration

Table 8-6 Store Configuration Parameters in the ics.conf File

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

Calendar Lookup Database Configuration

Table 8-7 Calendar Lookup Database Parameters in the ics.conf File

Parameter	Default Value	Description
`csapi.plugin.calendarlookup`	"n"	Enable ("y") or disable ("n") calendar lookup plug-ins.
`csapi.plugin.calendarlookup.name`	"*"	Specifies the name of a specific calendar lookup plug-in to load. If this value is an asterisk ("*"), the Calendar Server loads all plug-ins.
`caldb.cld.type`	"local"	Specifies the type of calendar lookup plug-in to use: •"local" specifies that all calendars are stored on the local machine where the Calendar Server is running, and no plug-in is loaded. •"algorithmic" loads a plug-in that uses a regular expression to determine the physical machine where a specified calendar ID is stored. This expression is specified with the setting: caldb.cld.server.*[hostname]*.regexpr •"directory" loads a plug-in that uses an LDAP directory schema entry to determine the physical machine where a specified calendar ID is stored.
`caldb.dwp.connthreshold`	"1"	Maximum number of backlogged requests before the server obtains a new network connection.

Table 8-7 Calendar Lookup Database Parameters in the ics.conf File *(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`	`"20"`	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.
`csapi.plugin.authentication`	`"n"`	If `"y"`, load only the plug-in specified in `csapi.plugin.authentication.name` or if not specified, load all authentication class plug-ins in alphabetical order. For authentication, use each of these plug-ins in alphabetical order.
`csapi.plugin.authentication.name`	`" "`	If `csapi.plugin. loadall` is `"n"` and `csapi.plugin.authentication` is `"y"`, only load this specific plug-in. If not specified or `" "`, load all authentication class plug-ins.

Table 8-7 Calendar Lookup Database Parameters in the ics.conf File *(Continued)*

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

Single Sign-on (SSO) Configuration

Table 8-8 Single Sign-on (SSO) Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`sso.appid`	`"ics50"`	Unique application ID for this Calendar Server installation. Each trusted application must also have a unique application ID. For example: `sso.appid="94043"`
`sso.appprefix`	`"ssogrp1"`	The prefix value to be used for formatting the SSO cookies. The same value needs to be used by all trusted applications, because only SSO cookies with this prefix will be recognized by the Calendar Server. The application prefix must not end with a hyphen (-), because the Calendar Server appends a hyphen to the value. For example: `sso.appprefix="d98"`

Table 8-8 Single Sign-on (SSO) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`sso.cookiedomain`	`".red.iplanet.com"`	Causes the browser to send a cookie only to servers in the domain specified here. The value must begin with a period (.). For example: `".sesta.com"`
`sso.enable`	`"1"`	Enables or disables SSO: • `"1"` (default) enables SSO functions. • `"0"` disables SSO functions. If this setting is missing from `ics.conf`, the Calendar Server ignores SSO functions.
`sso.singlesignoff`	`"true"`	If set to `"true"`, the server removes all SSO cookies for the user that match the value for `sso.appprefix` when the user logs out. If `"false"` the server removes only its SSO user cookie.
`sso.userdomain`	`" "`	Sets the domain used as part of the user's SSO authentication.

`sso.`*appid*`.url =`
`"`*verifyurl*`"`
	`" "`	Specifies the verify URL values for peer SSO hosts. A parameter is required for each trusted peer.
		appid is the application ID of a peer SSO host whose SSO cookies are to be trusted. For Calendar Server, the *appid* is `ics50`.
		`"`*verifyurl*`"` identifes the URL of the trusted peer in the format: `"http://`*host*`:`*port*`/VerifySSO?"`. Do not omit the question mark (?) after `VerifySSO`.
		host is the URL of the host, and *port* is the port number for the host.
		For example, for the Calendar Server on `sesta.com` with port number 8883:
		`sso.ics50.url = "http://sesta.com:8883/VerifySSO?"`

Group Scheduling Configuration

Table 8-9 Group Scheduling Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`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.
`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.

Database Configuration

Table 8-10 Database Configuration Parameters in the ics.conf File

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.
`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 or 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.

