Administrator’s Guide

Sun™ ONE Calendar Server

6.0 Release

December 2003
816-6708-10

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.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

List of Tables

List of Figures

About This Guide

This guide explains how to administer Sun™ Open Net Environment (Sun ONE) Calendar Server (formerly iPlanet™ Calendar Server). Calendar Server 6.0 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
- Related Third-Party Web Site References
- Accessing Sun Documentation Online

Who Should Read This Guide

This guide is intended for Calendar Server administrators and support specialists who are responsible for administering and configuring Calendar Server.

What You Need to Know

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

- Basic administrative procedures of the Solaris™ Operating 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 Sun ONE 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 Sun ONE Calendar Server”	Provides a high-level overview of 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 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 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, “Managing the Delete Log Database”	Describes how to manage the Delete Log database (ics50deletelog.db).
Chapter 8, “Using Hosted Domains”	Describes how to set up and manage hosted (virtual) domains.
Chapter 9, “Using SSL With Calendar Server”	Describes how to set up and manage SSL.
Chapter 10, “Setting Up a High Availability (HA) Configuration”	Describes how to set up and manage an HA configuration with Calendar Server.
Chapter 11, “Calendar Server Command-Line Utilities”	Describes the Calendar Server command-line utilities, including the commands, options, values, syntax, usage rules, and examples.

Table 1 Organization of the Sun ONE Calendar Server Administrator's Guide *(Continued)*

Document Conventions Used in This Guide

This guide uses conventions for the Solaris Operating System. If you are running Calendar Server on another 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:

cal_svr_base`/opt/SUNWics5/cal`

In these situations, *cal_svr_base* represents the base or root directory Calendar Server 6.0 is installed.

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.

Braces { }

Braces { } enclose a group of items from which you must choose one item. For example, in the following syntax, you must specify either the `-a` or `-f` option:

```
{-a attr[=value] | -f filename}
```

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.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

NOTE Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Accessing Sun Documentation Online

Sun ONE Calendar Server includes documentation for administrators, developers, and end users on the following Web site:

`http://docs.sun.com/coll/S1_CalendarServer_60`

In addition to this guide, the following Calendar Server documents are available in PDF and HTML formats on the documentation Web site:

- *Sun ONE Calendar Server 6.0 Release Notes*
- *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*
- *Sun ONE Calendar Express 6.0 New Features*
- *Sun ONE Calendar Server 6.0 Programmer's Manual*
- *Sun ONE Messaging and Collaboration 6.0 Schema Reference Manual*
- *Sun ONE Messaging and Collaboration 6.0 Event Notification Service Manual*
- *Sun ONE Messaging and Collaboration 1.0 User Management Utility Installation and Reference Guide*

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

Chapter 1

Introduction to Sun ONE Calendar Server

Sun™ ONE Calendar Server is a scalable, Web-based solution for centralized calendaring and scheduling for enterprises and service providers. Calendar Server supports personal and group calendars for both events and tasks as well as calendars for resources such as conference rooms and equipment.

This chapter includes the following information:

- Calendar Server Configurations
    - Single-Server Minimal Configuration
    - Network Front-End / Database Back-End Server Configuration
    - Multiple Front-End / Back-End Server Configuration
- Calendar Server Installation and Configuration
- Calendar Server Administrators
- Calendar Server End Users
- Calendar Server Data
- Calendar Server Internal Subsystems
- Calendar Server Services
- Calendar Server APIs and SDKs

Calendar Server Configurations

Calendar Server configurations can vary depending on a site's specific requirements. This chapter describes the following three basic configurations:

- Single-Server Minimal Configuration
- Network Front-End / Database Back-End Server Configuration
- Multiple Front-End / Back-End Server Configuration

This chapter provides an overview of these configurations. For more information, see "Calendar Server Configurations for the LDAP CLD Plug-in" on page 80.

Single-Server Minimal Configuration

In a single-server minimal configuration (shown in Figure 1-1), all Calendar Server services (processes) run on the same server, either in the same CPU (processor) or across multiple CPUs. The directory server and Sun ONE Identify Server processes can run on the same server or on different servers. A single-server minimal configuration includes the following components.

Sun ONE Calendar Server

A Calendar Server instance on a single server includes the following services:

- Administration service (csadmind process) provides support for administration functions such as commands to start or stop Calendar Server, create or delete calendar users or resources, or fetch and store calendars.
- HTTP service (cshttpd process) handles incoming SHTML and WCAP requests.
- Event Notification Service (enpd and csnotifyd processes) handles event (email) notifications, if you want Calendar Server to send event notifications.

For a description of Calendar Server services, see "Calendar Server Services" on page 43.

The Database Wire Protocol (DWP) service (csdwpd process), which provides networking capability when the calendar database is on another server, is not required for a minimal configuration because the database is on the same server.

Directory Server

Calendar Server requires a directory server to authenticate users and to store user preferences. Usually, the directory server is an LDAP directory server such as Sun ONE Directory Server. However, if you prefer, you can use the Calendar Server API (CSAPI) to write a plug-in to use a non-LDAP directory server.

The directory server can run on the same server where Calendar Server is running or on a remote server.

Sun ONE Identify Server

Sun ONE Identify Server 6.1 (or later) provides the following functions:

- `commadmin` utility–Use this CLI utility to provision and manage hosted (virtual) domains, users, groups, organizations, resources, and roles for Sun ONE communications servers, including Calendar Server.

  For information about the `commadmin` utility, see the *Sun ONE Messaging and Collaboration 1.0 User Management Utility Installation and Reference Guide*.

- Single Sign-on (SSO)–You can implement SSO for Sun ONE servers, including Calendar Server and Messaging Server, using Identity Server. Identity Server serves as the SSO gateway for the Sun ONE servers. Users log in to Identity Server and then can access other the servers, as long as all servers are configured properly for SSO.

  For more information, see "Configuring SSO Through Identity Server" on page 73.

- Sun ONE LDAP Schema, v.2–Identify Server 6.1 (or later) is required if you want to use this version of the schema.

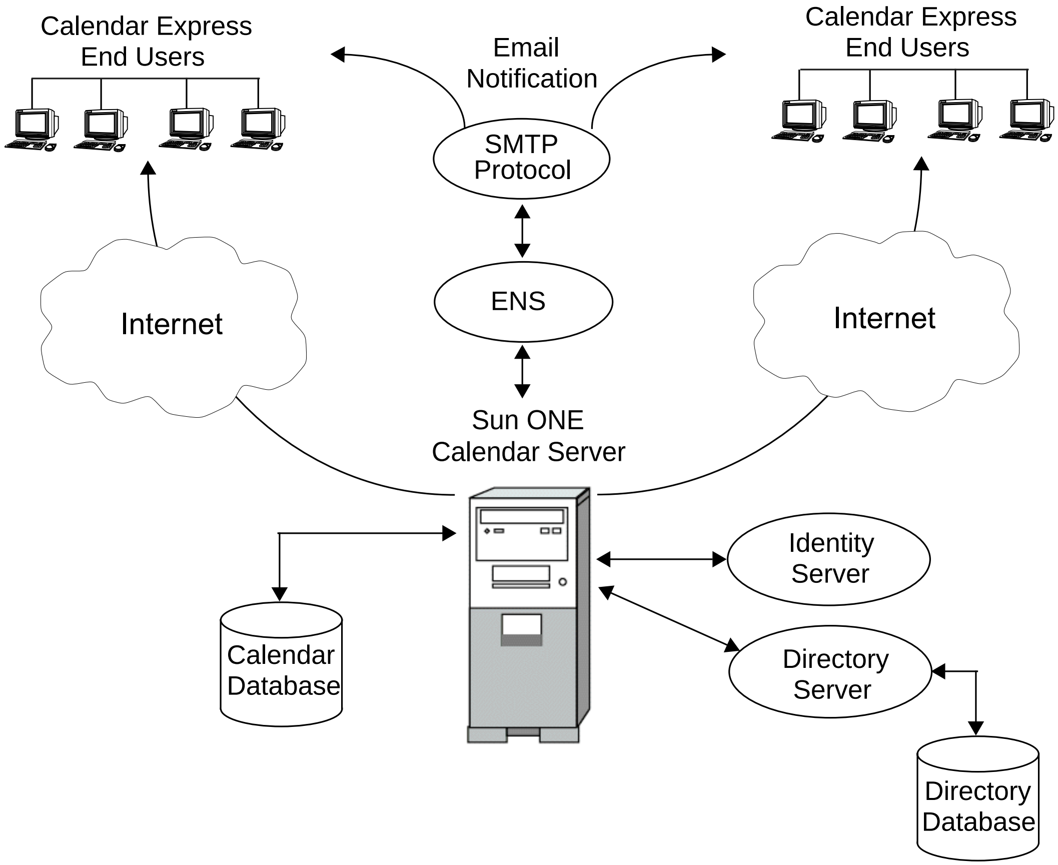
  For more information, see "Sun ONE LDAP Schema, v.2" on page 137.

Identify Server can run on the same server where Calendar Server is running or on a remote server.

End Users

End users connect to Calendar Server from client machines by using the Sun ONE Calendar Express Web user interface (UI). For information, refer to the Calendar Express online Help.

Figure 1-1 Single-Server Minimal Calendar Server Configuration



Network Front-End/Database Back-End Server Configuration

Calendar Server supports scalability by distributing a configuration over multiple front-end and back-end servers. On each server, Calendar Server services (processes or daemons) can also be distributed across multiple CPUs (or processors).

In network front-end / database back-end configuration (shown in Figure 1-2), users log in to a front-end server and connect to a back-end server using the Database Wire Protocol (DWP) service (csdwpd process). The calendar database is connected only to the back-end servers.

Sun ONE Calendar Server

Calendar Server processes run on both the front-end and back-end servers as follows:

- Users log in only to a front-end server, so each front-end server requires these services:
  - Administration service (csadmind process)
  - HTTP service (cshttpd process)
- Each back-end server is connected to a calendar database, so each back-end server requires these services:
  - Administration service (csadmind process)
  - Event Notification Service (enpd and csnotifyd processes)
  - Database Wire Protocol (DWP) service (csdwpd process) to provide networking capability to the front-end servers for the calendar database

  In this configuration, users do not log in to the back-end servers, so the HTTP service (cshttpd process) is not required.

For a description of Calendar Server services, see “Calendar Server Services” on page 43.

Directory Server

A scalable Calendar Server configuration requires a Directory Server to authenticate users and to store user preferences.

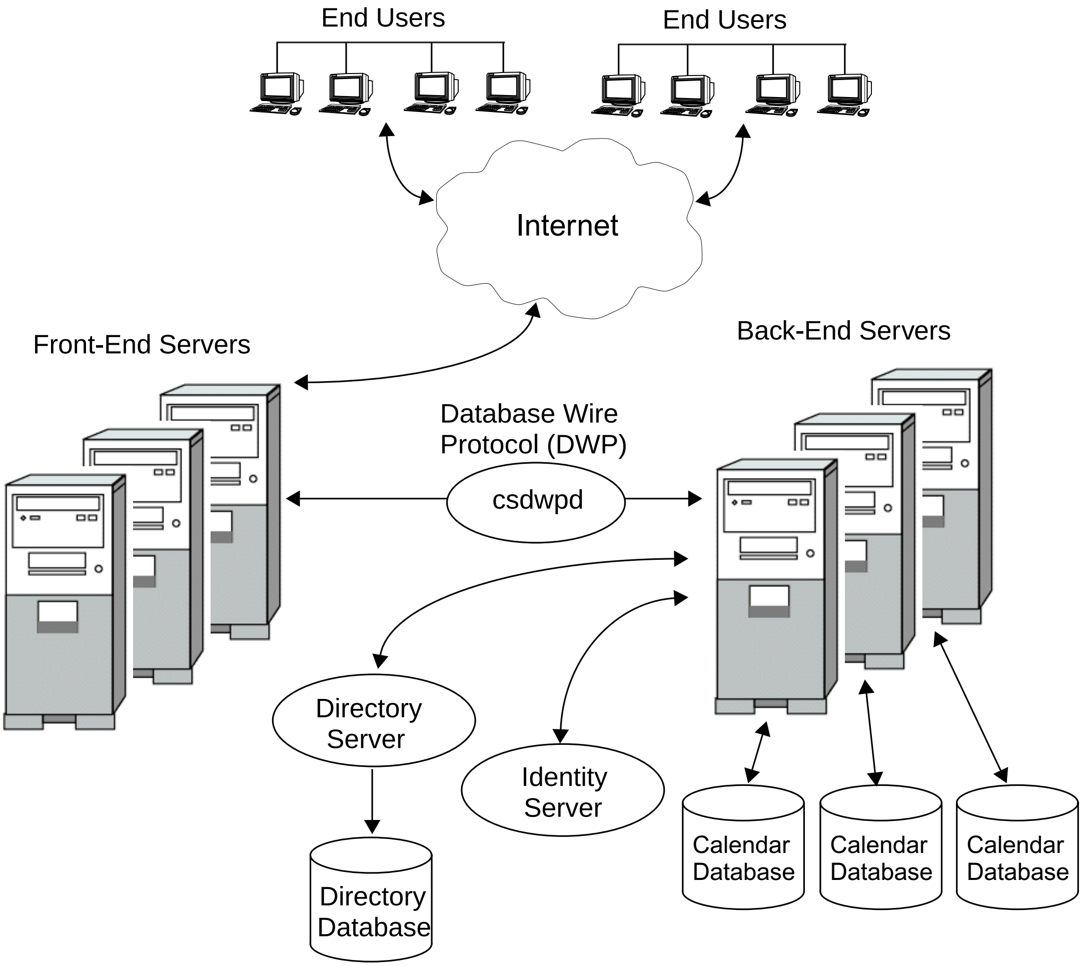
Sun ONE Identify Server

You can use Sun ONE Identify Server (release 6.1 or later) to implement Single Sign-on (SSO), to use Sun ONE LDAP Schema, v.2, or to provision and manage hosted (virtual) domains, users, groups, organizations, resources, and roles.

End Users

End users connect from client machines to a front-end server by using the Sun ONE Calendar Express Web user interface (UI). For information, refer to the Calendar Express online Help.

Figure 1-2 Network Front-End / Database Back-End Server Configuration



Multiple Front-End/Back-End Server Configuration

In a multiple front-end / back-end server configuration (shown in Figure 1-3), users log in to a specific server, and each server is connected to a calendar database. This configuration allows calendars to be geographically distributed, with each calendar residing on the server where its owner logs in to Calendar Server.

Sun ONE Calendar Server

Each front-end / back-end server requires all Calendar Server services: Administration service (csadmind process), HTTP service (cshttpd process), Event Notification Service (enpd and csnotifyd processes), and Database Wire Protocol (DWP) service (csdwpd process).

For a description of Calendar Server services, see "Calendar Server Services" on page 43.

Directory Server

A multiple front-end / back-end server configuration requires a Directory Server to authenticate users and to store user preferences.

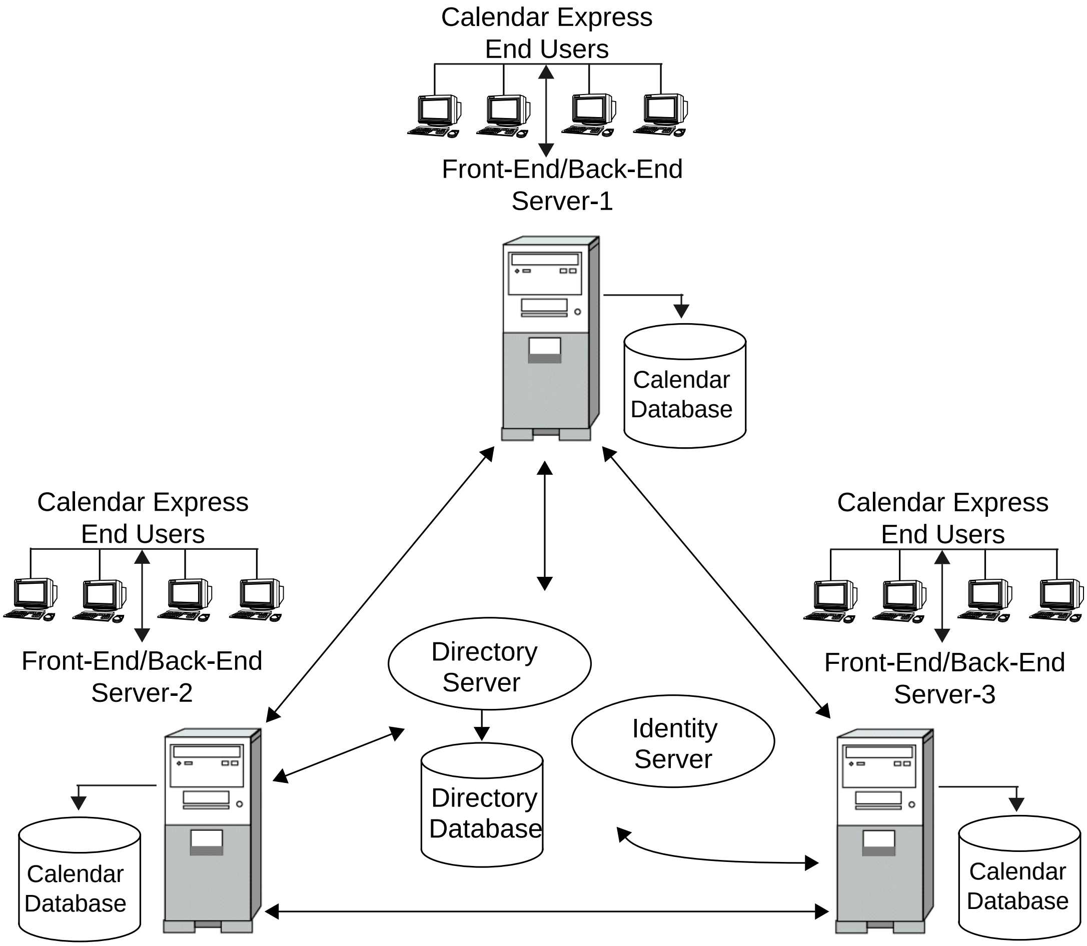
Sun ONE Identify Server

You can use Sun ONE Identify Server (release 6.1 or later) to implement Single Sign-on (SSO), to use Sun ONE LDAP Schema, v.2, or to provision and manage hosted (virtual) domains, users, groups, organizations, resources, and roles.

End Users

An end user connects from the client machine to a front-end server by using the Sun ONE Calendar Express Web user interface (UI). For information, refer to the Calendar Express online Help.

Figure 1-3 Multiple Front-End / Back-End Server Configuration


Calendar Express End Users
Front-End/Back-End Server-1
Calendar Database
Calendar Express End Users
Front-End/Back-End Server-2
Calendar Database
Directory Server
Directory Database
Identity Server
Calendar Express End Users
Front-End/Back-End Server-3
Calendar Database

Calendar Server Installation and Configuration

The installation and configuration of Sun ONE Calendar Server 6.0 (and later) on Solaris Systems has significant changes from previous Calendar Server releases. To install Calendar Server on Solaris Systems, you must use the Sun Java Enterprise System installer, which also installs other Sun component products and packages.

For information about the Java Enterprise System installer, refer to the *Sun Java Enterprise System Installation Guide*.

After you install Calendar Server using the Java Enterprise System installer, you must configure Calendar Server as follows:

1. Run the Directory Server Setup script (`comm_dssetup.pl`) to configure Sun ONE Directory Server 5.x (if the script has not already been run).
2. Run the Calendar Server configuration program (`csconfigurator.sh`) to configure your site's specific requirements and to create a new `ics.conf` configuration file. For a description of the parameters in the `ics.conf` file, see Chapter 12, "Calendar Server Configuration Parameters."