Table 8-10 Database Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description

`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 8-11 CSAPI Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`csapi.plugin.authentication`	`"n"`	If (`"y"`), load only the plug-in specified in `csapi.plugin.authentication.name`.
`csapi.plugin.accesscontrol`	`"n"`	Enable (`"y"`) or disable (`"n"`) Access Control plug-in.
`csapi.plugin.authentication`	`"n"`	If `"y"`, load only the plug-in specified in `csapi.plugin.authentication.name` or if not specified, load all authentication class plug-ins in alphabetical order. For authentication, use each of these plug-ins in alphabetical order.
`csapi.plugin.authentication.name`	`" "`	If `csapi.plugin. loadall` is `"n"` and `csapi.plugin.authentication` is `"y"`, only load this specific plug-in. If not specified or `" "`, load all authentication class plug-ins.
`csapi.plugin.database`	`"y"`	If `"y"`, load only the plug-in specified in `csapi.plugin.database.name` or if not specified, load all database plug-ins in alphabetical order.

Table 8-11 CSAPI Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`csapi.plugin.database.name`	`"cs_caldb_berkeley10"`	If `csapi.plugin.loadall` is `"n"` and `csapi.plugin.database` is `"y"`, load only this plug-in. If not specified or `" "`, load all database plug-ins in alphabetical order.
`csapi.plugin.datatranslator`	`"y"`	If `"y"`, load only the plug-in specified in `csapi.plugin.datatranslator.name` or if not specified, load all data translator class plug-ins in alphabetical order. For data translation, use each of these plug-ins 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 plug-ins. Otherwise, only load this specific plug-in.
`csapi.plugin.dbtranslator`	`"y"`	Enable (`"y"`) or disable (`"n"`) database-to-output format plug-ins.
`csapi.plugin.dbtranslator.name`	`"*"`	If the setting `csapi.plugin.dbtranslator` is `"y"`, then either: • load all the database-to-output format plug-ins if this value is `"*"`. • load only this specific plug-in 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 plug-ins found in the plug-ins directory. For Windows NT systems, these plug-ins have a `.dll` file name extension. For UNIX systems, these plug-ins have an `.so` extension. If `"n"`, only load the specific class of plug-ins flagged by their respective parameters. For example, set `csapi.plugin.authentication` to `yes` to load authentication class plug-ins.

Table 8-11 CSAPI Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`csapi.plugin.userprefs`	`"n"`	If `"y"`, load only the plug-in specified in `csapi.plugin.userprefs.name` or if not specified, load all user preferences class plug-ins in alphabetical order. For user preferences, use each of these plug-ins 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 plug-ins. Otherwise, only load this specific plug-in.

Event Notification Server (ENS) Configuration

The 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. The Calendar Server uses ENS to send and receive alarm notifications that include the creation, deletion, or modification of calendar events and tasks as well as general operational warning and error messages.

Table 8-12 shows the Event Notification Server (ENS) configuration parameters in `ics.conf`.

Table 8-12 Event Notification Server (ENS) Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`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 plug-in.

Table 8-12 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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.
`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.
`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`

Table 8-12 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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?
`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 todo (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.

Table 8-12 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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.
`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.

Table 8-12 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`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 8-13 User Interface Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
`ui.config.file`	`" "`	The Calendar Server can read an optional (`xml` based) configuration file at startup that can hide parts of the user interface. The Calendar Server allows only one configuration file, and the value of this setting determines the name of the configuration file to use. The Calendar Server looks for the file in the data directory where the user interface `xml` and `xslt` files are: `server-root/cal/bin/data/..` The Calendar Server provides the following files that provide customized versions of the user interface: • `nogroup_config.xml` — disables group scheduling • `ui_config.xml` — default user interface
`ui.allow.anyone`	`"yes"`	Allows Calendar Express to show and use the "Everybody" access control list (ACL).
`ui.allow.domain`	`"no"`	Allows Calendar Express to show and use the "This Domain" access control list (ACL).
`ui.proxyaddress.url`	`" "`	Specifies the proxy server address to prepend in an HTML UI JavaScript file. For example: `"https://web_portal.sesta.com/"`
`ui.base.url`	`" "`	Specifies the base server address. For example: `"https://proxyserver/"`
`ine.invitation.enable`	`"yes"`	Controls the notification for an invitation to an event: `"yes"` – Notification is sent; `"no"` – No notification is sent.
`ine.cancellation.enable`	`"yes"`	Controls the notification for a cancellation of an event: `"yes"` – Notification is sent; `"no"` – No notification is sent.

Counters Configuration (counter.conf) File

iPlanet Calendar Server counters (statistics) configuration parameters are in the following file:

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

The counter.conf file is an ASCII text file, with each line defining a counter and its parameters: name, type, size (in bytes), and description. A parameter with spaces must be enclosed in double quotation marks (" "). A comment line must begin with an exclamation point (!). Comment lines are for informational purposes only.

The first part of a counter's name identifies the counter object used with the csstats utility. For more information about the command-line utilities, see Chapter 7, "Calendar Server Command-Line Utilities."

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

This section describes the Calendar Server counter.conf parameters, including:

- Alarm Counters
- Server Response Counters
- Group Scheduling Counters
- HTTP Counters
- Database Counters
- WCAP Counters
- Disk Usage Counters
- Session Status Counters
- Authentication Counters

Alarm Counters

Table 8-14 Alarm Counters in the `counter.conf` File

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	Last time current value was set.
`alarm.timelastwarning`	TIME	4	Last time warning was sent.
`alarm.timereset`	TIME	4	Last time reset was performed.
`alarm.timestatechanged.desc`	TIME	4	Last time alarm state changed.

Disk Usage Counters

Table 8-15 Disk Usage Counters in the `counter.conf` File

Name	Type	Size	Description
`diskusage.availSpace`	GAUGE	5	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.

HTTP Counters

Table 8-16 HTTP (`httpstat`) Counters in the `counter.conf` File

Name	Type	Size	Description
`httpstat.avgConnectionTime`	GAUGE	4	Average connection response time.
`httpstat.currentStartTime`	TIME	4	When the 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.

Group Scheduling Counters

Table 8-17 Group Scheduling Counters in the `counter.conf` File

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.

Authentication Counters

Table 8-18 Authentication (`authstat`) Counters in the `counter.conf` File

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.

WCAP Counters

Table 8-19 WCAP (`wcapstat`) Counters in the `counter.conf` File

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

Database Counters

Table 8-20 Database (`dbstat`) Counters in the `counter.conf` File

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.

Server Response Counters

Table 8-21 Server Response Counters in the `counter.conf` File

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

Session Status Counters

Table 8-22 Sessions Status Counters in the `counter.conf` File

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.

Notification Messages

iPlanet Calendar Server sends the types of email messages described in Table 8-23. 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) in the following directory:

server-root`/cal/bin/config`

For example, the English version of the task alarm message format is specified in the file:

`/opt/SUNWics5/cal/bin/config/en/mail_todoalarm.fmt`

This section describes:

- Calendar Server Mail Parameters
- Special Character Sequences for Events
- Date Sub-Formatting
- Conditional Printing
- Special Character Sequences for Task Notification
- Special Character Sequences for Dates
- Simple Event Reminder Example
- Complex Event Reminder Example

Calendar Server Mail Parameters

Table 8-23 iPlanet Calendar Server Email Formats in the ics.conf File

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

Table 8-23 iPlanet Calendar Server Email Formats in the ics.conf File *(Continued)*

Message Type	Parameter	Format File (default)	Description	Recipients
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

The 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 8-24 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 (semicolon-separated list of ISO8601 date strings listing dates to exclude)
%a	rdates in iCalendar format (semicolon-separated list of ISO8601 date strings listing recurrence dates)
%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)
%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

Table 8-24 Special Character Sequences for Event Notifications *(Continued)*

Format Code	**Meaning**
`%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`	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 195.)

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 198). 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 following example:

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

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

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 8-25 Special Character Sequences for Task Notifications

Format Code	Meaning
%A	exdates in iCalendar format (semicolon-separated list of ISO8601 date strings listing dates to exclude)
%a	rdates in iCalendar format (semicolon-separated list of ISO8601 date strings listing recurrence dates)
%B	start time (also see %Z)
%C	create time
%c	task class
%D	due date / time.
%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

Table 8-25 Special Character Sequences for Task Notifications *(Continued)*

Format Code	Meaning
`%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 task does not recur)
`%S`	is the task's Sequence Number
`%s`	summary
`%t`	the status
`%U`	the UID
`%u`	URL to the task
`%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 195)

Special Character Sequences for Dates

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

Table 8-26 Special Character Sequences for Dates

Format Code	Meaning
`%a`	Abbreviated weekday name
`%A`	Full weekday name

Table 8-26 Special Character Sequences for Dates *(Continued)*

Format Code	Meaning
`%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
`%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 195.
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@sesta.com (James Smith)