Both `comm_dssetup.pl` and `csconfigurator.sh` are located in the following directory:

```
/opt/SUNWics5/cal/sbin directory.
```

For information about running `comm_dssetup.pl` and `csconfigurator.sh`, refer to the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

Calendar Server Administrators

Administrators for Calendar Server include:

- Calendar Server Administrator (calmaster)
- Calendar Server User and Group
- Superuser (root)

Calendar Server Administrator (calmaster)

The Calendar Server administrator is the user name and associated password that can manage Calendar Server. For example, a Calendar Server administrator can start and stop Calendar Server services, add and delete users, create and delete calendars, and so on. This user has administrator privileges for Calendar Server but not necessarily for the directory server.

The default user ID for the Calendar Server administrator is `calmaster`, but you can specify a different user during Calendar Server configuration, if you prefer. After installation you can also specify a different user in the `service.admin.calmaster.userid` parameter 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 configuration, the configuration program can create it for you.

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 the LDAP server for user authentication of the user specified in service.admin.calmaster.userid. The default is “yes”.

Calendar Server User and Group

On Solaris Systems, these special accounts are the user ID and group ID under which Calendar Server runs. Sun recommends that you use the default values, `icsuser` and `icsgroup`, which are automatically created by the configuration program, if they do not exist. If you prefer, however, you can specify values other than `icsuser` and `icsgroup` when you run the Calendar Server configuration program. These values are stored in the `local.serveruid` and `local.servergid` parameters, respectively, in the `ics.conf` file.

Superuser (root)

On Solaris Systems, you must log in as or become superuser (`root`) to install Calendar Server. You can also run as superuser to manage Calendar Server using the command-line utilities. For some tasks, however, you should run as `icsuser` and `icsgroup` (or the values you have selected) rather than superuser to avoid access problems for Calendar Server files.

Calendar Server End Users

End users connect to Calendar Server from client machines by using the Sun ONE Calendar Express Web user interface (UI). This section describes:

- Creation of Calendar Server Users
- Authentication of Calendar Server Users
- Calendar Groups
- Calendar Server User Preferences

Creation of Calendar Server Users

Calendar Server users are created either manually or automatically:

- Manually — An administrator can add users to the directory server using the directory server utilities 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.

  An administrator can also use the Sun ONE Identify Server 6.1 (or later) `commadmin` utility to provision and manage Calendar Server users (as well as hosted virtual domains, groups, organizations, resources, and roles).

- Automatically (auto provisioning) — If a user already exists in the directory server, Calendar Server automatically creates a default calendar the first time the user logs in. Calendar Server uses the user's user ID for the calendar ID (`calid`) of the default calendar (unless a calendar by that name already exists).

  For auto provisioning to occur, the following criteria must be met:

  - The `local.autoprovision` parameter must be set to "yes" (which is the default) in the `ics.conf` file.
  - In hosted (virtual) domain mode, the domain must be calendar enabled. A domain is calendar enabled if it has the LDAP `icsCalendarDomain` object class.

  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 Calendar Server for the first time, Calendar Server automatically enables `TChang` for calendaring and creates a default calendar with the `calid` `TChang`.

Authentication of Calendar Server Users

Calendar Server requires a directory server such Sun ONE Directory Server to authenticate users (and to store user preferences). However, To allow access for users defined in a non-LDAP directory server, Calendar Server includes the Calendar Server API (CSAPI), which you can use to write a plug-in to access a non-LDAP directory. For information about CSAPI, refer to the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

Calendar Server User Preferences

Calendar Server allows users to customize their views of calendar data by setting user preferences attributes, which are stored in the directory server. User preferences (as opposed to Calendar Server configuration parameters) refer to the user interface representation of calendar data and include items such as user name, email address, and preferred colors to use when rendering calendar views.

For a list of preferences, refer to the `get_userprefs` and `set_userprefs` WCAP commands in the *Sun ONE Calendar Server 6.0 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 viewing. For example, a user can have a calendar group consisting of a private calendar, department calendar, and company holidays calendar. Users can also use a calendar group to select a list of calendars and view them side-by-side or invite the calendar owners to an event.

For more information about Calendar Server users, see Chapter 2, "Managing Calendar Server Users and Calendars."

Calendar Server Data

This section describes the following information about Calendar Server data:

- Calendar Server Data Format
- Import and Export of Calendar Data
- Calendar Links for Data Exchange
- Calendar Server Alarms

Calendar Server Data Format

Calendar Server data format is modeled after RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar). Calendar Server supports the following formats:

- 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. You can also use CSAPI to develop a translator DLL or shared library for the WCAP protocol. For information about WCAP and CSAPI, see the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

Import and Export of Calendar Data

Calendar data can be imported and exported in either iCalendar (`.ical`) or XML (`.xml`) format. End users import and export data using Sun ONE Calendar Express. For information, see the Calendar Express online help. Calendar Server administrators can import and export calendar data using the Calendar Server `csimport` and `csexport` utilities.

Calendar Links for Data Exchange

Calendars can be referenced as links embedded in email messages and on Web pages. Users can then click a link to view a calendar as long as the calendar allows read access, without having to log into Calendar Server. For example, the following link specifies a resource room named `Auditorium`:

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

Calendar Server Alarms

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. 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.

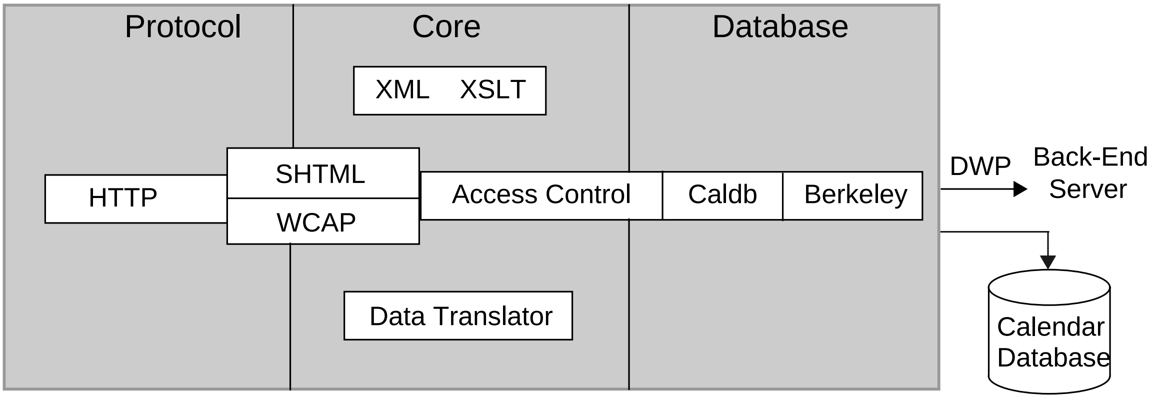
Calendar Server Internal Subsystems

Sun ONE Calendar Server includes the following internal subsystems:

- Protocol Subsystem
- Core Subsystem
- Database Subsystem

Figure 1-4 shows the logical flow through these subsystems.

Figure 1-4 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 SHTML or Web Calendar Access Protocol (WCAP) commands to submit requests:

- The Calendar Express UI uses SHTML commands, which are based on XML and XSLT specifications that generate the user interface. 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 allows you to write your own interface to Calendar Server. Using WCAP commands (.wcap extension), you can perform most server commands, except for certain administrative commands. You can use WCAP commands to return raw, unformatted calendar information, or you can also use WCAP commands to request output as XML or iCalendar wrapped in HTML.

  For information about WCAP commands, see the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

Core Subsystem

The Core subsystem includes the access control subsystem, user interface (UI) generator subsystem (either SHTML using XML and XSLT or WCAP using data translators), calendar database 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) and Proxy Authentication SDK (authSDK).

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 database, Calendar Server to provide a networking capability. For more information, see "Distributed Database Service: csdwpd" on page 44.

For information about the calendar database, refer to Chapter 5, "Managing Calendar Server Databases."

Calendar Server Services

Calendar Server services run as daemons (or processes) on Solaris Systems. These services include:

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

Administration Service: csadmind

The `csadmind` service provides a single point of authentication for administering Calendar Server, including most administration utilities (for example, start and stop commands, create and delete users, create and delete calendars, and so on). 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 Calendar Server uses HTTP as its primary transport, the `cshttpd` service listens for HTTP commands from Calendar Server end users, receives the user commands, and returns calendar data, 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) or XML format (text/xml).

Event Notification Service (ENS): csnotifyd and enpd

The ENS service consists of these individual services:

- csnotifyd–The csnotifyd service sends notifications of events and todos (tasks). The csnotifyd service also subscribes to alarm events. When an alarm event occurs, csnotifyd sends an SMTP message reminder to each recipient.
- enpd–The enpd service acts as the broker for event alarms. The enpd service 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. The enpd service also receives and stores subscriptions and cancellations of subscriptions (unsubscribe) from csnotifyd.

NOTE The enpd and csnotifyd services are not required to run on the same server as the cshttpd, csdwpd, or csadmind processes.

Distributed Database Service: csdwpd

The csdwpd service is required only on a server that has a local calendar database. The csdwpd service allows you to link front-end / back-end servers within the same Calendar Server configuration to form a distributed calendar store.

The csdwpd service runs in the background on a back-end server and accepts requests that follow the Database Wire Protocol (DWP) for accessing the calendar database. DWP is an internal protocol used to provide networking capability for the Calendar Server database.

Calendar Server APIs and SDKs

Calendar Server includes the following APIs and SDKs:

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

Web Calendar Access Protocol (WCAP)

Calendar Server supports WCAP 3.0, 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 generally follow RFC 2445, RFC 2446, and RFC 2447 specifications.

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

- Standard RFC2445 iCalendar format (`text/calendar`)
- XML format (`text/xml`)

Using WCAP commands, a Calendar Server administrator who logs in using the `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 *Sun ONE Calendar Server 6.0 Programmer's Manual*.

Calendar Server API (CSAPI)

The Calendar Server API (CSAPI) allows you to customize functional areas of Calendar Server such as user login authentication, access control, and calendar lookup. For example, by default Calendar Server uses entries in an LDAP directory server to authenticate users and to store user preferences. The CSAPI allows you to override the default Calendar Server authentication by implementing another authentication mechanism that is not based on an LDAP directory server.

For information about CSAPI, see the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

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 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, and Publish and Subscribe Dispatcher API.

For information about the ENS API, see the *Sun ONE Messaging and Collaboration 6.0 Event Notification Service Manual*.

Proxy Authentication SDK (authSDK)

Calendar Server provides the authSDK for user authentication. With authSDK, you can integrate an existing portal service with Calendar Server, thus allowing users to access various applications without requiring re-authentication. The authSDK consists of the functions packaged in a DLL / shared-object library and a header file.

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

For information about authSDK, see the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

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:

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 administrative rights to the system where Calendar Server is running. For more information, see Chapter 11, “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
  - Required LDAP mail Attribute for Calendar Server Users
  - Email Alias (mailalternateaddress Attribute)
- Calendar Identifiers (calids)
- Checking if a User is Enabled for Calendaring
- Provisioning a New User
- Creating a New Calendar

Directory Server Requirements

Calendar Server requires that a calendar user be stored in a directory server. 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 Sun ONE Directory Server. If your users are already stored in an LDAP directory, you can simply upgrade your directory server to Sun ONE Directory Server, which supports the schema extensions that allow users to access Calendar Server.

For information about installing and configuring Sun ONE Directory Server, see the following documentation Web site:

http://docs.sun.com/coll/S1_DirectoryServer_52

You can also modify your directory schema manually to allow your users to access Calendar Server data. For information, see the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems.*

Required LDAP mail Attribute for Calendar Server Users

Calendar Server 6.0 (and later) requires users to have the LDAP `mail` attribute for both user and resource calendars (for example, for meeting rooms or equipment such as a notebook computer or overhead projector). Each resource calendar must have an email address, even if the email address is not actually used.

You might specifically need to add the LDAP `mail` attribute as follows:

Existing 5.x Installation. Before you run the `cs5migrate` migration utility, add the `mail` attribute to users for both user and resource calendars. To add the `mail` attribute, use the Calendar Server csattribute utility or a utility such as the Directory Server `ldapmodify` utility.

New 6.0 Installation. Provision the LDAP `mail` attribute for existing users for both user and resource calendars using the Calendar Server `csattribute` utility or a utility such as the Directory Server `ldapmodify` utility.

If you create new calendars or users after installation, use the required `-m` *email* option to specify an email address when you run these Calendar Server utilities:

- csresource utility for new resource calendars
- csuser utility for new users

For information about the `cs5migrate` migration utility, refer to the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems.*

For information about `ldapmodify` utility, refer to the *Sun ONE Directory Server Resource Kit Tools Reference*, which is available on the following Web site:

`http://docs.sun.com/coll/S1_DirectoryServer_52`

Example: Adding the email LDAP Attribute to a Resource Calendar

The following example adds the LDAP mail attribute for a conference room named "Room100" on the `sesta.com` server. This example uses Sun ONE Messaging Server. If you are using another email server, refer to that product's documentation for the equivalent process.

1. Add the mail attribute to the LDAP server using the `csattribute` utility:

```
# ./csattribute -a mail=Room100@sesta.com add Room100
```

2. To check that the attribute has been set, use the csattribute list command with the -v (verbose) option:

```
# ./csattribute -v list Room100

...

cn=Room 100,ou=conferenceRooms,dc=sesta,dc=com has mail:
Room100@sesta.com
```

Example: Setting up the bitbucket Channel for the Resource Email

The following examples set up the bitbucket channel for Sun ONE Messaging Server or the equivalent for Sendmail for the email generated for resource calendars. These examples use a resource named "Room100" on the sesta.com server. If you don't set up the bitbucket channel (or equivalent), you will need to periodically delete the email messages sent to the resource calendar.

If you are using Sun ONE Messaging Server, perform these steps:

1. Ensure the bitbucket channel is defined in the imta.cnf file.

2. To direct messages to the bitbucket channel, create the email address for the resource using the csresource utility:

```
# ./csattribute -a mail=Room100@bitbucket.sesta.com add Room100
```

If you are using Sendmail, perform these steps:

1. In the /etc/aliases file on the appropriate host, add an entry such as:

```
# Resource/Conference room aliases
Room100: /dev/null
```

2. Add the email address for the resource to the LDAP directory using the csresource utility:

```
# ./csattribute -a mail=Room100@sesta.com add Room100
```

NOTE To enable these changes, you might also need to rebuild alias tables or configurations. Refer to the documentation for Sun ONE Messaging Server (or your email product) as well as your site's own documentation and procedures regarding changes to mail services. Sun ONE Messaging Server is available on this Web site:

http://docs.sun.com/coll/S1_MsgServer_60

Email Alias (mailalternateaddress Attribute)

If you need to setup an email aliases for a calendar user, use the LDAP mailalternateaddress attribute. The LDAP mail attribute provides the primary mail address, and the LDAP mailalternateaddress attribute is used for email aliases. Both attributes map the mail addresses to the user's calendar ID (calid).

For example, to add the mailalternateaddress attribute for a user named John Smith with these values:

- User ID (uid) and calid: johnsmith
- *password*: password for John Smith
- Email address: john.smith@sesta.com
- Email Alias's: johns@sesta.com and jsmith@sesta.com

Use these Calendar Server utility commands:

```
# ./csuser -g John -s Smith -y password -l en -m john.smith@sesta.com
-c johnsmith create johnsmith
# ./csattribute -a mailalternateaddress=johns@sesta.com add johnsmith
# ./csattribute -a mailalternateaddress=jsmith@sesta.com add johnsmith
```

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[@domain][:calendar-name]
```

where:

userid is the user ID.

domain is an optional domain name for the user. The default is the value specified by the service.defaultdomain parameter.

calendar-name is an optional 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 calendar ID cannot contain spaces and is limited to the following characters:

- Alphabetic (a-z, A-Z) and numeric (0-9) characters (non-ASCII characters are not allowed)
- Special characters: period (.), underscore (_), hyphen or dash (-), at sign (@), apostrophe (‘), percent sign (%), slash (/), or exclamation point (!)

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

Because the user ID is part of the `calid`, the user ID should not contain spaces (for example, `j smith`). A user with a user ID that contains a space can log into Calendar Server, but a space can cause subsequent problems.

Calendar Lookup Database (CLD) Plug-in

If the calendar database is distributed over several back-end servers, Calendar Server uses a plug-in to determine the actual server where a calendar is stored. Calendar Server accesses the calendar data on the back-end server using the Database Wire Protocol (DWP). DWP is an internal protocol that runs as the `csdwpd` service and provides the networking capability for the calendar database.

Calendar Server loads the plug-in, depending on the value of the `caldb.cld.type` parameter in the `ics.conf` file:

- `local` (the default) specifies that all calendars are stored on the local server where Calendar Server is running, and no plug-in is loaded.
- `directory` specifies the LDAP CLD plug-in, which uses a calendar owner’s `icsDWPHost` LDAP attribute to determine the server where a specified calendar 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. You can also use Sun ONE Directory Server utilities such as `ldapsearch` and `ldapmodify`. For information about these utilities, see the Sun ONE Directory Server documentation on the following Web site:

http://docs.sun.com/coll/S1_DirectoryServer_52

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 53.

- **Automatically -** The first time a new user logs into Calendar Server, Calendar Server automatically creates a new default calendar for the user if `local.autoprovision` is set to "yes" (the default) and the domain has been assigned the calendar service. 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.

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

  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), 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.

If your site is using the LDAP Calendar Lookup Database (CLD) plug-in, you must create a new calendar on the same back-end server where the user's calendars reside (or will reside), as indicated by the user's `icsDWPHost` LDAP attribute. If you try to create a calendar on a different back-end server, the `cscal` utility returns an error. For information about the LDAP CLD plug-in, see "Configuring the LDAP Calendar Lookup Database (CLD) Plug-in" on page 78.

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. (-k no would disable 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 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 Calendar Server:

```
csuser disable JSmith
```

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

To enable a user to access 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 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 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
```

NOTE After a calendar user has been reset, all of the calendar attributes are removed from the user's LDAP entry, including `icsCalendarUser` (object class), `icsSubscribed`, `icsCalendarOwned`, `icsCalendar`, and `icsDWPHost` (if in the LDAP CLD setup). A Calendar Server administrator will not be able to create calendars on the user's behalf.

These attributes are restored in the user's LDAP entry when:

- The user logs back into Calendar Server, or
- The Calendar Server administrator issues a `csuser enable` command for the user (although in this case, the `icsDWPHost` attribute is not restored).

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 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 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

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 the ics.conf file

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 the ics.conf file *(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 249.

Creating a Resource Calendar

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, 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
```

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”, Calendar Server deletes the calendar and displays a message that it has been deleted.

Linking to a Calendar

You can create a link to one or more user or resource calendars, as long as each calendar has the permissions set to allow read access. For example, you can embed a calendar link in a web page or email message. Other users can then view the calendar anonymously without having to log into Calendar Server.

To create a link to one or more user calendars, use this syntax:

`http://`*hostname*`:`*port*`/[command.shtml]?calid=`*calid-1*`;`*calid-2*`; ... ;`*calid-n*`&view=`*viewname*

Separate each calendar ID (calid) with a semicolon (;).

viewname can be `overview`, `dayview`, `weekview`, or `monthview`. (View can also be `yearview`, but it is not as useful.)

Note: If you are linking to only one calendar without the view (or another) option, omit `command.shtml`.

For example, to link to the default calendars for `jsmith`, enter:

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

To link to a resource calendar for an overhead projector with the `calid` `overhead_projector10`:

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

However, to link to the default calendars for `jsmith` and `tchang` and display the calendars in day view, enter:

```
http://calendar.sesta.com:8080/command.shtml?calid=jsmith;tchang&view=dayview
```

Chapter 3

Managing Calendar Server

This chapter describes how to configure and manage Sun™ ONE Calendar Server.

This chapter contains these sections:

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

You manage 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 Calendar Server is running.

For more information, see Chapter 11, "Calendar Server Command-Line Utilities" and Chapter 12, "Calendar Server Configuration Parameters"."

Starting and Stopping Calendar Server

You can start and stop Calendar Server using the `start-cal` and `stop-cal` commands. See "Using the start-cal and stop-cal Utilities" on page 68.

NOTE Calendar Server provides the `csstart` and `csstop` utilities only for compatibility with earlier releases. It is recommended that you use the `start-cal` and `stop-cal` utilities to start and stop Calendar Server.

Using the start-cal and stop-cal Utilities

The `start-cal` and `stop-cal` utilities are located in the *cal_svr_base*`/opt/SUNWics5/cal/sbin` directory. You must run these utilities on the local machine where Calendar Server is installed. For problems that might occur, see "Troubleshooting the start-cal and stop-cal Utilities" on page 69.

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

1. `enpd` — Event Notification Service (ENS)
2. `csnotifyd` — Notification Service
3. `csadmind` — Administration Service
4. `csdwpd` — Database Wire Protocol (DWP) service, the distributed database service that is started only with a remote Calendar Server database configuration
5. `cshttpd` — HTTP Service

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

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

1. Log in as a user who has administrative rights to the system.
2. Change to the *cal_svr_base*`/opt/SUNWics5/cal/sbin` directory.
3. Start Calendar Server:

   ```
   ./start-cal
   ```

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

1. Log in as a user who has administrative rights to the system where Calendar Server is running.
2. Change to the *cal_svr_base*`/opt/SUNWics5/cal/sbin` directory.
3. Stop Calendar Server:

   ```
   ./stop-cal
   ```

Troubleshooting the start-cal and stop-cal Utilities

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

- The `start-cal` utility does not start all 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 Calendar Server processes before you try to restart Calendar Server.
- The `stop-cal` utility does not stop all 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 Solaris Systems:

1. Log in as a user who has administrative rights to the system where 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 Calendar Server processes and before you restart 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 115.

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 249.

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. Sun ONE Communications servers, including Calendar Server and Messaging Server, can implement SSO as follows:

- Configuring SSO Through Identity Server
- Configuring SSO Through Communications Servers Trusted Circle Technology

Configuring SSO Through Identity Server

Sun ONE servers, including Calendar Server and Messaging Server, can implement SSO using Sun ONE Identity Server 6.1 or later.

Identity Server serves as the SSO gateway for Sun ONE servers. That is, users log in to Identity Server and then can access other Sun ONE servers, as long as the servers are configured properly for SSO.

To use SSO with Calendar Server, follow these steps:

1. Make sure that Sun ONE Identity Server and Sun ONE Directory Server are installed and configured. For information about installing and configuring these products, refer to the *Sun Java Enterprise System Installation Guide*.
2. Configure SSO for Calendar Server by setting the parameters shown in Table 3-3 and then restarting Calendar Server for the values to take effect. If necessary, remove the comment character (!) when you set each parameter.

   Note When you set the `local.calendar.sso.amnamingurl` parameter, you must use a fully qualified name for Identity Server.
3. To configure SSO for Messaging Server, refer to the *Sun ONE Messaging Server 6.0 Administrator's Guide*.
4. Users log into Identity Server using their Directory Server LDAP user name and password. (A user who logs in through another server such as Calendar Server or Messaging Server will not be able to use SSO to access the other Sun ONE servers.)
5. After logging in, users can access Calendar Server through Calendar Express using the appropriate URL. Users can also access other Sun ONE servers such as Messaging Server, if the servers are configured properly for SSO.

Table 3-3 Calendar Server Configuration Parameters for Using SSO With Identity Server

Parameter	Description
local.calendar.sso.amnamingurl	Specifies the URL of the Identity Server SSO naming service. Default is “http://*IdentityServer*:*port*/amserver/namingservice“ where *IdentityServer* is the **fully qualified name** of Identity Server, and *port* is the Identity Server port number.
local.calendar.sso.amcookiename	Specifies the name of the Identity Server SSO cookie. Default is “iPlanetDirectoryPro”.
local.calendar.sso.amloglevel	Specifies the log level for Identity Server SSO. Range is from 1 (quiet) to 5 (verbose). Default is “3“.
local.calendar.sso.logname	Specifies the name of the Identity Server SSO API log file. Default is “am_sso.log”.
local.calendar.sso.singlesignoff	Enables (“yes“) or disables (“no“) single sign-off from Calendar Server to Identity Server. If enabled, a user who logs out of Calendar Server is also logged out of Identity Server, and any other sessions the user had initiated through Identity Server (such as a Messaging Server webmail session) are terminated. Because Identity Server is the authentication gateway, single sign-off is always enabled from Identity Server to Calendar Server. Default is “yes“.

Considerations for Using SSO With Identity Server

- A calendar session is valid only as long as the Identity Server session is valid. If a user logs out of Identity Server, the calendar session is automatically closed (single sign-off).
- SSO applications must be in the same domain.
- SSO applications must have access to the Identity Server verification URL (naming service).
- Browsers must support cookies.
- If you are using the Sun ONE Portal Server gateway, set the following Calendar Server parameters:
    - `service.http.ipsecurity="no"`
    - `render.xslonclient.enable="no"`

Configuring SSO Through Communications Servers Trusted Circle Technology

When configuring SSO through Communications Servers trusted circle technology (that is, not through Identity Server), 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`.

Table 3-4 describes the Calendar Server configuration parameters for SSO through Communications Servers trusted circle technology.

Table 3-4 Calendar Server SSO Parameters Through Communications Servers Trusted Circle Technology

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".
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 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.

Table 3-4 Calendar Server SSO Parameters Through Communications Servers Trusted Circle Technology

Parameter	Description

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.

Table 3-5 describes the Messaging Server configuration parameters for SSO through Communications Servers trusted circle technology.

Table 3-5 Messaging Server SSO Parameters Through Communications Servers Trusted Circle Technology

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.

Table 3-5 Messaging Server SSO Parameters Through Communications Servers Trusted Circle Technology *(Continued)*

Parameter	Description

local.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 these items:
	• 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.

For more information about configuring Messaging Server for SSO, see the *Sun ONE Messaging Server 6.0 Administrator's Guide*.

Configuring the LDAP Calendar Lookup Database (CLD) Plug-in

The LDAP CLD plug-in provides horizontal scalability of the calendar database by allowing user and resource calendars to be distributed over a number of back-end servers for a single calendar instance. The LDAP CLD plug-in uses the `icsDWPHost` attribute to determine the back-end server where a calendar is located.

NOTE In Calendar Server 5.1.1 and later releases, the major version number of the CLD plug-in changed from 1 to 2. The minor version number is still 0. If you have written your own CLD plug-in, you must modify your plug-in to support this new major version number.

This section describes the following topics:

- How the LDAP CLD Plug-in Works
- Calendar Server Configurations for the LDAP CLD Plug-in
- Maintaining Security Between Front-End and Back-End Servers
- Improving Performance of the LDAP CLD Plug-in
- Clearing the CLD Cache
- Moving a Calendar to a Different Back-End Server

How the LDAP CLD Plug-in Works

The LDAP CLD plug-in allows the calendar database to be distributed over a number of back-end servers. Each calendar in the database is identified by a unique calendar ID (`calid`) in the following format:

```
userid[@domain][:calendar-name]
```

where:

- `userid` is a unique user ID for the Calendar Server instance.
- `domain` is an optional domain name for the user.
- `calendar-name` is an optional calendar name that is unique to the specific user.

Calendar Server accesses calendar data on a back-end server as follows:

1. When a Calendar Express end user accesses a calendar, the LDAP CLD plug-in extracts the `userid` from the calendar's `calid` and then looks up the calendar owner in the LDAP server database.

2. After finding the calendar owner, the plug-in uses the owner's `icsDWPHost` LDAP attribute to determine the host name of the back-end server where the calendar resides. This host name must be resolvable by your Domain Name Service (DNS) into a valid IP address.

3. Using the host name, Calendar Server accesses the calendar data on the back-end server using the Database Wire Protocol (DWP). DWP is an internal protocol that runs as the `csdwpd` service and provides the networking capability for the calendar database.

4. Using DWP, Calendar Server sends the calendar data to the server where the user is logged in, and then Calendar Express renders the data in the end user's browser.

NOTE If your site is using the LDAP CLD plug-in and you are using the `cscal` utility to create a new calendar, you must create the new calendar on the same back-end server where the user's calendars reside (or will reside), as indicated by the user's `icsDWPHost` LDAP attribute. If you try to create a calendar on a different back-end server, Calendar Server returns an error.

For more information, see "cscal" on page 184.

Calendar Server Configurations for the LDAP CLD Plug-in

The LDAP CLD plug-in supports the following Calendar Server configurations:

- Multiple Front-End Servers with Multiple Back-End Servers
- Multiple Front-End / Back-End Servers

In these configurations, each front-end and back-end server must:

- Be running the same Solaris Operating System.
- Be running the same Calendar Server release, including patches.
- Use the same port number for the DWP port (`service.dwp.port` parameter). The default port number is “59779”.

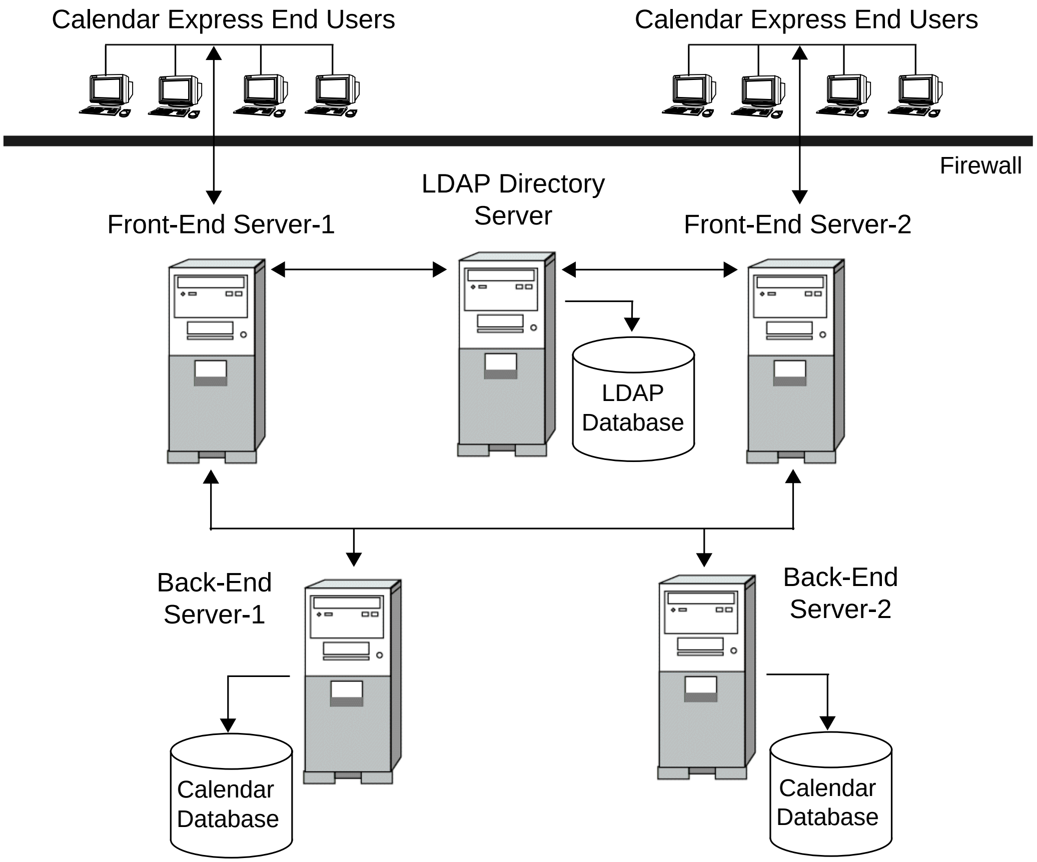
Multiple Front-End Servers with Multiple Back-End Servers

The following figure shows two front-end servers and two back-end servers running a single Calendar Server instance. You can also configure more than two front-end or back-end servers, if you wish.

This configuration allows the servers to be protected by a firewall to restrict access to the LDAP and calendar databases. The calendar database is distributed across the two back-end servers.

The front-end servers are CPU intensive, with most CPU time spent rendering calendar data for end-users. The back-end servers are disk intensive, with most CPU time spent accessing the calendar database.

Figure 3-1 Multiple Front-End Servers with Multiple Back-End Servers



Configuring a Front-End Server

To configure a front-end server, set the following parameters in the `ics.conf` file on each front-end server.

1. Enable calendar database lookup plug-ins:

```
csapi.plugin.calendarlookup = "y"
```

2. Specify that Calendar Server load all plug-ins:

```
csapi.plugin.calendarlookup.name = "*"
```

3. Set the type of calendar lookup plug-in for the LDAP CLD plug-in:

   ```
   caldb.cld.type = "directory"
   ```

4. Set the port number for the DWP service (`csdwpd`):

   ```
   service.dwp.port = "59779"
   ```

   The default is “59779”. The port number must be the same for all configured front-end and back-end servers.

5. Set the server name for each back-end server in the configuration:

   ```
   caldb.dwp.server.backend-server-1.ip = "backend-server-1"
   caldb.dwp.server.backend-server-2.ip = "backend-server-2"
   ...
   caldb.dwp.server.backend-server-n.ip = "backend-server-n"
   ```

   The server name must be fully qualified and be resolvable by your Domain Name Service (DNS) into a valid IP address. In each part of the parameter, the server name must be identical and fully qualified. For example:

   ```
   caldb.dwp.server.calendar.sesta.com.ip = "calendar.sesta.com"
   ```

   The server name must also match the name used for the `icsDWPHost` LDAP attribute for the applicable calendar owners.

6. Set the default DWP server name:

   ```
   caldb.dwp.server.default = "server-name"
   ```

   where *server-name* is the fully qualified default server name used by Calendar Server if the user or resource entry in the LDAP server database does not have an `icsDWPHost` attribute. This name must be resolvable by your Domain Name Service (DNS) into a valid IP address. For example:

   ```
   caldb.dwp.server.default = "calendar.sesta.com"
   ```

7. Restart Calendar Server for the changes to take effect.

Example Configuration Parameters for a Front-End Server

The following example shows the configuration parameters for a front-end server with two back-end servers named `calendar.sesta.com` and `calendar.siroe.com`. The default DWP server is `calendar.sesta.com`.

Code Example 3-1 LDAP CLD Configuration Parameters for a Front-End Server

```
service.dwp.port = "59779"
csapi.plugin.calendarlookup = "y"
csapi.plugin.calendarlookup.name = "*"
caldb.cld.type = "directory"
! Default DWP server
caldb.dwp.server.default = "calendar.sesta.com"
! Back-end servers
caldb.dwp.server.sesta.com.ip = "calendar.sesta.com"
caldb.dwp.server.siroe.com.ip = "calendar.siroe.com"
```

Configuring a Back-End Server

To configure a back-end server, set the following parameters in the `ics.conf` file on each back-end server.

1. Enable the DWP service (`csdwpd`) and set the DWP port number:

   ```
   service.dwp.enable = "y"
   service.dwp.port = "59779"
   ```

   The default port number is “59779”. The port number must be the same for all configured front-end and back-end servers.

2. Disable the HTTP service because it is not required on a back-end server (the Admin service should be set to the default value of `"yes"`):

   ```
   service.http.enable = "no"
   service.admin.enable = "yes"
   ```

3. Set the type of calendar lookup plug-in for the LDAP CLD plug-in:

   ```
   caldb.cld.type = "local"
   ```

4. Set `csapi.plugin.calendarlookup` to `"n"` because a back-end server does not need to lookup any calendar data:

   ```
   csapi.plugin.calendarlookup = "n"
   ```

5. Restart Calendar Server for the changes to take effect.

Example Configuration Parameters for a Back-End Server

The following example shows the configuration parameters for a back-end server.

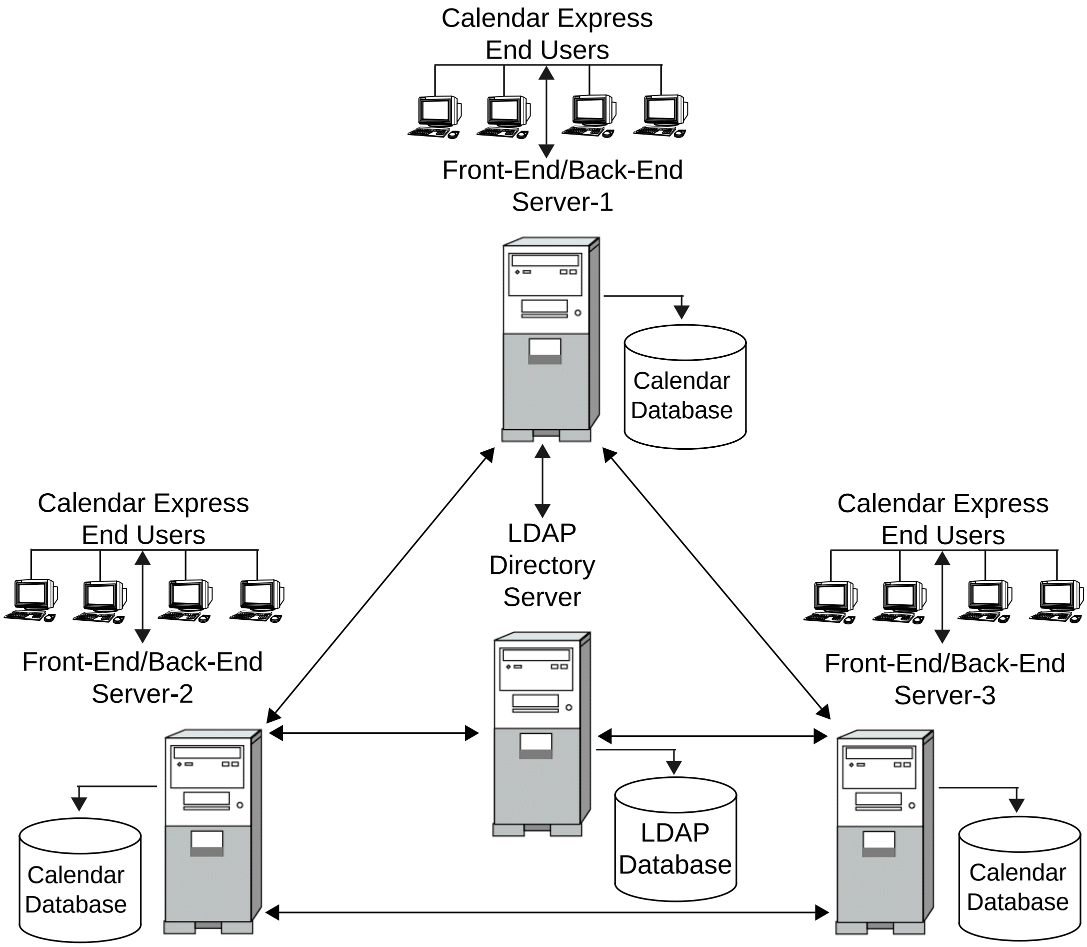
Code Example 3-2 LDAP CLD Configuration Parameters for a Back-End Server