Date: Wed, 15 Nov 1999 19:13:49

To: jsmith@sesta.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--
```

Appendix A

Monitoring Tools

This appendix describes system utilities you can use to monitor your server environment.

This appendix contains these sections:

- General UNIX Tools
- Various Platform-Specific Tools

General UNIX Tools

Table A-1 lists some general UNIX 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 A-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.

Table A-1 General UNIX Tools *(Continued)*

Tool	Description
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.
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.

Various Platform-Specific Tools

This section lists system monitoring tools for the following operating systems supported by iPlanet Calendar Server:

- Solaris Operating Environment
- HP-UX Operating Environment

Solaris Operating Environment

Table A-2 Solaris System Monitoring Tools

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 the Sun 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.

HP-UX Operating Environment

Table A-3 HP-UX System Monitoring Tools

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.

Appendix B

Time Zones

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

This appendix contains the following sections:

- Time Zone Administration Overview
- Managing Calendar Server Time Zones

Time Zone Administration Overview

The `timezones.ics` file contains the representation of the time zones supported by the Calendar Server. This file, which contains 92 time zones, is located in the *server-root*`/cal/bin/data` directory (for example, on Solaris systems: `opt/SUNWics5/cal/bin/data`).

At startup, the Calendar Server reads the `timezones.ics` file, generates 92 time zones, and stores them in an array in memory. Thus, all the time zones are kept in memory while the Calendar Server is running.

If a WCAP command includes a time zone `ID (tzid)`, it should reference a time zone in the array in memory. For example, if a command such as `storeevents` or `fetchcomponents_by_range` specifies a `tzid` parameter, the parameter value must be a `tzid` defined in the list of 92 time zones. The Calendar Server then returns data in that time zone, and all data will have dates applied to that time zone.

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

For more information about WCAP, refer to the *iPlanet Calendar Server Programmer's Manual*.

Managing Calendar Server Time Zones

This section describes how to add or modify a time zone:

- To Add a New Time Zone
- To Modify an Existing Time Zone

To modify the Calendar Server time zone list, you must edit the `timezones.ics` file located in the *server-root*`/cal/bin/data` directory (for example, on Solaris systems: `opt/SUNWics5/cal/bin/data`). This file contains the Calendar Server format for 92 time zones.

Time zones are identified by the `TZID` property. For example, the Calendar Server recognizes the Pacific Standard Time Zone (PST/PDT) as the `TZID America/Los_Angeles` (shown in the following example). Time zones with daylight savings usually contain two parts: STANDARD and DAYLIGHT.

Figure B-1 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;19891029T090
000Z;
  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;
```

Figure B-1 America/Los_Angeles Time Zone *(Continued)*

```
BEGIN:VTIMEZONE
  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;
  20360406T120000Z;20361026T110000Z;20370405T120000Z;20371025T110000Z
END:VTIMEZONE
```

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. For more information about time-zone properties, refer to RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar).

To Add a New Time Zone

1. Create a `TZID` name not already included in the `TZID` list.

2. Create a representation of the time zone.

3. To support the Calendar Express user interface, you must generate the `X-NSCP-TZCROSS` list for the new time zone. The Calendar Express user interface uses the dates in `X-NSCP-TZCROSS` to determine when to display a change in the time zone.

   The `X-NSCP-TZCROSS` list contains a list of dates that indicate when the time zone changes from DAYLIGHT to STANDARD and STANDARD to DAYLIGHT:

   - Odd number dates indicate STANDARD to DAYLIGHT changes.

   - Even number dates indicate DAYLIGHT to STANDARD changes.

4. Stop and then restart the Calendar Server so the new time zone will be read into memory.

To Modify an Existing Time Zone

1. Modify a `TZID` to represent the desired time zone data.
2. To support the Calendar Express client, you must generate the `X-NSCP-TZCROSS` list for the modified time zone. See the previous Step 3 for more information.
3. Stop and then restart the Calendar Server so the modified time zone will be read into memory.

Customizing Time Zones in the User Interface

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

For example, to use a customized time zone table called `US Pacific` instead of the supplied `America/Los_Angeles` table, you must provide a program to map the `US Pacific` dates to the `America/Los_Angeles` table.

NOTE Do 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 named `timezones.ics` located in the *server-root*`/cal/bin/data` directory (for example, `opt/SUNWics5/cal/bin/data`). It includes 92 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.