```
service.dwp.enable = "y"
service.dwp.port = "59779"
service.http.enable = "no"
service.admin.enable = "yes"
caldb.cld.type = "local"
csapi.plugin.calendarlookup = "n"
```

Multiple Front-End/Back-End Servers

The following figure shows three front-end / back-end servers, with each server connected to a calendar database. This configuration allows calendars to be geographically distributed, with each calendar residing on the server where its owner logs into Calendar Server.

Figure 3-2 Multiple Front-End / Back-End Servers



Configuring a Front End/Back-End Server

To configure a front-end / back-end server, set the following parameters in the `ics.conf` file on each server.

1. Enable the DWP service (`csdwpd`):

```
service.dwp.enable = "y"
```

2. Set the port number for the DWP service (`csdwpd`):

   ```
   service.dwp.port = "59779"
   ```

   The default is “59779”. The port number must be the same for all configured front-end and back-end servers.

3. Enable calendar lookup plug-ins:

   ```
   csapi.plugin.calendarlookup = "y"
   ```

4. Have Calendar Server load all plug-ins:

   ```
   csapi.plugin.calendarlookup.name = "*"
   ```

5. Specify the type of calendar lookup plug-in that Calendar Server should use:

   ```
   caldb.cld.type = "directory"
   ```

6. Set the default DWP server name:

   ```
   caldb.dwp.server.default = "server-name"
   ```

   where *server-name* is the fully qualified default server name used by Calendar Server if the user or resource entry in the LDAP server database does not have an `icsDWPHost` attribute. This name must be resolvable by your Domain Name Service (DNS) into a valid IP address. For example:

   ```
   caldb.dwp.server.default = "calendar.sesta.com"
   ```

7. Set the server name for all front-end / back-end servers in the configuration, including the local server:

   ```
   caldb.dwp.server.server-1.ip = "server-1"
   caldb.dwp.server.server-2.ip = "server-2"
   ...
   caldb.dwp.server.server-n.ip = "server-n"
   ```

   The server name must be fully qualified and be resolvable by your Domain Name Service (DNS) into a valid IP address. In each part of the parameter, the server name must be identical and fully qualified. For example:

   ```
   caldb.dwp.server.calendar.sesta.com.ip = "calendar.sesta.com"
   ```

   The server name must also match the name used for the `icsDWPHost` LDAP attribute for the applicable calendar owners.

8. Restart Calendar Server for the changes to take effect.

Example Configuration Parameters for Each Front-End/Back-End Server

The following example shows the configuration parameters for each front-end / back-end server. The servers are sesta.com, siroe.com, and varrius.com. The default DWP server is sesta.com.

Code Example 3-3 LDAP CLD Configuration Parameters for Each Front-End / Back-End Server

```
service.dwp.enable = "y"
service.dwp.port = "59779"
csapi.plugin.calendarlookup = "y"
csapi.plugin.calendarlookup.name = "*"
caldb.cld.type = "directory"
! Default DWP server
caldb.dwp.server.default = "calendar.sesta.com"
! Back-end servers
caldb.dwp.server.calendar.sesta.com.ip = "calendar.sesta.com"
caldb.dwp.server.calendar.siroe.com.ip = "calendar.siroe.com"
caldb.dwp.server.calendar.varrius.com.ip = "calendar.varrius.com"
```

Maintaining Security Between Front-End and Back-End Servers

A front-end server uses the Database Wire Protocol (DWP) to communicate with a back-end server. Because DWP uses HTTP as the transport mechanism, Calendar Server 6.0 provides authentication for DWP connections between front-end and back-end servers using the configuration parameters in Table 3-6 and Table 3-7.

These parameters are optional and by default are not included in the `ics.conf` file. To use authentication for DWP connections, you must add the required parameters to the `ics.conf` file on each front-end and back-end server.

Table 3-6 Back-end Configuration Parameters for Authentication of a DWP Connection

Parameter	Description
service.dwp.admin.userid	On a back-end server, specifies the user ID that is used to authenticate a DWP connection. If a back-end server does not specify a user ID, no authentication is performed.
service.dwp.admin.cred	On a back-end server, specifies the password that is used to authenticate a DWP connection. If a back-end server does not specify a password, no authentication is performed.

Table 3-7 Front-end Configuration Parameters for Authentication of a DWP Connection

Parameter	Description
caldb.dwp.server.*back-end-server*.admin	On a front-end server, specifies the user ID that is used for authentication for a DWP connection to a back-end server, where *back-end-server* is the name of the server.
caldb.dwp.server.*back-end-server*.cred	On a front-end server, specifies the password that is used for authentication for a DWP connection to a back-end server, where *back-end-server* is the name of the server.

To set up the authentication for DWP connections between front-end and back-end servers, follow these steps:

1. In the `ics.conf` file on each front-end server, add these parameters:

   ```
   caldb.dwp.server.back-end-server.admin = "userid"
   caldb.dwp.server.back-end-server.cred = "password"
   ```

   where *back-end-server* is the name of the back-end server and *userid* and *password* are the user ID and password you want Calendar Server to use to authenticate the connection.

2. In the `ics.conf` file on each back-end server indicted by *back-end-server*, add these parameters:

   ```
   service.dwp.admin.userid = "userid"
   service.dwp.admin.cred = "password"
   ```

   where *userid* and *password* are the same user ID and password you specified on the front-end server.

When the front-end server first connects to the back-end server, it sends the user ID and password specified by the parameters. The back-end server checks the parameters, and if both parameters match, the authentication is successful. The back-end server then sends a session ID back to the front-end server. The front-end server uses the session ID in subsequent DWP commands to the back-end server.

Subsequent connections from the same front-end server do not need to be authenticated again, unless:

- The back-end server is restarted.

  or

- The session expires because of no activity between the two servers.

If you have multiple front-end and back-end server, you can use the same user ID and password for each one.

If a back-end server does not specify a user ID and password in this parameter, no authentication is performed.

Improving Performance of the LDAP CLD Plug-in

To improve the performance of Calendar Server with the LDAP CLD plug-in, make sure the following configuration parameters are set to “yes” (the default value for each parameter):

- `caldb.cld.cache.enable` enables the CLD cache option. This option stores the DWP host server information (`icsDWPHost` LDAP attribute) for calendar users and thus reduces calls to the LDAP directory server.
- `service.calendarsearch.ldap` specifies that calendar searches are done using the LDAP or a user preference plug-in.

Clearing the CLD Cache

If you are using the CLD cache option and you have updated a server name for an `ics.conf` parameter or moved a calendar to a different back-end server, you should clear the CLD cache to remove the server names. An out-of-date entry in the CLD cache can prevent a front-end server from establishing a connection to the correct back-end server or cause Calendar Server not to find a calendar after it have been moved.

To clear the CLD cache, follow these steps:

1. Stop Calendar Server.
2. Remove all files in the *cal_svr_base*`/var/opt/SUNWics5/csdb/cld_cache` directory, but do not remove the `cld_cache` directory itself.
3. Restart Calendar Server.

Moving a Calendar to a Different Back-End Server

To move a user or resource calendar from one back-end server to another back-end server, follow these steps:

1. On the original server, disable the calendar user using the csuser utility for a user calendar or csresource for a resource calendar. For example to disable the user with the user ID and `calid` `bkamdar`:

   ```
   csuser disable bkamdar
   ```

2. On the original server, export the calendar from the calendar database to a file using the csexport utility. For example:

   ```
   csexport -c bkamdar calendar bkamdar.ics
   ```

   If a user has more than one calendar, you must perform this step for each calendar.

3. Copy the exported calendar (`*.ics`) files from the original server to the new server.

4. On the new server, import the calendar from the file to the calendar database using the csimport utility. For example:

   ```
   csimport -c bkamdar calendar bkamdar.ics
   ```

   Again, you must perform this step for each of the calendars you exported.

5. On the LDAP directory server, update the calendar owner's `icsDWPHost` LDAP attribute to point to the new back-end server using the csattribute utility. To update an attribute, you must first delete the attribute and then add it with the new value. For example, to set the new server name to `sesta.com`:

   ```
   csattribute -a icsDWPHost delete bkamdar
   csattribute -a icsDWPHost=sesta.com add bkamdar
   ```

6. On the new server, enable the calendar user using the csuser utility for a user calendar or csresource for a resource calendar. For example:

   ```
   csuser enable bkamdar
   ```

7. On the new server, use the following commands to verify that the attributes are correct and that each calendar has been moved correctly. For example:

   ```
   cscal -v -o bkamdar list bkamdar
   ...
   csattribute -v list bkamdar
   ```

8. On the original server, delete each calendar you just moved. For example:

   ```
   cscal -o bkamdar delete bkamdar
   ```

   The `-o` option deletes all calendars whose primary owner is `bkamdar`.

Managing LDAP Attributes

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

NOTE	If your site is using the LDAP CLD plug-in, do not use `csattribute` to change the `icsDWPHost` attribute to specify a new back-end host server. Modifying `icsDWPHost` does not cause a new calendar to be created on the new back-end host. For more information, see "Configuring the LDAP Calendar Lookup Database (CLD) Plug-in" on page 78.

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 cssschedule utility. You must run cssschedule on the local machine where Calendar Server is installed.

Listing Entries in the GSE Queue

To list entries in the GSE queue, use the cssschedule 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 Calendar Server

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

- Listing Counter Statistics
- Monitoring 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 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 291.

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 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 3-8:

Table 3-8 Calendar Server Log Files

Service Name	Log File Name
Administration Service (csadmind)	admin.log
Distributed Database Service (csdwpd)	dwp.log
HTTP Service (cshttpd)	http.log
Notification Service (csnotifyd)	notify.log

Calendar Server log files are stored in the following default directory on Solaris Systems:

```
/var/opt/SUNWics5/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

Calendar Server provides eight levels of severity for events reported to the log files as described in Table 3-9.

Table 3-9 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.

Table 3-9 Calendar Server Log Error Severity Levels

Severity Level	Meaning
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 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 258.

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 Calendar Server. The NOTICE and INFORMATION level log events are generated during normal operation of 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 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 DWP service (csdwpd), Event Notification Service (enpd), or Notification Service (csnotifyd).

To run cstool, 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

In the current release, do not use the cstool refresh command to refresh a configuration. Instead, use the stop-cal and start-cal commands. For more information, see "Starting and Stopping Calendar Server" on page 68.

Chapter 4

Managing Calendar Server Access Control

Sun™ ONE 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:

- Secure Calendar Server Logins
- Access Control by Users
- Access Control Lists (ACLs)
- Public and Private Events and Tasks Filter
- Proxy Administrator Logins
- Configuration Parameters for Access Control
- Command-Line Utilities for Access Control

Secure Calendar Server Logins

When users log in to Calendar Server through Calendar Express, by default the authentication process does not encrypt the login information, including user names and passwords. If you want secure logins as your site, configure Calendar Server to use the Secure Sockets Layer (SSL) protocol to encrypt the login data. For more information, see Chapter 9, "Using SSL With Calendar Server."

Access Control by Users

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. 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 35.

- 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 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 Calendar" on page 65.

Access Control Lists (ACLs)

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.

Note: ACE characters are case insensitive. The default is expressed in lower case, but you will see uppercase for any characters modified through the Calendar Express GUI. No special meaning is attached to either upper or lower case.

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.
`l`	Lookup access for a domain.
`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.
`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.

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

Type	Description
z	Self-administrating access - the authenticated user is granted the ability to add or remove an Access Control E.ntry. Users with this privilege can add and remove privileges for themselves. For example, UserA may not have write access to UserB's calendar, but UserA has been granted self-administrating access to UserB's calendar. Therefore, UserA can add an Access Control E.ntry that grants himself write access to UserB's calendar. **Note:** This privilege does not allow UserA to grant other users access to UserB's calendar. For example, the self-administrating privilege does not allow UserA to grant UserC access to UserB's 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 all 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 users. Then, Calendar Server encounters a second ACE that denies `bjones` read access to this calendar. In this case, 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 users of a calendar.

Configuration Parameters for Access Control

Table 4-5 describes the configuration parameters in the `ics.conf` file that Calendar Server uses for access control. For more information see Chapter 12, "Calendar Server Configuration Parameters"."

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 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", Calendar Server treats Private and Time and Date Only events and tasks as if they are Public.

Proxy Administrator Logins

To allow administrator proxy logins for Calendar Server, perform these steps:

1. In the `ics.conf` file set, the following parameter:

   ```
   service.http.allowadminproxy = "yes"
   ```

2. Restart Calendar Server for the new value to take effect.

3. Verify that administrator proxy logins are working by using the following WCAP command:

   ```
   http://server[:port]/login.wcap?user=admin-user
   &password=admin-password&proxyauth=calendar-user
   ```

   where:

   - *server* is the name of the server where Calendar Server is running.
   - *port* is the Calendar Server port number. The default port is 80.
   - *admin-user* is the Calendar Server administrator. For example, calmaster.
   - *admin-password* is the password for *admin-user*.
   - *calendar-user* is the calid of the Calendar Server user.

   If the command is successful, Calendar Server displays the calendar for *calendar-user*. If problems occur, Calendar Server displays “Unauthorized”. Causes might be:

   - The *admin-user* does not have Calendar Server administrator privileges.
   - The admin-*password* is incorrect.
   - The *calendar-user* is not a valid Calendar Server user.

Command-Line Utilities for Access Control

Table 4-6 describes the Calendar Server command-line utilities that 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 Sun ONE Calendar Server databases, including these sections:

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

Calendar Server Database Files

By default, the Calendar Server database files (as well as the `cld_cache` and `ldap_cache` directories) are created and maintained in the following directory:

cal_svr_base`/var/opt/SUNWics5/csdb`

If you prefer, you can specify a different directory by running the Calendar Server configuration program (`csconfigurator.sh`). For information about the configuration program, refer to the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

Table 5-1describes the calendar database files:

Table 5-1 Calendar Server Database Files

File	Description
ics50calprops.db	Calendar properties for all calendars. Includes the calendar ID (`calid`), calendar name, Access Control List (ACL), and owner.
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 calendars. Journals are not implemented in the current release.
ics50caldb.conf	Database version identifier.
ics50recurring.db	Recurring events.
ics50deletelog.db	Deleted events and todos (tasks). See also Chapter 7, “Managing the Delete Log Database” on page 129.

Using the csdb Utility to Manage a Database

This section describes how to use the csdb utility to perform these functions:

- Specifying a Target Database
- Viewing Calendar Database Status
- Importing and Exporting Calendar Data
- Deleting a Calendar Database

To run the csdb utility, you must log in as a user who has administrative rights to the system where Calendar Server is running. For more information, see Chapter 11, “Calendar Server Command-Line Utilities.”

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.

Viewing Calendar Database Status

To view the status of a calendar database, use the csdb utility list command. 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
```

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. 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 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 115.

Deleting a Calendar Database

To delete a calendar database, use the csdb utility delete command. 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.

Detecting and Recovering From Database Corruption

Calendar database corruption can be caused by various reasons: system resource contention, hardware failures, application errors, database failures, and of course human error. This section describes how to detect calendar database corruption and to recover a corrupted database, including:

- Minimizing Data Loss
- Checking and Rebuilding a Calendar Database
- Using the Dump and Load Procedure to Recover a Database

Minimizing Data Loss

No matter how good a recovery procedure is, it is always better to detect database corruption early to minimize any data loss. To detect possible database corruption, follow these suggestions:

- Backup your database daily using a utility such as `csbackup`, the Sun StorEdge Enterprise Backup™ software, or Legato Networker®. For more information refer to Chapter 6, "Backing Up and Restoring Calendar Server Data."

  If you have backed up your database daily, you can minimize data loss if a recovery procedure fails and you are forced to revert to your last good database (which is the worst case scenario).

- Monitor the Calendar Server log files, including the alarm logs, for any error messages that might indicate database corruption. For information about the log files, refer to "Monitoring Calendar Server Log Files" on page 95.
- Use the `csmonitor` utility to monitor Calendar Server and to alert an administrator by e-mail if it detects problems, such as more than one transaction log file or a shortage of disk space for the calendar database. For more information, see "csmonitor" on page 211.
- Never remove any transaction log files in the database directory. The transaction log files contain the transaction updates (additions, modifications, or deletions), and removing them can corrupt the calendar database beyond recovery.

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).

Run the check command (and rebuild command, if necessary) after any event that might cause database problems. For example, if your site experiences a power failure, run the check command to determine if any database corruption occurred.

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 Calendar Server is installed.
2. Calendar Server can be either running or stopped; however, if possible, stop Calendar Server.

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

4. Change to the *cal_svr_base*`/opt/SUNWics5/cal/sbin` directory. For example, on Solaris Systems, enter:

   ```
   cd /opt/SUNWics5/cal/sbin
   ```

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 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 *cal_svr_base*`/opt/SUNWics5/cal/sbin/rebuild_db` directory.

The `rebuild` command without the `-g` option rebuilds all databases except the group scheduling engine (GSE) database. To also rebuild the GSE database, include the `-g` option.

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 calendar database

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

2. Stop Calendar Server.

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

4. Change to the *cal_svr_base*/opt/SUNWics5/cal/sbin directory. For example, on Solaris systems, enter:

```
cd /opt/SUNWics5/cal/sbin
```

Note If disk space is a problem for the sbin directory, run the rebuild command in a different directory.

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