Figure B-2 iPlanet Calendar Server 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;19891029T110
000Z;
```

Figure B-2 iPlanet Calendar Server Time Zone Table

```
BEGIN:VCALENDAR
 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
```

Glossary

access control entry (ACE) A string that provides access control for calendars, calendar properties, and calendar components such as events and todos (tasks). An example of an ACE is`jsmith^c^wd^g`.

access control list (ACL) A set of access control entry (ACE) strings that collectively provide access control for calendars, calendar properties, and calendar components such as events and todos (tasks). An example of an ACL with three ACEs, with each ACE separated by a semi-colon is `@@o^a^r^g;@@o^c^wdeic^g;@^a^sf^g`.

alarm event An event generated and sent by the Calendar Server Event Notification Service (ENS). When an alarm event occurs, a message reminder is sent to specific recipients.

authentication The verification of a user ordinarily done using a user ID and a corresponding password. Knowledge of the password is assumed to guarantee that the user is authentic. The Calendar Server requires a directory service such as an LDAP server for user authentication.

base DN The distinguished name (DN) that identifies the starting point of a search in an LDAP directory. Also known as a search base. For example, `ou=people,o=sesta.com`.

Berkeley DB A transactional database intended for high-concurrency read-write workloads and for applications that require transactions and recoverability. The Calendar Server uses the Berkeley DB from Sleepycat Software Inc. for storing calendar data.

Calendar Express A Web-based calendar client program that provides access to the Calendar Server for end users.

calendar group A collection of several calendars that can help a user manage more than one calendar.

calendar ID (`calid`) A unique identifier associated with a calendar in the Calendar Server database. The format for a calendar ID is `userid[:calendar]` where `userid` is the user ID and `calendar` is the calendar name.

Calendar Lookup Database (A plug-in that determines the physical location of a calendar when the calendar database is distributed over several servers. The Database Wire Protocol (DWP) uses the Calendar Lookup Database plug-in to fully qualify a calendar ID (`calid`). DWP can use the returned URL to determine the location of the calendar, along with its access protocol.

Calendar Server Application Programming Interface (CSAPI) A programmatic interface that provides the capability to modify or enhance the feature set of the Calendar Server. CSAPI modules are plug-ins that are loaded from the `cal/bin/plugins` directory when the Calendar Server is started.

Calendar Access Protocol (CAP) A standard Internet protocol for calendaring based on requirements identified by the Internet Engineering Task Force (IETF).

common name (cn) An attribute that identifies the person or object defined by the entry in an LDAP directory.

component state A set of attributes that describe a calendar event such as a meeting. In WCAP, the `compstate` parameter allows fetch commands to return events by component state. For example, `compstate` might be REPLY-DECLINED (attendee has declined a meeting) or REQUEST_NEEDS-ACTION (attendee has not taken action on a meeting yet).

Calendar User Agent (CUA) An application that a calendar client uses to access the Calendar Server.

default calendar. The calendar a user first sees after logging into Calendar Express. Usually, the calendar ID of a default calendar is the same as the user's user ID. For example, `wchang@sesta.com` would have a default calendar named `wchang`.

directory service A centralized repository of directory information for use by other servers. The Calendar Server requires that a calendar user be stored in a directory server such as an LDAP server. The Calendar Server then uses the directory server for user authentication and for the storage and retrieval of user preferences. See also LDAP (Lightweight Directory Access Protocol).

distinguished name (DN) A string representation that uniquely identifies a user, system, or organization. A DN identifies an entry in an LDAP directory from which searches will occur. Also known as a search base. For example, `ou=people,o=sesta.com`.

Database Wire Protocol (DWP) A Calendar Server proprietary protocol that allows multiple servers to be linked together within the same Calendar Server system to form a distributed calendar store. The Calendar Servers uses DWP to retrieve remote data stored in the calendar database.

event A entry with an associated date and time in a calendar. For example, an event might be a new meeting or appointment on a calendar.

Event Notification Service (ENS) A generic service that accepts reports of server-level events that can be categorized and then notifies other servers that have registered interest in certain categories of events.

Extensible Markup Language (XML) A flexible programming language developed by the World Wide Web Consortium (W3C) to create common information formats and share both the format and the data on the Web, intranets, and elsewhere. XML is extensible because, unlike HTML, the markup symbols are unlimited and self-defining. The Calendar Server uses XML and XSL to generate the Calendar Express user interface.

Extensible Style Language (XSL) A language used to create style sheets for XML. XSL describes how data sent over the Web using the XML is to be presented to the user. The Calendar Server uses XSL and XML to generate the Calendar Express user interface.