```
./csdb rebuild /tmp/db /tmp/
```

If you don't specify a database directory, rebuild uses the database in the current directory.The /tmp/ parameter species the destination directory for the rebuilt database.

To also rebuild the GSE database, include the -g option.

The rebuild command can generate a lot of information, so consider redirecting all output, including stdout and stderr, to a file.

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 *cal_svr_base*/opt/SUNWics5/cal/sbin/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 `rebuild_db` directory to your production database.

8. If you have any share (`__db`*name*`.share`) or log (`log.*`) files from the corrupted database, move them to another directory.

9. Restart Calendar Server.

Using the Dump and Load Procedure to Recover a Database

If the `csdb rebuild` command is not successful, the next step is to use the dump and load procedure to try to recover a corrupted database. The dump and load procedure uses the Berkeley database `db_dump` and `db_load` utilities, which Calendar Server includes in the following directory:

cal_svr_base`/opt/SUNWics5/cal/tools/unsupported/bin`

The `db_dump` utility reads a database file and writes the database entries to an output file, using a format that is compatible with the `db_load` utility.

For documentation about the `db_dump` and `db_load` utilities, refer to the Sleepycat Software Web site:

`http://www.sleepycat.com/docs/utility/index.html`

Your success in recovering a database using the `db_dump` and `db_load` utilities depends on the degree of corruption of your database. You might need to try several `db_dump` options before you successfully recover your database. If your database is severely corrupted, however, recovery might not be possible, and you might need to revert to your last good backup of your database.

Calendar Database Version

Before you perform the dump and load procedure, your calendar database must be Berkeley DB version 3.2.9. If you have an earlier version, first run the `cs5migrate` utility to upgrade your calendar database.

For information about running `cs5migrate`, refer to the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

To perform the dump and load procedure

1. On Solaris Systems, log in as the user and group under which Calendar Server is running, such as icsuser and icsgroup, or as superuser (root).

2. Stop Calendar Server, if necessary.

3. Backup your corrupted database using a utility such as csbackup, the Sun StorEdge Enterprise Backup™ software, or Legato Networker®. For more information refer to Chapter 6, “Backing Up and Restoring Calendar Server Data.”

4. Dump each corrupted database file using the db_dump utility. The database files are ics50calprops.db, ics50journals.db, ics50alarms.db, ics50events.db, ics50todos.db, and ics50gse.db.

   Run db_dump using the following options, in order, until your database is recovered (or until you determine that the database can’t be recovered):

   - No options for minor database corruption.
   - -r option for moderate database corruption.
   - -R option for severe database corruption. The -R option dumps more data than the -r option, including partial and deleted records, from the corrupted database.

   For example, to run db_dump with the -r option:

   ```
   db_dump -r ics50events.db > ics50events.db.txt
   ```

5. Load the output file into a new database file using the db_load utility. For example:

   ```
   db_load new.ics50events.db < ics50events.db.txt
   ```

   Note If db_load reports an odd number of keys or data entries, edit the db_dump output file from Step 4 and remove the odd key or data entries. Then run db_load again.

6. Repeat Step 4 and Step 5 for the other corrupted database files.

7. Rebuild the recovered database files using the csdb rebuild command, as described in Rebuilding a Calendar Database.

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

   ```
   Calendar database has been rebuilt
   ```

If the `csdb rebuild` command was not successful, return to Step 4 and dump your database using the next `db_dump` option (`-r` or `-R`).

If the `db_dump -R` option does not recover your corrupted database, contact your Sun Microsystems technical support or sales account representative for assistance. You might need to revert to your last good backup of your database.

Chapter 6

Backing Up and Restoring Calendar Server Data

To back up and restore Calendar Server data in the `/var/opt/SUNWics5/csdb` directory, 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 includes these sections:

- Backing Up Calendar Server Data
- Restoring Calendar Server Data
- Using Sun StorEdge Enterprise Backup™ or Legato Networker®

CAUTION The Calendar Server 2.x and 6.0 `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 *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

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 ics50caldb.conf version file in the backup directory shows the version number of the calendar database that was backed up.

NOTE The 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 csbackup.

You can also specify a non-existent target backup directory and let csbackup create the directory for you.

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 Calendar Server is installed, and you must first stop Calendar Server. (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 calendar 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 Sun StorEdge Enterprise Backup™ or Legato Networker®

You can also use either Sun StorEdge Enterprise Backup software (formerly Solstice Backup) or Legato Networker to back up and restore Calendar Server data. The Sun StorEdge Enterprise Backup software and Legato Networker are similar, and the instructions in this section apply to both products.

Before attempting to backup Calendar Server, however, see the Sun StorEdge Enterprise Backup or Legato Networker documentation.

For the Sun StorEdge Enterprise Backup software documentation, see `http://docs.sun.com`.

This section describes:

- Calendar Server Backup / Restore Files
- Backing Up Calendar Server Data Using Sun StorEdge Enterprise Backup software or Legato Networker
- Restoring Calendar Server Data Using Sun StorEdge Enterprise Backup software or Legato Software

Calendar Server Backup/Restore Files

Calendar Server provides these files in the `/opt/SUNWics5/cal/sbin` directory to use with the Sun StorEdge or Legato backup software:

- `icsasm`–Calendar Server Application Specific Module (ASM). An ASM is a program that can be invoked by the Sun StorEdge or Legato backup software to back up and restore data.
- `legbackup.sh`–Script that calls the `csbackup` uitlity.
- `legrestore.sh`–Script that calls the `csrestore` uitlity.

Backing Up Calendar Server Data Using Sun StorEdge Enterprise Backup software or Legato Networker

To backup the calendar database using the Sun StorEdge or Legato backup software:

1. Copy the Sun StorEdge or Legato `nsrfile` binary file to the `/usr/lib/nsr` directory.
2. Create these symbolic links in the `/usr/lib/nsr` directory:

   ```
   icsasm -> /opt/SUNWics5/cal/sbin/icsasm
   nsrfile -> /usr/lib/nsr/nsrfile
   ```

3. Change to the `/opt/SUNWics5/cal/sbin` directory and run the `csbackup` utility with the `-l` option. For example:

   ```
   cd /opt/SUNWics5/cal/sbin
   ./csbackup -l
   ```

   The the `-l` option creates a backup directory image under the current directory. The files in this directory are empty and are used only to provide information to the backup program about how calendars will be stored on the backup media. If the backup directory already exists, it is synchronized with the current directory structure.

4. Use the `save` command to back up calendar data. For example:

   ```
   /usr/bin/nsr/save -s /opt/SUNWics5/cal/sbin/budir
   ```

   You can also use the Sun StorEdge or Legato backup GUI to schedule backups by setting up a client saveset to periodically backup the database.

   Notes Do not modify the `.nsr` files. These generated files contain directives that are interpreted by the `save` command and the `icsasm` ASM during the backup process.

   Calendar Server does not support the incremental backup feature. Do not use this feature because the backup directory is only an image of the folder structure and contains no actual data.

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

5. Automate the backup procedure.

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

Restoring Calendar Server Data Using Sun StorEdge Enterprise Backup software or Legato Software

To restore Calendar Server data:

1. Use the Sun StorEdge Enterprise Backup software `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

Managing the Delete Log Database

Calendar Server 6.0 includes the Delete Log database (`ics50deletelog.db`) to store deleted events and todos (tasks). In previous releases, Sun ONE Calendar Server did not maintain a database of deleted events and tasks. Users were forced to save the unique identifiers (`uid`) or recurrence identifiers (`rid`) of events or todos (tasks) to determine which components had been deleted. This limitation directly affected installations that used WCAP commands to develop a client user interface (UI).

This chapter describes:

- Creation of the Delete Log Database
- Querying the Delete Log Database
- Purging the Delete Log Database
- Using CLI Utilities for the Delete Log Database

Creation of the Delete Log Database

Calendar Server creates the Delete Log database (ics50deletelog.db) in the csdb directory along with the other Calendar Server database files. Calendar Server writes events and todos to the Delete Log database as follows:

- Non-Recurring Events and Todos

  When a non-recurring event or todo is deleted, Calendar Server removes it from the Events database (ics50events.db) or Todos database (ics50todos.db) and then writes it to the Delete Log database (ics50deletelog.db).

- Recurring Events and Todos

  When individual instances of a recurring event or task are deleted, Calendar Server writes each deleted instance of the event or task to the Delete Log database (ics50deletelog.db).

  To verify deleted components in the respective events or todos database to make sure they are actually deleted, use the fetch_deletedcomponents WCAP command.

  When all instances of a recurring event or todo are deleted, Calendar Server deletes the master component from the event or todo database and then writes it to the Delete Log database. A master component in the Delete Log database will contain the rrules, rdates, exrules, and exdates recurrence parameters.

Querying the Delete Log Database

To return entries from the Delete Log database, use the fetch_deletedcomponents WCAP command in either Expanded Mode or Compressed Mode:

- Expanded Mode (recurring parameter = 0)

  If the recurring parameter is 0, fetch_deletedcomponents returns all instances of recurring events that match the criteria, but it does not return the master component for recurring events.

- Compressed Mode (recurring parameter = 1)

  If the recurring parameter is 1, fetch_deletedcomponents returns non-recurring events and the master components for any recurring events, but it does not return individual recurring events.

If all instances in a recurring chain are deleted, the master component returns the `dtstart`, `dtend`, `rrules`, `rdates`, `exrules`, `exdates`, and `uid` parameters.

Also, `fetch_deletedcomponents` does not return master components associated with the deleted recurring instances that are still active. To return active master components, use the `fetchcomponents_by_lasmod` WCAP command. The `fetch_deletedcomponents` command should be used in conjunction with the `fetchcomponents_by_lasmod` command.

For more about WCAP commands, see the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

Purging the Delete Log Database

Calendar Server 6.0 provides both the Automatic Purge of the Delete Log Database and the Manual Purge of the Delete Log Database.

Automatic Purge of the Delete Log Database

If you wish, you can have Calendar Server 6.0 automatically purge entries in the Delete Log database.

Table 7-1 describes the parameters in the `ics.conf` file that control the automatic purge.

Table 7-1 Configuration Parameters for Automatic Purge of the Delete Log Database

Parameter	Description
service.admin.purge.deletelog	Enables ("yes") or disables ("no") the automatic purge of Delete Log database (ics50deletelog.db) entries. The default is "no".
caldb.berkeleydb.purge.deletelog.interval	Specifies the interval time in seconds to automatically purge entries in the Delete Log database (ics50deletelog.db). The default is 60 seconds.
caldb.berkeleydb.purge.deletelog.beforetime	Specifies a time in seconds before which to purge entries in the Delete Log database (ics50deletelog.db). The default is 86400 seconds (1 day).

For example, to have Calendar Server automatically purge Delete Log database entries every five minutes (600 seconds) that are more than 2 days old (172800 seconds), set parameters in Table 7-1 as follows:

```
service.admin.purge.deletelog="yes"
caldb.berkeleydb.purge.deletelog.interval=600
caldb.berkeleydb.purge.deletelog.beforetime=172800
```

After you set these parameters, restart Calendar Server for the new values to take effect.

Manual Purge of the Delete Log Database

To manually purge entries in the Delete Log database (ics50deletelog.db), use the cspurge utility:

`cspurge -e` *endtime* `-s` *starttime*

where *endtime* and *starttime* specify the ending and starting times in Zulu time (also referred to as GMT or UTC).

To run `cspurge` on Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (defaults are `icsuser` and `icsgroup`) or as `root`.

For example, to purge entries from July 1, 2003 through July 31, 2003:

```
cspurge -e 20030731T235959Z -s 20030701T120000Z
```

For more information, see cspurge in Chapter 11, “Calendar Server Command-Line Utilities.”

Using CLI Utilities for the Delete Log Database

Table 7-2 lists the Calendar Server command-line utilities that support the Delete Log database (`ics50deletelog.db`):

Table 7-2 Utilities that Support the Delete Log Database

Utility	Description
cspurge	Allows the manual purge of entries in the Delete Log database.
csbackup and csrestore	Supports the backup and restore of the Delete Log database.
csstats	Reports Delete Log database statistics.
csdb	Supports the rebuild, recover, and check operations on the Delete Log database.
cscomponents	Lists (read-only) the number of entries in the Delete Log database.

For more information, including the syntax for these utilities, see Chapter 11, “Calendar Server Command-Line Utilities.”

Chapter 8

Using Hosted Domains

Sun ONE Calendar Server 6.0 supports hosted (or virtual) domains. In a hosted domain installation, each domain shares the same instance of Calendar Server, which allows multiple domains to exist on a single server. Each domain defines a name space within which all users, groups, and resources are unique. Each domain also has a set of attributes and preferences that you specifically set.

This chapter describes these topics:

- Overview of Hosted Domains
  - Organization of the LDAP Directory
  - Calendar Server Logins
  - Cross Domain Searches
  - Support for a Calendar Server Legacy Installation
- Creation and Management of Hosted Domains
  - Running the Directory Server Setup Script
  - Creating New Domains
  - Using Domains Created by Messaging Server
  - Setting Domain Specific Attributes and Preferences
  - Provisioning New Calendar Server Users
- Hosted Domain Configuration Parameters
- Using WCAP Commands
- Migration to a Hosted Domain Environment

Overview of Hosted Domains

This section provides an overview of hosted domains, including:

- Organization of the LDAP Directory
- Calendar Server Logins
- Cross Domain Searches
- Support for a Calendar Server Legacy Installation

Organization of the LDAP Directory

With a hosted domain installation, the LDAP directory is organized into distinct, non-intersecting sections, each of which represents a domain found in the Domain Name System (DNS). Each domain includes unique users, groups, and resources. A distinguished name (DN) describes the root of each domain.

Calendar Server 6.0 (or later) supports both of these LDAP directory schema versions for hosted domains:

- Sun ONE LDAP Schema, v.2 (compatibility or native mode)
- Sun ONE LDAP Schema, v.1

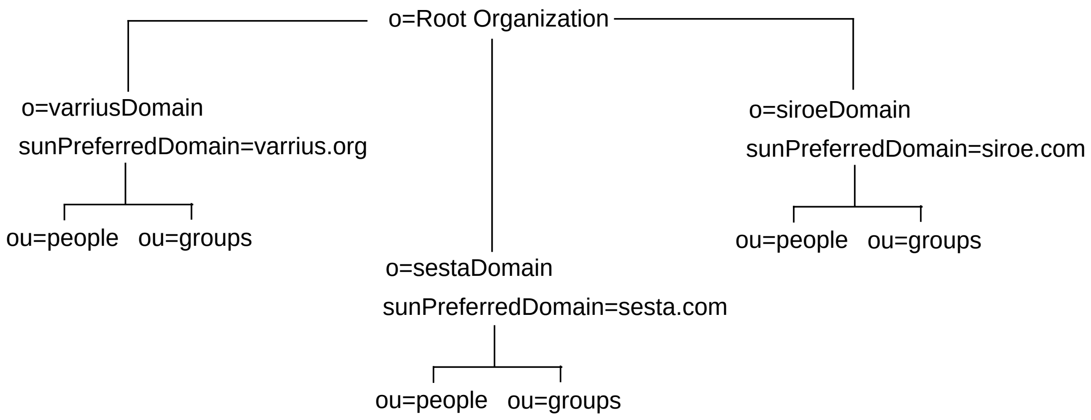
NOTE When you run the Directory Server Setup script (comm_dssetup.pl), you can choose either LDAP Schema, v.1 or LDAP Schema, v.2. Several considerations are:

- **New Installation**. If your site is installing Sun ONE Calendar Server 6.0 as a new installation, use LDAP Schema, v.2.
- **Upgrade**. If your site is upgrading from Calendar Server 5.x, use the schema version as follows:
    - If you want to use Sun ONE Identity Server 6.1 features such as the `commadmin` utility or Single Sign-on (SSO), choose LDAP Schema, v.2.
    - If you don’t want to use Identity Server 6.1 features, you can use either version. However, Sun recommends that you use LDAP Schema, v.2, if possible.

Sun ONE LDAP Schema, v.2

Figure 8-1 shows an LDAP directory organization for a hosted domain installation that uses Sun ONE LDAP Schema, v.2.

Figure 8-1 LDAP Directory Organization Using LDAP Schema v.2



LDAP Schema v.2 uses a flat LDAP directory organization. For a hosted domain installation, the first level entries (`varriusDomain`, `sestaDomain`, and `siroeDomain` in the figure) must be parallel in the directory organization. These entries cannot be nested.

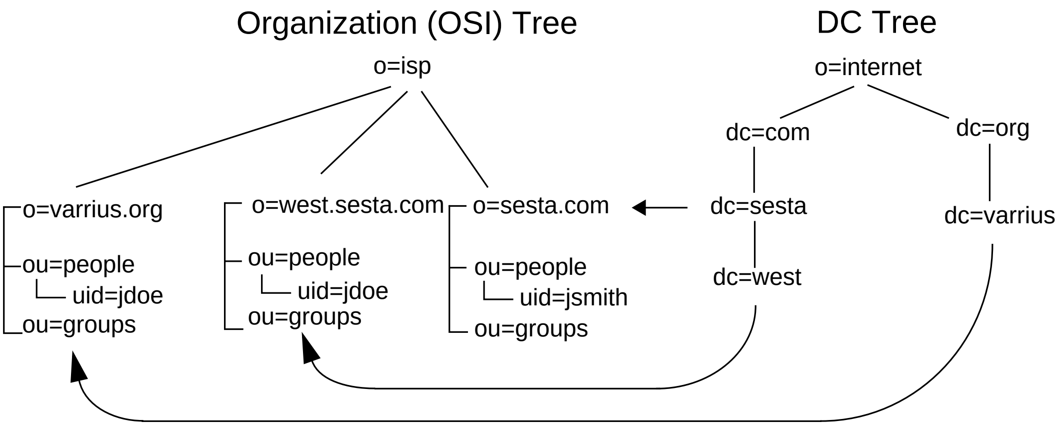
If you want to use Sun ONE Identity Server features such as the `commadmin` utility or Single Sign-on (SSO), LDAP Schema v.2 is required.

Sun ONE LDAP Schema, v.1

Figure 8-2 shows an LDAP directory organization for a hosted domain installation that uses Sun ONE LDAP Schema, v.1. This organization includes two trees (or nodes) for domain management:

- DC tree
- Organization (OSI) tree

Figure 8-2 LDAP Directory Organization Using LDAP Schema v.1



The DC tree (node) is similar to the DNS, which determines a domain entry given the domain name. The `inetdomainbasedn` LDAP attribute points to the base DN, which is the root of the domain's users, resources and groups in the OSI tree (node). Within each domain, the identifiers for Calendar Server users, resources, and groups must be unique.

In a hosted domain installation using LDAP Schema v.1, a directory search requires these two steps to find an entry:

1. In the DC tree, the search operation locates the domain entry that contains the value of the DN pointing to the base DN (`inetDomainBaseDN` attribute) of domain in the OSI tree.

2. In the OSI tree, the search operation locates the domain entry and then searches from that entry's base DN to find the user, resource, or group within the domain.

Calendar Server Logins

For a hosted domain installation, each user must have a unique user ID (`uid`) within a domain. A login to Calendar Server uses the following format:

userid[*@domain-name*]

If *domain-name* is omitted, Calendar Server uses the domain name specified by the `service.defaultdomain` parameter in the `ics.conf` file. Thus, if a user is logging into the default domain, only the *userid* is required.

For an installation with a directory that doesn't follow the organization shown in Figure 8-2, *domain-name* is not required. If a domain name is specified, it will be ignored.

When a new user logs into Calendar Server for the first time, Calendar Server automatically provisions the user if `local.autoprovision` is set to "yes" (the default) and the domain has been assigned the calendar service. The login permission is based on the `icsStatus` or `icsAllowedServiceAccess` attribute. For more information, see Table 11-17.

Cross Domain Searches

By default, users can search only within their domain for users and groups to invite to events. Cross domain searches, however, allow users in one domain to search for users and groups in other domains, as long as these requirements are met:

- Each domain can specify an access control list (ACL) in the `domainAccess` property of the `icsExtendedDomainPrefs` attribute that grants or denies cross domain searches from other domains. Thus, a domain can allow or disallow either specific domains, or all domains, from searching it. For a description of `domainAccess`, see Table 11-16 on page 202. For general information about ACLs, see "Access Control Lists (ACLs)" on page 101.
- Each domain can specify the external domains its users can search. The `icsDomainNames` LDAP attribute specifies the external domains that a domain's users can search when looking for users and groups (as long as the ACL for the external domain allows the search). For example, if

icsDomainNames for the various.org domain lists sesta.com and siroe.com, users in various.org can perform cross domain searches in sesta.com and siroe.com. For a description of icsDomainNames, see Table 11-17 on page 205.

To set the icsDomainNames and icsExtendedDomainPrefs LDAP attributes, use the Calendar Server csdomain utility. If you add or update domain LDAP attributes using csdomain (or another utility such as commadmin or ldapmodify), restart Calendar Server for the new values to take effect.

Support for a Calendar Server Legacy Installation

Calendar Server 6.0 supports an existing or legacy Calendar Server 5.x installation. In this case, the following parameter must be set to “no” in the ics.conf file:

```
service.virtualdomain.support = "no"
```

You will, however, need run the cs5migrate utility to migrate Calendar Server 5.x to 6.0. For migration information, see the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

If you decide to migrate a Calendar Server 5.x installation to use hosted domains, you must also run the csvdmig utility, See “Migration to a Hosted Domain Environment” on page 146 for more information.

Creation and Management of Hosted Domains

This section provides the following information about creating and managing hosted domains:

- Running the Directory Server Setup Script
- Creating New Domains
- Using Domains Created by Messaging Server
- Provisioning New Calendar Server Users
- Calendar Server Logins
- Cross Domain Searches

Running the Directory Server Setup Script

The Directory Server Setup (`comm_dssetup.pl`) script configures Sun ONE Directory Server 5.x for Calendar Server 6.0 (and Messaging Server 6.0). You run `comm_dssetup.pl` after you install Calendar Server 6.0 using the Sun Java Enterprise System installer and before you run the Calendar Server configuration program (`csconfigurator.sh`).

The `comm_dssetup.pl` script allows you to select these options:

- Directory Server 5.x installation directory path and instance you want to use for Calendar Server 6.0 (and Messaging Server 6.0).
- Directory Manager Distinguished Name (DN).
- Whether Directory Server 5.x will be used for users and groups. If yes, you must also specify the DC tree base suffix and a User and Group base suffix for your Organization tree.
- Whether to use Sun ONE LDAP Schema v.1 or v.2 (either compatibility mode or native mode). See Organization of the LDAP Directory for information about these schemas.
- To update your schema according to the version you have selected.
- To add Directory Server indexes to improve the efficiency of directory searches.

For information about `comm_dssetup.pl`, see the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

Creating New Domains

To create a new domain, use one of these utilities:

- Sun ONE Identity Server `commadmin` utility–To create and manage hosted domains when you are using LDAP Schema, v.2. For information about the `commadmin` utility, refer to the *Sun ONE Messaging and Collaboration 1.0 User Management Utility Installation and Reference Guide*.
- Calendar Server csdomain utility–To create and manage a new hosted domain in the LDAP directory when you are using either LDAP Schema, v.1 or LDAP Schema, v.2. This utility allows you to add, delete, and list Calendar Server attributes in the `icsCalendarDomain` object class and their associated values in the LDAP directory for a specific domain.

NOTE Use `csdomain` to create domains only if you don't want to use the Identity Server `commadmin` utility to manage the domain.

For LDAP Schema, v.1, both the DC Tree and the OSI tree (that is, the node to which the domain points) must already be present in the LDAP directory server, as shown in Figure 8-2. The csdomain utility does not create these trees.

Calendar Server does not support the Identity Server Console to create domains.

Using Domains Created by Messaging Server

If Sun ONE Messaging Server has already created a hosted domain, Calendar Server can provision users in that domain. To use a domain created by Sun ONE Messaging Server, follow these steps:

1. Add the `icsCalendarDomain` object class to the `o=internet` domain entry in the directory server and set up your domain entries with your Calendar Server users in their respective domains. Also set `icsStatus` to "active" and `domainAccess` to the ACL you want to use for access control. For an example, see Code Example 8-1.

   To modify the LDAP directory, use the Directory Server `ldapmodify` tool. For information about using `ldapmodify`, refer to the *Sun ONE Directory Server Resource Kit 5.2 Tools Reference*.

2. If you are migrating from Calendar Server 5.x, run these utilities (if you have not already run them):

   - Run `cs5migrate` utility to migrate the installation to Calendar Server 6.0.
   - Run `csvdmig` utility to migrate the installation to use hosted domains.

For information about running the migration utilities, refer to the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

Code Example 8-1 Modifying the LDAP Directory Server

```
dn:dc=sesta,dc=com,o=internet
changetype: modify
add: objectclass
objectClass: icsCalendarDomain

-

add: icsStatus
icsStatus: active

-

add: icsExtendedDomainPrefs
icsExtendedDomainPrefs: domainAccess=@@d^a^slfrwd^g;anonymous^a^r^g;@^a^s^g
```

Setting Domain Specific Attributes and Preferences

Each domain has a set of attributes and preferences that you can set using the csdomain utility or the `commadmin` utility. These attributes are part of the `icsCalendarDomain` object class. The attributes include preferences such as access rights, access control lists (ACLs), domain searches, access rights for domain searches, user status, and proxy logins. For a complete list, see the tables under the description of the `csdomain` utility:

- "icsAllowRights Attribute: csdomain Utility" on page 200
- "icsExtendedDomainPrefs Attribute: csdomain Utility" on page 202
- "Other LDAP Directory Attributes: csdomain Utility" on page 205

Provisioning New Calendar Server Users

When a new user logs into Calendar Server for the first time, that user is automatically provisioned, if specific requirements are met as described in Calendar Server Logins. New users must have an LDAP user ID and password to log in.

To provision new Calendar Server users in a domain, use one of these utilities:

- Calendar Server csuser utility.
- Identity Server `commadmin` utility. For information about the `commadmin` utility, refer to the *Sun ONE Messaging and Collaboration 1.0 User Management Utility Installation and Reference Guide*.

Managing Domains Using Calendar Server Utilities

Use the following Calendar Server command-line utilities to manage domains in a hosted domain installation. Each utility allows you to include the `-d` *domain* option to operate on a specific target domain.

- csdomain manages Calendar Server LDAP attributes in the LDAP directory for a domain when you using LDAP Schema, v.1 or v.2. You can create a new domain as well as add, delete, and list the LDAP attributes in the LDAP directory for a domain. For more information, see "Creating New Domains" on page 141.
- csuser manages Calendar Server users in a domain.
- csresource manages Calendar Server resource calendars in a domain.
- cscal manages calendars and their properties in a domain.
- csattribute manages Calendar Server LDAP attributes in the LDAP server for a domain.

Hosted Domain Configuration Parameters

Table 8-1 describes the configuration parameters in the `ics.conf` file used for hosted domain support. If any of the following parameters are not in the `ics.conf` file, add the parameter and its associated value to the file and then restart Calendar Server for the values to take effect.

Table 8-1 Configuration Parameters for Hosted Domain Support

Parameter	Description
service.virtualdomain.support	Enables ("y") or disables ("n") support for hosted (virtual) domain mode. Default is "n".
local.schemaversion	Specifies the version of the LDAP schema: • "1" = Sun ONE LDAP Schema, v.1. See also service.dcroot. • "2" = Sun ONE LDAP Schema, v.2. See also service.schema2root. Default is "1".
service.dcroot	Specifies the root suffix of the DC tree in the LDAP directory, if local.schemaversion = "1". For example: "o=internet". In hosted (virtual) domain mode, Calendar Server uses the service.dcroot parameter and not the local.ugldapbasedn and local.authldapbasedn parameters. Conversely, in non-hosted (virtual) domain mode, Calendar Server uses the local.ugldapbasedn and local.authldapbasedn parameters and not the service.dcroot parameter.
service.schema2root	Specifies the root suffix underneath which all domains are found, if local.schemaversion = "2". For example: "o=sesta.com".
service.defaultdomain	Specifies the default domain for this instance of Calendar Server. Used when a domain name is not supplied during a login. For example: "sesta.com".
service.loginseparator	Specifies a string of separators used for the *login-separator* when Calendar Server parses "userid[*login-separator*]domain". Calendar Server tries each separator in turn. Default is "@+".
service.siteadmin.userid	Specifies the user ID of the domain administrator. For example: DomainAdmin@sesta.com.

Table 8-1 Configuration Parameters for Hosted Domain Support *(Continued)*

Parameter	Description
service.virtualdomain.scope = "select"	Controls cross domain searching: • "primary" = Search only within the domain where the use is logged in. • "select" = Search in any domain that allows the search. Default is "select".
local.domain.language	Specifies the language for the domain. Default is "en" (English).

Using WCAP Commands

If your site is configured for hosted domains, you must fully qualify each calendar ID (`calid`) and user IDs with the domain name in all WCAP commands. For example: `jsmith@sesta.com`.

Migration to a Hosted Domain Environment

To migrate a site to use hosted domains, use the `csvdmig` utility. This utility modifies the calendar database and the LDAP directory by assigning a domain name to each calendar ID (`calid`).

CAUTION Before you run `csvdmig`, first check with your Sun Microsystems technical support or sales account representative to ensure that you have the latest version of the utility.

Calendar Server 6.0 does not support multiple instances of Calendar Server on the same server.

If your site is currently configured for multiple instances of Calendar Server or for limited virtual domain mode, contact your Sun Microsystems sales account representative for an evaluation of your migration requirements.

The csvdmig migration utility makes these changes:

- Converts calendar IDs (calids) from the format userid[:calendar-name] to userid@domain[:calendar-name].
- Converts access control list (ACL) rules from the format userid to userid@domain.
- Converts directory server user entries for the icsCalendar, icsCalendarOwned, and icsSubscribed attributes from the format userid[:calendar-name] to userid@domain[:calendar-name].

For information about running csvdmig, refer to the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems*.

In addition to migration, you must also perform these tasks:

- Set service.virtualdomain.support to "yes" in the ics.conf file.
- Set up your directory server organization depending on the schema you are using. See "Organization of the LDAP Directory" on page 136.
- Add the icsCalendarDomain object class to the o=internet domain entry in the directory server. See "Using Domains Created by Messaging Server" on page 142.
- Set up the domain entries with your Calendar Server users in their respective domains. Then, set icsStatus to "active" and domainAccess to the ACL you want to use for access control.

For the most recent information, refer to the Release Notes on the following documentation Web site:

http://docs.sun.com/coll/S1_CalendarServer_60

Chapter 9

Using SSL With Calendar Server

Sun ONE Calendar Server 6.0 supports the Secure Sockets Layer (SSL) protocol to encrypt data between calendar client end users and Calendar Server. To support SSL, Calendar Server uses SSL libraries from Netscape Security Services (NSS), which are also used by Sun ONE Messaging Server.

You can configure Calendar Server in the `ics.conf` file to encrypt only the Calendar Server login and password or an entire calendar session.

This chapter describes these topics:

- Configuring SSL for Calendar Server
    - Create the SSL Certificate Database
    - Request and Import a Certificate From a Root Certificate Authority (CA)
    - Configure SSL Parameters in the ics.conf File
- Troubleshooting SSL

Configuring SSL for Calendar Server

Calendar Server SSL configuration is "self-contained" and does not require the Delegated Administrator.

To configure SSL for Calendar Server, follow these steps:

1. Create the SSL Certificate Database
2. Request and Import a Certificate From a Root Certificate Authority (CA)
3. Configure SSL Parameters in the ics.conf File

Create the SSL Certificate Database

An SSL implementation for Calendar Server requires a certificate database. The certificate database must define a Certificate Authority (CA) and certificates for Calendar Server.

Mozilla Tools

This release includes the following Mozilla tools:

- Certificate Database Tool (`certutil`) to create and manage the certificate database. For information, refer to the following Web site:

  http://mozilla.org/projects/security/pki/nss/tools/certutil.html

- Security Module Database Tool (`modutil`) to display information about available security modules. For information, refer to the following Web site:

  http://mozilla.org/projects/security/pki/nss/tools/modutil.html

These utilities are available in the following directory:

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

Library Path Variable

Before you use the Mozilla tools, set your LD_LIBRARY_PATH variable appropriately. For example:

```
setenv LD_LIBRARY_PATH /opt/SUNWics5/cal/lib
```

Example Files and Directories

The examples in this chapter use these files and directories:

- `alias` is a directory that contains the certificate database. Is is recommended that you create the `alias` directory in the following directory:

  `/var/opt/SUNWics5`

  Also, make sure you backup the `alias` directory regularly.

- `sslPasswordFile` is a text file that contains the certificate database password. This file is used by the `certutil` utility but not by Calendar Server. Is is recommended that you create `sslPasswordFile` in the following directory:

  `/etc/opt/SUNWics5/config`

- `/etc/passwd` introduces entropy for the random number generation.

To create a certificate database

1. Log in as or become superuser (`root`).

2. Specify the certificate database password for `certutil` in `/etc/opt/SUNWics5/config/sslPasswordFile`. For example:

   ```
   # echo "password" > /etc/opt/SUNWics5/config/sslPasswordFile
   ```

   where *password* is your specific password.

3. Create the certificate database `alias` directory. For example:

   ```
   # cd /var/opt/SUNWics5
   # mkdir alias
   ```

4. Move to the `bin` directory and generate the certificate database (`cert7.db`) and key database (`key3.db`). For example:

   ```
   # cd /opt/SUNWics5/cal/bin
   # ./certutil -N -d /var/opt/SUNWics5/alias
   -f /etc/opt/SUNWics5/config/sslPasswordFile
   ```

5. Generate a default self-signed root Certificate Authority certificate. For example:

```
# ./certutil -S -n SampleRootCA -x -t "CTu,CTu,CTu"
-s "CN=My Sample Root CA, O=sesta.com" -m 25000
-o /var/opt/SUNWics5/alias/SampleRootCA.crt
-d /var/opt/SUNWics5/alias
-f /etc/opt/SUNWics5/config/sslPasswordFile -z
/etc/passwd
```

6. Generate a certificate for the host. For example:

```
# ./certutil -S -n SampleSSLServerCert -c SampleRootCA -t "u,u,u"
-s "CN=hostname.sesta.com, O=sesta.com" -m 25001
-o /var/opt/SUNWics5/alias/SampleSSLServer.crt
-d /var/opt/SUNWics5/alias -f /etc/opt/SUNWics5/config/sslPasswordFile
-z /etc/passwd
```

where *hostname*`.sesta.com` is the server host name.

7. Validate the certificates. For example:

```
# ./certutil -V -u V -n SampleRootCA        -d /var/opt/SUNWics5/alias
# ./certutil -V -u V -n SampleSSLServerCert -d /var/opt/SUNWics5/alias
```

8. List the certificates. For example:

```
# ./certutil -L -d /var/opt/SUNWics5/alias
# ./certutil -L -n SampleSSLServerCert -d /var/opt/SUNWics5/alias
```

9. Use `modutil` to list the available security modules (`secmod.db`). For example:

```
# ./modutil -list -dbdir /var/opt/SUNWics5/alias
```

10. Change the owner of the `alias` file to `icsuser` and `icsgroup` (or the user and group identity under which Calendar Server will run). For example:

```
# find /var/opt/SUNWics5/alias -exec chown icsuser {} \;
# find /var/opt/SUNWics5/alias -exec chgrp icsgroup {} \;
```

Request and Import a Certificate From a Root Certificate Authority (CA)

The following steps generate a certificate request, submit it to the Public Key Infrastructure (PKI) Web site, and then import the certificate.

To request and import a certificate from a root Certificate Authority

1. Log in as or become superuser (`root`).

2. Move to the `bin` directory:

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

3. Use `certutil` to generate a Certificate Request based on the Certificate Authority or Public Key Infrastructure (PKI) Web site. For example:

   ```
   # ./certutil -R -s "CN=hostname.sesta.com, OU=hostname / SSL Web Server,
   O=Sesta C=US" -p "408-555-1234" -o hostnameCert.req -g 1024
   -d /var/opt/SUNWics5/alias
   -f /etc/opt/SUNWics5/config/sslPasswordFile
   -z /etc/passwd -a
   ```

   where *hostname*`.sesta.com` is the host name.

4. Request an test certificate for an SSL web server from the Certificate Authority or Public Key Infrastructure (PKI) Web site. Copy and paste the contents from the *hostname*`Cert.req` file into the Certificate Request.

   You will be notified by when your certificate is signed and can be picked up.

5. Copy the Certificate Authority Certificate Chain and SSL server certificate into text files.

6. Import the Certificate Authority Certificate Chain into the certificate database to establish a Chain of Authority. For example:

   ```
   # ./certutil -A -n "GTE CyberTrust Root" -t "TCu,TCu,TCuw"
   -d /var/opt/SUNWics5/alias -a -i
   /export/wspace/Certificates/CA_Certificate_1.txt
   -f /etc/opt/SUNWics5/config/sslPasswordFile
   ```

   ```
   # ./certutil -A -n "Sesta TEST Root CA" -t "TCu,TCu,TCuw"
   -d /var/opt/SUNWics5/alias -a -i
   /export/wspace/Certificates/CA_Certificate_2.txt
   -f /etc/opt/SUNWics5/config/sslPasswordFile
   ```

7. Import the signed SSL server certificate:

```
# ./certutil -A -n "hostname SSL Server Test Cert" -t "u,u,u"
-d /var/opt/SUNWics5/alias -a -i
/export/wspace/Certificates/SSL_Server_Certificate.txt
-f /etc/opt/SUNWics5/config/sslPasswordFile
```

8. List the certificates in the certificate database:

```
# ./certutil -L -d /var/opt/SUNWics5/alias
```

9. Configure the SSL Server Nickname in the `ics.conf` file to be the signed SSL server certificate, For example: "*hostname* SSL Server Test Cert".

   Note The host name for the `service.http.calendarhostname` and `service.http.ssl.sourceurl` parameters in the `ics.conf` file should match the host name on the SSL certificate (in case your system has several aliases). For example: calendar.sesta.com

Configure SSL Parameters in the ics.conf File

To implement SSL with Calendar Server, you must set specific parameters in the `ics.conf` file. This section describes:

- SSL Configuration Parameters
- Encryption of Calendar Server Login or Entire Calendar Session

SSL Configuration Parameters

Code Example 9-1 shows the SSL configuration parameters. If any of these parameters are not in the `ics.conf` file, add them to the file. After you set these parameters, restart Calendar Server for the values to take effect. For a description of these SSL parameters, see "SSL Configuration" on page 265.

Code Example 9-1 SSL Configuration Parameters for Calendar Server

```
service.http.ssl.cachedir = "."
service.http.ssl.cachesize = "10000"
service.http.ssl.certdb.password = "password"
service.http.ssl.certdb.path = "/var/opt/SUNWics5/alias"
service.http.ssl.port.enable = "yes"
service.http.ssl.port = "443"
service.http.ssl.securelogin = "yes"
service.http.ssl.securesession = "yes"

! Set localhost to the name of your local host.
! Set the port numner (default: 443) to the SSL port you are using.
service.http.ssl.sourceurl = "https://localhost:443"

service.http.ssl.ssl2.ciphers = ""
service.http.ssl.ssl2.sessiontimeout = "0"
service.http.ssl.ssl3.ciphers =
"rsa_rc4_40_md5,rsa_rc2_40_md5,rsa_des_sha,rsa_rc4_128_md5,rsa_3des_sha"
service.http.ssl.ssl3.sessiontimeout = "0"
service.http.ssl.usessl = "yes"
encryption.rsa.nssslactivation = "on"
encryption.rsa.nssslpersonalityssl = "SampleSSLServerCert"
encryption.rsa.nsssltoken = "internal"
service.http.tmpdir = "/var/opt/SUNWics5/tmp"
service.http.uidir.path = "html"
```

CAUTION Remote administration is not enabled for Calendar Server. Do not change the `service.admin.port` parameter, because it is already set to its required value by Calendar Server. Otherwise, the `csadmind` process might not run properly.

Encryption of Calendar Server Login or Entire Calendar Session

You can configure Calendar Server to encrypt only the Calendar Server login and password or an entire calendar session by setting the following parameters:

- `service.http.ssl.securelogin = "yes"` encrypts the login.
- `service.http.ssl.securesession = "yes"` encrypts the entire session.

Troubleshooting SSL

First, always backup your certificate database on a regular basis in case unrecoverable problems occur. If you have problems with SSL, here are some things to consider:

- Checking for the cshttpd Process
- Verifying Certificates
- Reviewing Calendar Server Log Files
- Connecting to the SSL Port

Checking for the cshttpd Process

SSL requires the Calendar Server cshttpd process to be running. To determine if cshttpd is running, use this command on Solaris systems:

```
# ps -ef | grep cshttpd
```

Verifying Certificates

To list the certificates in the certificate database and checking their validity dates, use this command on Solaris systems:

```
# ./certutil -L -d /var/opt/SUNWics5/alias
```

Reviewing Calendar Server Log Files

Check the Calendar Server log files for any SSL errors. For more information see “Monitoring Calendar Server Log Files” on page 95.

Connecting to the SSL Port

Connect to the SSL port using a browser and the following URL:

`https://`*server-name*`:`*ssl-port-number*

where:

server-name is the name of the server where Calendar Server is running.

ssl-port-number is the SSL port number as specified by the `service.http.ssl.port` parameter in the `ics.conf` file. The default is 443.

Chapter 10

Setting Up a High Availability (HA) Configuration

A Sun ONE Calendar Server high availability (HA) configuration provides monitoring and recovery from software and hardware failures. Calendar Server is available only as a failover service. This chapter describes a Calendar Server HA configuration using Sun Cluster software.

This chapter describes how to install and configure a Calendar Server HA service on Solaris Servers, including:

- Requirements for an HA Configuration
- Installation and Configuration
- Starting and Stopping Calendar Server HA Service
- Related Documentation

Appendix E, “High Availability (HA) Configuration Worksheets” on page 333 provides a set of worksheets to help you plan a Calendar Server HA configuration.

Requirements for an HA Configuration

A Calendar Server HA configuration requires the software shown in Table 10-1. (QFS is not supported, because an HA agent is not available.)

Table 10-1 Supported Software for a Calendar Server HA Configuration

Software and Version	Notes and Patches
Solaris 9 OS SPARC Platform only	All versions of Solaris 9 OS are supported. Solaris 9 OS requires Sun Cluster 3.0 U3 or later. Solaris 9 OS includes Solaris Logical Volume Manager (LVM).
Solaris 8 OS SPARC Platform only	Solaris 8 Maintenance Update 7 (MU7) OS or later plus required patches is recommended.

Sun Cluster 3.0 U3 or 3.1	Sun Cluster software must be installed and configured on all nodes in the cluster.
	To install Sun Cluster 3.1, use the Java Enterprise System installer by following the installation process in the *Sun Java Enterprise System Installation Guide*.
	After you install the Sun Cluster software, you must configure the cluster. For information, refer to the *Sun Cluster 3.1 Software Installation Guide*. For related documentation, see "Sun Cluster Documentation" on page 172.
	Sun Cluster Patches
	For the Solaris 9 OS, see the Sun Cluster InfoDoc 49704:
	`http://sunsolve.Central.Sun.COM/cgi/retrieve.pl?doc=finfodoc%2F49704`
	For the Solaris 8 OS, see the Sun Cluster InfoDoc 49705:
	`http://sunsolve.Central.Sun.COM/cgi/retrieve.pl?doc=intinfodoc%2F49705`

Solstice DiskSuite 4.x	Solstice DiskSuite is available for Solaris 8 OS only. Solstice DiskSuite is not required for Solaris 9 OS, which includes the Logical Volume Manager (LVM).
Veritas Volume Manager (VxVM) 3.x	Solaris 8 OS requires version 3.2 or later plus required patches. Solaris 9 OS requires version 3.5 or later plus required patches.
Veritas File System (VxFS) 3.x	Solaris 8 OS requires version 3.4 or later plus required patches. Solaris 9 OS requires version 3.5 or later plus required patches. HAStoragePlus requires patch 110435-08 or later.

Installation and Configuration

The Calendar Server HA configuration examples in this section use the following names:

Name in Example	Description
/global/cal	Global file-system mount point
cal-logical-host	Logical host name
cal-logical-host-ip	Logical host IP numeric address
cs-admin@cal-logical-host	Email address for the Calendar Server administrator
cal-node-1	Node 1
cal-node-2	Node 2
cal-resource-group	Calendar resource group
cal-resource-group-store	Calendar Server storage resource
cal-resource	Calendar Server resource

To install and configure a Calendar Server HA configuration

1. Log in as root
2. Prepare Each Node in the Cluster
3. Install Sun Products and Packages
4. Configure the Logical Host
5. Activate the Storage Resource
6. Configure Calendar Server
7. Relocate the Calendar Server config Directory
8. Edit the Calendar Server ics.conf File
9. Start the HA Calendar Server
10. Verify the HA Configuration

Log in as root

To install and configure a Calendar Server HA configuration, log in as or become superuser (`root`) and specify a console or window for viewing messages sent to `/dev/console`.

Prepare Each Node in the Cluster

On each node in the cluster, perform these steps:

1. Create the Calendar Server runtime user and group under which Calendar Server will run as follows:

   a. Add `icsgroup` (or the value you selected) to the `/etc/group` file.

   b. Add `icsuser` (or the value you selected) to the `/etc/passwd` file.

   Note The default names are `icsuser` and `icsgroup`. You can use other names if you prefer, but the UID and GID numbers **must** be the same on all nodes in the cluster. The user name should **not** be `root`.

   You must provide the user and group names when you Configure Calendar Server on Node 1.

2. Add or set the following fields in the `/etc/vfstab` file:

   ❍ `mount point` to /*global*/*cal* (or the file-system mount point you selected in Selecting the Calendar Server Installation Directory)

   ❍ `mount at boot` option to `no`

   ❍ `mount options` to `logging` for FFS or `global,logging` for GFS

Install Sun Products and Packages

The installation of Sun products, including Sun ONE Calendar Server 6.0, on Solaris Systems has significant changes from previous releases. To install Calendar Server 6.0 and other products such as Sun Cluster 3.1 and Sun ONE Directory Server 5.2, you must use the Sun Java Enterprise System installer.

For information about the installer, refer to the *Sun Java Enterprise System Installation Guide*.

Table 10-2 describes the Sun products and packages required for a Calendar Server HA configuration.

Table 10-2 Sun Products and Packages Required for a Calendar Server HA Configuration

Product or Package	Node 1	Node 2
Sun Cluster Software	yes	yes
Sun ONE Calendar Server 6.0	yes	no
Sun Cluster Agent for Calendar Server (SUNWscics package)	yes	yes
Shared components (SUNWicu, SUNWldk, SUNWpr, SUNWsasl, and SUNWtls packages)	yes	yes

Node 1

On node 1, install all selected products and packages using the Java Enterprise System installer. When you install Calendar Server, you must specify a directory other the default directory. See Selecting the Calendar Server Installation Directory.

Node 2

On node 2, follow these steps:

1. Install Sun Cluster 3.1 and the Sun Cluster Agent for Calendar Server (SUNWscics package) using the Java Enterprise System installer. (

   Note You cannot install only the Sun Cluster Agent for Calendar Server. When you chose the Sun ONE Agents for Sun Cluster, the Java Enterprise System installer installs all agents.

2. Install shared components (SUNWicu, SUNWldk, SUNWpr, SUNWsasl, and SUNWtls packages) using the `pkgadd` command. See Installing Shared Components.

Selecting the Calendar Server Installation Directory

For Calendar Server 6.0, the Java Enterprise System installer uses the following default installation directory: /opt

However, for an HA configuration, you must specify a global installation directory. For example: /*global*/*cal*/opt/

Installing Shared Components

To make the required shared components available on node 2, you must install the following packages:

- SUNWicu–International Components for Unicode User Files
- SUNWldk–LDAP C SDK
- SUNWpr–Netscape Portable Runtime Interface
- SUNWsasl–Simple Authentication and Security Layer (SASL)
- SUNWtls–Network Security Services

These packages are available in the following directories:

```
.../Solaris_sparc/Product/shared_components/Packages/SUNWldk
.../Solaris_sparc/Product/shared_components/Solaris_8/Packages
.../Solaris_sparc/Product/shared_components/Solaris_9/Packages
```

To install these packages, change to one of the directories shown above and use the `pkgadd` command. For example:

```
# pkgadd -d . SUNWicu SUNWpr SUNWsasl SUNWtls
```

Configure the Logical Host

To configure the logical host:

1. Create a Calendar Server failover resource group named *cal-resource-group*:

   ```
   # scrgadm -a -g cal-resource-group -h cal-node-2,cal-node-1
   ```

2. Add the logical host name named *cal-logical-host* to the resource group. Calendar Server will listen on this host name.

   ```
   # scrgadm -a -L -g cal-resource-group -l  cal-logical-host
   ```

3. Bring the resource group online:

   ```
   # scswitch -Z -g cal-resource-group
   ```

Activate the Storage Resource

To activate the storage resource:

1. Register the storage resource specifying the mount point as the ServicePaths property:

   ```
   # scrgadm -a -j cal-resource-group-store -g cal-resource-group -t SUNW.HAStorage \
             -x ServicePaths=/global/cal \
             -x AffinityOn=True
   ```

2. Enable the storage resource:

   ```
   # scswitch -e -j cal-resource-group-store
   ```

If `SUNW.HAStoragePlus` has also chosen to setup a global file system (GFS), the `FileSystemMountPoints` property must be set instead of `ServicePaths`.

Configure Calendar Server

After you install Calendar Server, run the Directory Server Setup script (`comm_dssetup.pl`) and the Calendar Server configuration program (`csconfigurator.sh`) as described in the *Sun ONE Calendar Server 6.0 Installation Guide for Solaris Operating Systems.*

Table 10-3 describes the specific configuration information that you must provide for an HA configuration.

Table 10-3 Calendar Server Configuration Options for an HA Configuration

Configuration Panel	Description
Runtime Configuration	**Runtime User ID and Group ID** • Runtime user ID is the user name under which Calendar Server will run. This name should **not** be root. The default is icsuser. • Runtime Group ID is the group under which Calendar Server will run. The default is icsgroup. Although the configuration program can create these names for you, you should create them before you run the configuration program. See Prepare Each Node in the Cluster. These names must be in the following files: • icsuser (or the name you select) in /etc/passwd on all nodes in the cluster • icsgroup (or the name you select) in /etc/group on all nodes in the cluster **Calendar Server Startup** Do **not** check either of these options. • Start after successful installation • Start on system startup
Select Directories	For the location of the database, temporary, and log files, select global partitions. For example: • Database: /*global*/*cal*/var/csdb • Temporary Files: /*global*/*cal*/var/tmp • Logs: /*global*/*cal*/var/logs

Relocate the Calendar Server config Directory

Calendar Server stores configuration files in the following directory:

```
/etc/opt/SUNWics5/config/
```

For an HA configuration, you must perform the following steps after you run the Calendar Server configuration program (`csconfigurator.sh`):

1. Change to the /*global*/*cal*/`opt/SUNWics5/cal` directory, For example:

   ```
   # cd /global/cal/opt/SUNWics5/cal/
   ```

   where /*global*/*cal* is the file-system mount point.

2. Check that `config` is a symbolic link to the new `config` directory. For example:

   ```
   # ls -l config
   ... config -> /etc/opt/SUNWics5/config/
   ```

3. In the `/opt/SUNWics5/cal/` directory, remove the `config` symbolic link:

   ```
   # rm config
   ```

4. Copy the contents of the `/etc/opt/SUNWics5/config` directory, preserving the ownership and permissions:

   ```
   # cp -pr /etc/opt/SUNWics5/config .
   ```

5. Make sure that the contents of `config` are owned by `icsuser` and `icsgroup` (or your choices you specified for the runtime User ID and Group ID):

   ```
   # ls -ld config
   ... icsuser icsgroup ... config/
   ```

6. In the /*global*/*cal*/`opt/SUNWics5/cal/lib` directory, check that `config` is a symbolic link to `/etc/opt/SUNWics5/config`.

   ```
   # cd /global/cal/opt/SUNWics5/cal/lib
   # ls -l config
   ... config -> /etc/opt/SUNWics5/config/
   ```

7. Remove the `config` symbolic link:

   ```
   # rm config
   ```

8. Create a new symbolic link to the new `config` location:

   ```
   # ln -s ../config config
   ```

9. Verify the new link:

```
# ls -l config
... config -> ../config/
```

10. In the /*global*/*cal*/opt/SUNWics5/cal/sbin directory, check that config is a symbolic link to /etc/opt/SUNWics5/config.

```
# cd /global/cal/opt/SUNWics5/cal/sbin
# ls -l config
... config -> /etc/opt/SUNWics5/config/
```

11. Remove the config symbolic link:

```
# rm config
```

12. Create a new symbolic link to the new config location:

```
# ln -s ../config config
```

13. Verify the new link:

```
# ls -l config
... config -> ../config/
```

NOTE If you need to uninstall Calendar Server, use the Java Enterprise System uninstaller, which removes the SUNWics5 and SUNWica5 packages.

For a Calendar Server HA configuration, however, you must first remove the relocated config directory and all of its contents before you run the uninstaller. For example:

```
# cd /global/cal/opt/SUNWics5/cal/
# rm -rf config
```

If you do not remove the config directory, the uninstall operation fails for the SUNWics5 package.

Edit the Calendar Server ics.conf File

In the `/opt/SUNWics5/cal/config` directory, edit the `ics.conf` configuration file as follows:

1. Add the following parameters:

```
local.server.ha.enabled = "yes"
local.server.ha.agent = "SUNWscics"
```

2. Rename the `service.listenaddr` parameter to `service.http.listenaddr` and then set the parameter to the IP address of the logical host. For example:

```
service.http.listenaddr = "cal-logical-host-ip"
```

   where "*cal-logical-host-ip*" is the numeric IP address of the logical host. For example: `123.321.12.2`.

3. Change all parameters that refer to a local host name to the logical host name. For example:

```
local.hostname = "cal-logical-host"
local.servername = "cal-logical-host"
service.ens.host = "cal-logical-host"
service.http.calendarhostname = "cal-logical-host.sesta.com"
```

Start the HA Calendar Server

Before you start the HA Calendar Server, register the calendar resource type SUNWscics and create a calendar resource as follows:

1. Register the calendar resource type:

```
# scrgadm -a -t SUNW.scics
```

2. Create the calendar resource:

```
# scrgadm -a -j cal-resource -g cal-resource-group -t SUNW.scics \
       -x Confdir_list=/global/cal/cal-resource-group \
       -y Resource_dependencies=cal-resource-group-store \
       -y Port_list=80/tcp
```

3. Enable the resource and start Calendar Server:

```
# scswitch -e -j cal-resource
```

Verify the HA Configuration

After you start Calendar Server check that all required processes or daemons (`csadmind`, `enpd`, `csnotifyd`, and `cshttpd`) are running.

Additionally, conduct a switchover of the service to the backup node to ensure the high availability. For example, if the service is running on *cal-node-1*, issue the following command to switch the service to *cal-node-2*.

```
# scswitch -z -g cal-resource-group -h cal-node-2
```

Then check that all processes are started on *cal-node-2*.

For troubleshooting, error messages are written to the console and `/var/adm/messages`.

The `/var/cluster/rgm/rt/SUNW.scics/loglevel` file contains the logging level. Use '9' for maximum verbosity.

For information about using the logging facility, refer to the "Sun Cluster Documentation" on page 172.

Starting and Stopping Calendar Server HA Service

To start and stop the Calendar Server HA service, use the Sun Cluster `scswitch` command. Do not use the Calendar Server `start-cal`, `csstart`, `stop-cal`, or `csstop` utilities. For example:

To start the Calendar Server HA service:

```
# scswitch -e -j cal-resource
```

To stop the Calendar Server HA service:

```
# scswitch -n -j cal-resource
```

To restart the Calendar Server HA service:

```
# scswitch -R -j cal-resource
```

For information about the Sun Cluster `scswitch` command, refer to the *Sun Cluster 3.1 Reference Manual*.

Related Documentation

Sun Java Enterprise System Documentation

- *Sun Java Enterprise System Installation Guide* describes the Java Enterprise System installer (and uninstaller) and the supported installation scenarios.
- *Release Notes* provide current information about the Sun Java Enterprise System product.

Sun Cluster Documentation

- *Sun Cluster 3.1 Concepts Guide* provides a general background about Sun Cluster software, data services, and terminology resource types, resources, and resource groups.
- *Sun Cluster 3.1 Data Services Planning and Administration Guide* provides general information on planning and administration of data services.
- *Sun Cluster 3.1 Software Installation Guide* provides guidelines for planning a Sun Cluster 3.1 configuration, and provides procedures to install, upgrade, and configure the Sun Cluster software.
- *Sun Cluster 3.1 System Administration Guide* provides the software procedures for administering a Sun Cluster configuration.
- *Sun Cluster 3.1 Reference Manual* describes the commands and utilities available with the Sun Cluster software, including commands found only in the SUNWscman and SUNWccon packages.

Solstice DiskSuite Documentation (Solaris 8 OS Only)

Solstice DiskSuite 4.2.1 Installation and Product Notes provides information about installing, upgrading and managing Solstice DiskSuite 4.2.1.

Chapter 11

Calendar Server Command-Line Utilities

Sun ONE 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 following directory:

cal_svr_base`/opt/SUNWics5/cal/sbin`

This chapter provides the following information:

Running the Command-Line Utilities

On Solaris Systems, run the command-line utilities while logged in as the user and group under which Calendar Server is running that was specified during installation (defaults are `icsuser` and `icsgroup`) or as `root`. In most cases, you must change to the directory where the utilities are located.For example, on Solaris Systems: *cal_svr_base*`/opt/SUNWics5/cal/sbin`.

Syntax for Command-Line Utilities

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 (`[]`) 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.

`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 specify only the utility name, it lists all commands, options, and several examples.
- If you do not specify a required password, the utility prompts you for it.
- The `-v` (verbose) and `-q` (quiet) options are available for each utility.
- If a command is dangerous (that is, one that could cause a 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.

Return Code in Scripts

If you run the command-line utilities from a script, the return code is “0” if the utility run successfully or “-1” for a failure.

Table of Command-Line Utilities

Table 11-1 describes the Calendar Server command-line utilities.

Table 11-1 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.
csclean	Removes user and resource calendars for Calendar Server users whose status attribute (inetUserStatus) has been marked as “deleted” by the Sun ONE Identity Server commadmin utility.
cscomponents	Manages calendar components: events and tasks (todos).
csdb	Manages the calendar database.
csdomain	Manages Calendar Server attributes in the LDAP directory for a hosted (virtual) domain.
csexport	Exports a calendar in iCalendar (.ics) or XML (.xml) format.
csimport	Imports a calendar in iCalendar (.ics) or XML (.xml) format.
csmonitor	Monitors LDAP connectivity, log files, and available disk space for the calendar database.
csplugin	Views, enables, or disables configured Calendar Server API (CSAPI) plug-ins.
cspurge	Allows the manual purge of entries in the Delete Log database (ics50deletelog.db).
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 processes.
csstats	Displays counters in a Calendar Server.
csstop	Stops the Calendar Server processes.
csstored.pl	Performs archival operations for the calendar database and log files
cstool	Pings a running Calendar Server instance.
csuser	Manages calendar users.
start-cal	Starts all Calendar Server processes.
stop-cal	Stops all Calendar Server processes.

csattribute

The `csattribute` utility manages 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.

NOTE If your site is using the LDAP CLD plug-in, do not use `csattribute` to change the `icsDWPHost` attribute to specify a new back-end host server. Modifying `icsDWPHost` does not cause a new calendar to be created on the new back-end host. For more information, see "Configuring the LDAP Calendar Lookup Database (CLD) Plug-in" on page 78.

Requirements

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

Syntax

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

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

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

Table 11-2 describes the commands available for `csattribute`.

Table 11-2 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 11-3 describes the `csattribute` utility command options.

Table 11-3 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`	Type of target (user or resource object). Default is user.
[`-d` *domain*]	Specifies the name of a hosted (virtual) domain. Default is taken from the service.defaultdomain parameter in the ics.conf file.

Examples

- Add the `icsCalendar` LDAP attribute 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 222.

Requirements

- Calendar Server can be running or stopped.
- You must run the utility locally on the machine where Calendar Server is installed.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

Syntax

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


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


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

Table 11-4 describes the commands available for `csbackup`.

Table 11-4 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: *cal_svr_base*`/opt/SUNWics5/cal/sbin/`*target-directory* If you specify only the target database directory, do not include the slash (/) before the directory name. For example: `csbackup database backupdir` **Note**: The 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 csbackup. You can also specify a non-existent target backup directory and let csbackup create the directory for you.
`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 11-5 describes the `csbackup` utility command options.

Table 11-5 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.

Table 11-5 csbackup Utility Command Options *(Continued)*

Option	Description
`-a` *userid*	The user ID of the calendar user to backup. This option is required for the default option. There is no default.
`-b` *basedn*	The base DN to be used for this user. The default is taken from the setting local.ugldapbasedn defined in the ics.conf file. The Base DN (distinguished name) is the entry in your LDAP directory used as the starting point from which searches occur. For example, if you specify a base DN of ou=people, o=sesta.com, all LDAP search operations executed by 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 51.
`-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 Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

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] [-O] delete|reset calid


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


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

Table 11-6 describes the commands available for the `cscal` utility.

Table 11-6 cscal Utility Commands

Command	Description
`create` *calid*	Creates the calendar specified by calid. Note: If your site is using the LDAP CLD plug-in, you must create a new calendar on the same back-end server where the user's calendars reside (or will reside), as indicated by the user's icsDWPHost LDAP attribute. If you try to create a calendar on a different back-end server, Calendar Server returns an error.
`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 by *calid*. If *calid* is not specified, enables all calendars. If the -o *owner* option is specified, enables all calendars whose primary owner is the specified owner.
`disable [`*calid*`]`	Disables the calendar specified by *calid*. If *calid* is not specified, disables all calendars. If the -o *owner* option is specified, disables all calendars whose primary owner is the specified owner.

Table 11-6 cscal Utility Commands *(Continued)*

Command	Description
`list [`*calid*`]`	Lists properties of the calendar specified by *calid*. If *calid* is not specified, lists properties of all calendars. If the -o *owner* option is specified, lists all calendars whose primary owner is the specified owner. ACE strings are case insensitive. If the ACE has been modified through the Calendar Express GUI, the ACE characters that were modified show up in upper case.
`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 11-7 describes the `cscal` utility command options.

Table 11-7 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.
`-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.

Table 11-7 cscal Utility Command Options *(Continued)*

Option	Description
`-g` *category*	Category. Multiple categories must be enclosed in quotation marks ("") and separated by spaces. The default is no category.
`-k yes|no`	Specifies whether double booking is allowed for a user calendar. For example, yes means the calendar can have more than one event scheduled for the same time slot. If the -k option is omitted, the default is taken from the user.allow.doublebook parameter in the ics.conf file. However, the user.allow.doublebook parameter is used only when a calendar is created. After a calendar is created, Calendar Server checks the calendar properties (ics50calprops.db) to determine if double booking is allowed. If you need to change the calendar properties for a calendar to allow or disallow double booking, reissue cscal with the -k option.
`-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.
-O	Specifies all calendars of the primary owner. Default is the named calendar only.
`-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
  ```

csclean

The `csclean` utility removes user and resource calendars for Calendar Server users whose status attribute (`inetUserStatus`) has been marked as “deleted” by the Sun ONE Identity Server `commadmin` utility.