Group ID (GID) On UNIX systems, the group for Calendar Server files such as counters and logs. The GID is stored in the `ics.conf` file in the `local.servergid` parameter.

GMT (Greenwich Mean Time) The mean solar time of the meridian of Greenwich, England, and the time standard against which all other time zones in the world are referred. GMT is not affected by Daylight Savings Time or Summer Time.

Group Scheduling Engine (GSE) The Calendar Server process that handles group scheduling. The GSE enables a user to schedule events with other calendar users on the same server or on a different server. The other user can then modify, cancel, or reply to the event.

High Availability (HA) A configuration that enables two Solaris servers to run a single instance of Calendar Server 5.1 that remains continuously available after any single point of failure in hardware (disk, server, or network) or software has occurred in either of the servers.

horizontal scalability The Calendar Server's capability to run on a single server or as a group of processes that are spread across multiple server with a wide variety of possible configuration options.

Hypertext Transfer Protocol (HTTP) A standard protocol that allows the transfer of hypertext documents over the Web. The Calendar Server uses HTTP as its primary transport.

instance A Calendar Server configuration of one or more server processes. Multiple Calendar Server instances can be configured per server.

ISO 8601 An ISO (International Organization for Standardization) standard that specifies the numeric representation of date and time. The Calendar Server uses ISO 8601 standard notations to represent date, time, and duration strings.

LDAP (Lightweight Directory Access Protocol) A directory service protocol defined by the Internet Engineering Task Force (IETF) used for the storage, retrieval, and distribution of information, including user profiles, distribution lists, and configuration data.

LDAP server A software server that maintains an LDAP directory and services queries to the directory. The Calendar Server uses the iPlanet Directory Server or Netscape Directory Server, which are implementations of an LDAP server.

notification A message describing an event occurrence. An example of a notification in Calendar Server is a reminder for an upcoming meeting.

notification service A service that receives subscriptions and notifications from other servers and then relays notifications to specific subscribers. The Calendar Server `csnotifyd` service sends notifications of events and todos (tasks) using the Event Notification Service (ENS) as the broker for the events.

permissions The settings that control the access to a calendar. For example, in Calendar Express, permissions include Availability, Invite, Read, Delete, and Modify. Calendar Server administrators set permissions as access control entry (ACE) strings using command-line utilities. See also access control entry (ACE) and access control list (ACL).

plug-in An accessory program that can be loaded and then used as part of the overall system. For example, the Calendar Server can use a plug-in to access a non-LDAP directory service.

resource calendar. A calendar associated with a resource such as a meeting room or equipment such as a notebook computer or overhead projector.

RFC (Request For Comments). A series of numbered international documents (such as RFC 2445, RFC 2446, and RFC 2447) that set standards that are voluntarily followed by Internet software developers. RFC standards are written informally by experts based on their technical experience and not by formal committees.

service A component of an overall system. The Calendar Server has the following services: Administration Service (csadmind). HTTP Service (cshttpd), Notification Service (csnotifyd), Event Notification Service (enpd), and Distributed Database Service (csdwpd).

server root A directory location relative to other files on a server. For example, the default Calendar Server installation on Solaris systems uses the path `/opt/SUNWics5/` as the server root.

SHTML (Server-side Include Hypertext Markup Language) An HTML file that includes embedded server-side includes (SSIs).

Single Sign-on (SSO) A authentication mechanism that enables a user to log in once and then access multiple applications. These applications form a circle of trust that use each other's cookies as verification of authority so that the user does not have to log into each application separately.

task In Calendar Express on the client side, a component of a calendar that specifies something to be done. On the server side, a task is also called a todo.

time zone A geographical region that uses the same time. There are 25 hourly time zones from -12 through +12 (GMT is 0). Each time zone is measured relative to GMT. Most time zones have localized designations in three-letter abbreviations. The Calendar Server also identifies time zones using a time zone ID (TZID) such as America/Los_Angeles or Asia/Calcutta.

todo On the server side, a a component of a calendar that specifies something to be done. In Calendar Express on the client side, a todo is called a task.

user ID (uid) A unique string identifying a user to a system. The Calendar Server identifies each user by a user ID.

Universal Principle Name (UPN) The value for a logged-in user that includes the login name combined with the domain to which the user belongs. For example, user `bill` in domain `sesta.com` has the UPN `bill@sesta.com`.

WCAP (Web Calendar Access Protocol) A high-level, command-based protocol used by clients to communicate with the Calendar Server.

Zulu time A military designation for GMT and UTC (Coordinated Universal Time).

Index