Deletion of a Calendar Server user involves these steps:

1. Run the `commadmin` utility `user delete` command to mark the either the user or the user’s calendar service as “deleted”.
2. Run the `csclean` utility to remove the user’s corresponding calendars from the calendar database and to mark the user’s the `inetUserStatus` attribute as “removed”.
3. Run the `commadmin` utility `domain purge` command to remove the user’s LDAP entry.

For information about the `commadmin` utility, refer to the *Sun ONE Messaging and Collaboration 1.0 User Management Utility Installation and Reference Guide*.

Sun ONE Messaging Server uses a similar utility, `msuerpurge`, to remove users from the message store. For information about the `msuserpurge` utility, refer to the *Sun ONE Messaging Server 6.0 Reference Manual*.

Requirements

- Calendar Server can be running or stopped.
- You must run `csclean` locally on the machine where Calendar Server is installed.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

```
csclean [-q | -v] [-g graceperiod] clean domain
```

Table 11-8 describes the csclean utility command options.

Table 11-8 csclean Utility Command Options

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.
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-g` *graceperiod*	Specifies the number of days to have elapsed since the calendar service was deleted for a user. The default is 10 days.
domain	Specifies the domain in which to remove calendars for all users and resources. An asterisk (*) removes all calendars for all users and resources in all domains.

Examples

- Remove calendars for all users and resources in `sesta.com` whose calendar service has been deleted for at least 5 days:

  ```
  csclean -g 5 clean sesta.com
  ```

- Remove calendars for all users and resources in all domains whose calendar service has been deleted for at least 10 days:

  ```
  csclean -g 10 clean "*"
  ```

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 Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

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

Table 11-9 describes the commands available for the `cscomponents` utility.

Table 11-9 cscomponents Utility Commands

Command	Description
`delete` *calid*	Deletes events and tasks in the calendar with the specified calendar ID. **Special Note**: For deleting tasks, you must specify the -s starting time. It must be a non-zero TimeDate Z string. You can lose data if you do not specify the starting date. If the date is not specified, all tasks are removed from the calendar.
`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 11-10 describes the `cscomponent` utility command options.

Table 11-10 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. For events, a start time of `0` means from the beginning of time. The default is `0`. Do not use the default (0)for tasks. Leaving this parameter off, or specifying zero (0) when deleting tasks, causes all tasks in the calendar to be deleted. You must have a valid DateTime Z string for the value of this option when deleting tasks.
`-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, Calendar Server creates one automatically.)
- delete an existing calendar database. A database cannot be deleted while it is open (when 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 Calendar Server is installed.
- Calendar Server must be stopped for the create, delete, or rebuild commands.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

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 [dbdir]

csdb rebuild [-g] [dbdir [dstdir]]
```

Table 11-11 describes the commands available for the csdb utility.

Table 11-11 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, 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 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 [-g] [`*dbdir* `[`*dstdir*`]]`	Scans a calendar database in the specified database directory to determine if any corruption has occurred and generates a rebuilt calendar database (.db files). If a database directory is not specified, the current directory is used. The -g option rebuilds the group scheduling engine (GSE) database in addition to the other calendar databases, and *dstdir* specifies an optional destination directory.

Table 11-11 csdb Utility Commands *(Continued)*

Command	Description
`version`	Displays the version of the utility.

Table 11-12 describes the `csdb` utility command options.

Table 11-12 csdb 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`	Force the recovery of the calendar database.
`-g`	For the rebuild command, rebuild the group scheduling engine (GSE) database in addition to the other calendar databases.
`-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
  ```

csdomain

The csdomain utility manages Calendar Server attributes in the LDAP directory for a hosted (virtual) domain. These attributes are part of the icsCalendarDomain object class. Commands are:

- create a new hosted domain in the LDAP directory.
- add a Calendar Server attribute and its associated value in the LDAP directory for a specific hosted domain.
- delete a Calendar Server attribute in the LDAP directory for a specific hosted domain or delete an entire hosted domain.
- list Calendar Server attributes in the LDAP directory for a specific hosted domain.

Requirements

- You must be in hosted (virtual) domain mode to run csdomain. That is, the following parameters in the ics.conf file must be set:
    - service.virtualdomain.support must be set to “yes”.
    - local.schemaversion must be set to the version of the LDAP schema (1 or 2).
    - If local.schemaversion = “1”, service.dcroot must be set to the root suffix of the DC tree in the LDAP directory.
    - If local.schemaversion = “2”, service.schema2root must be set to the root suffix underneath which all domains are found.
- You must run csdomain locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

Syntax

```
csdomain [-q | -v] -n node create domain

csdomain [-q | -v] {-a attr[=value] | -f filename} add domain

csdomain [-q | -v] [-a attr | -f filename] delete domain

csdomain [-q | -v] list domain
```

Table 11-13 describes the commands available for the `csdomain` utility.

Table 11-13 `csdomain` Utility Commands

Command	Description
`create`	Create a new hosted domain in the LDAP directory. All Calendar Server users and resources for the domain are then created under this entry in the directory.
`add`	Add a Calendar Server attribute and its associated value in the LDAP directory for a specific domain. If you add or update domain LDAP attributes using csdomain, restart Calendar Server for the new values to take effect.
`delete`	Delete a Calendar Server attribute in the LDAP directory for a specific hosted domain or delete all LDAP entries for an entire domain.
`list`	Display Calendar Server attributes in the LDAP directory for a specific domain.
`version`	Display the version of the utility.

Table 11-14 describes the `csdomain` utility command options.

Table 11-14 csdomain 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` *attr*`[=`*value*`]`	Specifies the LDAP attribute property name and its optional value. For a list of these attributes and property names, see LDAP Attributes and Property Names.

`-f` *filename*	Specifies a text file that contains Calendar Server LDAP directory property names and their associated values. For example:
	createLowerCase="yes"
	filterPrivateEvents="no"
	fbIncludeDefCal="no"
	subIncludeDefCal="no"
	uiProxyUrl="https://*proxyserver*"

`-n` *node*	Applies to the create command as follows: • For LDAP Schema v.1–Specifies the node under which all users and resources are created. For example: o=node2,o=node1,o=sesta • For LDAP Schema v.2–Specifies the name of the node created for this domain. For example: o=west.sesta.com If node is not specified, the domain name is used.
domain	For the add, delete, and list commands, specifies an existing domain in the LDAP directory. For the create command, specifies the unique name of a new domain that will be created in the LDAP directory. For example: west.sesta.com

LDAP Attributes and Property Names

The following tables describe the LDAP attributes and property names that apply to the csdomain utility. These attributes are part of the icsCalendarDomain object class. When you add or delete a value, you must use the property name and not the attribute name.

- icsAllowRights Attribute: csdomain Utility
- icsExtendedDomainPrefs Attribute: csdomain Utility
- Other LDAP Directory Attributes: csdomain Utility

If you add or update domain LDAP attributes using csdomain, restart Calendar Server for the new values to take effect.

icsAllowRights Attribute: csdomain Utility

Table 11-15 describes the icsAllowRights attribute and properties that you can set with the csdomain utility. This attribute is a 32-bit numeric string, with each bit in the string corresponding to a specific user right. (In the current release, some bits are not used and are set to zero by default.) If a bit corresponding to a specific right is set (value=1), the right is not allowed. If the bit is not set (value=0), the right is allowed.

Each property in the icsAllowRights attribute has a corresponding ics.conf parameter. If a property is not set (value = 0) or is not present (service.virtualdomain.support = “no”), Calendar Server uses the corresponding ics.conf parameter as the default value.

The value for icsAllowRights is a numeric string and not an integer. To use icsAllowRights programatically in bitwise operations, you must first convert its string value to an integer.

Table 11-15 icsAllowRights LDAP Directory Attribute and Properties

Bit	Property Name	Description
0	allowCalendarCreation	If set (bit 0=1), do not allow calendars to be created. Corresponding ics.conf parameter: service.wcap.allowcreatecalendars
1	allowCalendarDeletion	If set (bit 1=1), do not allow calendars to be deleted. Corresponding ics.conf parameter: service.wcap.allowdeletecalendars

Table 11-15 icsAllowRights LDAP Directory Attribute and Properties *(Continued)*

Bit	Property Name	Description
2	allowPublicWritableCalendars	If set (bit 2=1), do not allow public writeable calendars. Corresponding ics.conf parameter: service.wcap.allowpublicwriteablecalendars
3	n/a	Not used in the current release.
4	allowModifyUserPreferences	If set (bit 4=1), do not allow domain administrators to get or set user preferences using WCAP commands. Corresponding ics.conf parameter: service.admin.calmaster.wcap.allowgetmodifyuserprefs
5	allowModifyPassword	If set (bit 5=1), do not allow user to change password via this server. Corresponding ics.conf parameter: service.wcap.allowchangepassword
6,7	n/a	Not used in the current release.
8	allowUserDoubleBook	If set (bit 8=1), do not allow double booking for user’s calendars. Corresponding ics.conf parameter: user.allow.doublebook
9	allowResourceDoubleBook	If set (bit 9=1), do not allow double booking for resource calendars. Corresponding ics.conf parameter: resource.allow.doublebook
10	allowSetCn	If set (bit 10=1), do not allow user to set the common name (cn) attribute using the WCAP set_userprefs command. Corresponding ics.conf parameter: service.wcap.allowsetprefs.cn
11	allowSetGivenName	If set (bit 11=1), do not allow user to set the givenName attribute using the WCAP set_userprefs command. Corresponding ics.conf parameter: service.wcap.allowsetprefs.givenname
12	allowSetGivenMail	If set (bit 12=1), do not allow user to set the mail attribute using the WCAP set_userprefs command. Corresponding ics.conf parameter: service.wcap.allowsetprefs.mail

Table 11-15 icsAllowRights LDAP Directory Attribute and Properties *(Continued)*

Bit	Property Name	Description
13	allowSetPrefLang	If set (bit 13=1), do not allow user to set the preferredLanguage attribute using the WCAP set_userprefs command. Corresponding ics.conf parameter: service.wcap.allowsetprefs.preferredlanguage
14	allowSetSn	If set (bit 14=1), do not allow user to set the surname (sn) attribute using the WCAP set_userprefs command. Corresponding ics.conf parameter: service.wcap.allowsetprefs.sn
15		If set (bit 15=1), do not allow user to set the user's default calendar ID (nswccalid) using the WCAP set_userprefs command. Corresponding ics.conf parameter: service.wcap.allowsetprefs.nswccalid
16-31	n/a	Not used in the current release.

icsExtendedDomainPrefs Attribute: csdomain Utility

Table 11-16 describes the icsExtendedDomainPrefs attribute and properties that you can set with the csdomain utility. Each property has a corresponding `ics.conf` parameter. If a property is not set (value = 0) or is not present (`service.virtualdomain.support = “no”`), Calendar Server uses the corresponding `ics.conf` parameter as the default value.

Table 11-16 icsExtendedDomainPrefs LDAP Directory Attribute

Property Name	Description
allowProxyLogin	Specifies "yes" or "no" whether to allow proxy logins. Corresponding ics.conf parameter: service.http.allowadminproxy (default = "no")
calmasterAccessOverride	Specifies "yes" or "no" whether the Calendar Server administrator can override access control. Corresponding ics.conf parameter: service.admin.calmaster.overrides.accesscontrol (default = "no")

Table 11-16 icsExtendedDomainPrefs LDAP Directory Attribute *(Continued)*

Property Name	Description
calmasterCred	Specifies an ASCII string that is the password of the user ID specified as the Calendar Server domain administrator. Corresponding ics.conf parameter: service.admin.calmaster.cred (no default)
calmasterUid	Specifies an ASCII string that is the user ID of the person designated as the Calendar Server domain administrator. Corresponding ics.conf parameter: service.admin.calmaster.userid (no default)
createLowercase	Specifies "yes" or "no" whether Calendar Server should convert a calendar ID (calid) to lowercase when creating a new calendar or when searching for a calendar Corresponding ics.conf parameter: calstore.calendar.create.lowercase (default = "no")
domainAccess	Specifies an access control list (ACL) for the domain. For information about ACLs, see "Access Control Lists (ACLs)" on page 101. This ACL is used for cross domain searches. For more information, see "Cross Domain Searches" on page 139.
fbIncludeDefCal	Specifies "yes" or "no" whether a user's default calendar is included in user's free/busy calendar list. Corresponding ics.conf parameter: calstore.freebusy.include.defaultcalendar (default = "yes")
filterPrivateEvents	Specifies "yes" or "no" whether Calendar Server filters (recognizes) Private and Time and Date Only (confidential) events and tasks. If "no", Calendar Server treats them the same as Public events and tasks. Corresponding ics.conf parameter: calstore.filterprivateevents (default = "yes")
groupMaxSize	Specifies the maximum number of attendees allowed in an LDAP group when expanding an event. Corresponding ics.conf parameter: calstore.group.attendee.maxsize (default is "0" – expand the group entirely)
language	Specifies the language for a domain. Corresponding ics.conf parameter: local.domain.language

Table 11-16 icsExtendedDomainPrefs LDAP Directory Attribute *(Continued)*

Property Name	Description
resourceDefaultAcl	Specifies an access control list (ACL) that is the default access control permissions used when a resource calendar is created. Corresponding ics.conf parameter: resource.default.acl (default is "@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g")
setPublicRead	Specifies whether user default calendars are initially set to public read/private write ("yes") or private read/private write ("no"). Corresponding ics.conf parameter: service.wcap.login.calendar.publicread (default = "no")
searchFilter	Specifies a search filter for finding a user. Corresponding ics.conf parameter: local.userSearchFilter
ssoCookieDomain	Specifies that the browser should send a cookie only to servers in the specified domain. The value must begin with a period (.). For example: ".sesta.com" Corresponding ics.conf parameter: sso.cookiedomain (default is the current domain)
ssoUserDomain	Specifies the domain used as part of the user's SSO authentication. Corresponding ics.conf parameter: sso.userdomain (no default)
subIncludeDefCal	Specifies "yes" or "no" whether a user's default calendar is included in the user's subscribed calendar list. Corresponding ics.conf parameter: calstore.subscribed.include.defaultcalendar (default = "yes")
uiAllowAnyone	Specifies "yes" or "no" whether Calendar Express should show and use the "Everybody" access control list (ACL). Corresponding ics.conf parameter: ui.allow.anyone (default = "yes")
uiAllowDomain	Specifies "yes" or "no" whether Calendar Express should show and use the access control list (ACL) for this domain. Corresponding ics.conf parameter: ui.allow.domain (default = "no")
uiBaseUrl	Specifies a URL for the base server address. For example: "https://proxyserver". Corresponding ics.conf parameter: ui.base.url (no default)

Table 11-16 icsExtendedDomainPrefs LDAP Directory Attribute *(Continued)*

Property Name	Description
uiConfigFile	Specifies an optional `xml` based configuration file that Calendar Server can read at startup that allows parts of the user interface to be hidden. Corresponding ics.conf parameter: ui.config.file (no default)
uiProxyURL	Specifies a URL for the proxy server address to prepend in an HTML UI JavaScript file. For example: "https://web_portal.sesta.com/" Corresponding ics.conf parameter: ui.proxyaddress.url (no default)

Other LDAP Directory Attributes: csdomain Utility

Table 11-17 describes other LDAP attributes and properties that you can set with the csdomain utility.

Table 11-17 Other LDAP Directory Attributes for the csdomain Utility

LDAP Attribute	Property Name	Description
icsAllowedServiceAccess	allowedAccessProtocols	Specifies whether access to Calendar Server is allowed. If set to “http”, access is denied. If set to any other value, access is allowed. Calendar Server uses this attribute only if the icsStatus attribute is not set.
icsDefaultAccess	userDefaultAcl	Specifies the ACL for a newly created user calendar. Corresponding ics.conf parameter: calstore.calendar.default.acl
icsDomainNames	searchDomainNames	Specifies the external domains that this domain can search when looking for calendars or users. Corresponding ics.conf parameter: none
icsDWPBackEndHosts		Specifies the default back-end host (DNS name) for a user if a host name is not explicitly provided. This attribute is used when Calendar Server is in LDAP CLD mode.

Table 11-17 Other LDAP Directory Attributes for the csdomain Utility *(Continued)*

LDAP Attribute	Property Name	Description
icsStatus	statusCalendarDomain	Specifies that status of Calendar Server: • active–Calendar Server is accessible. • inactive–Calendar Server is inaccessible. Calendars remain in the database and Calendar Server LDAP attributes remain unchanged. • deleted–Calendar Server is inaccessible. because the person is marked as deleted. • removed–Calendars have been removed from the calendar database. If icsStatus is set, its value overrides the icsAllowedServiceAccess attribute. If icsStatus is not set, Calendar Server uses the icsAllowedServiceAccess attribute.
icsTimezone	timezone	Specifies the time-zone ID used when importing files. For example, America/New_York or Asia/Tokyo. For the supported time zones, refer to the timezones.ics file.

Examples

- Create a new hosted domain using LDAP schema v.1 named `west.sesta.com`:

  ```
  csdomain -v -n o=nodewest,o=sesta create west.sesta.com
  ```

- Create a new hosted domain using LDAP schema v.2 named `east.sesta.com`:

  ```
  csdomain -v -n nodeeast create east.sesta.com
  ```

- Display a list of Calendar Server LDAP attributes for the hosted domain named `west.sesta.com`:

  ```
  csdomain -v list west.sesta.com
  ```

- Set the time zone to `America/New_York` for the hosted domain named `west.sesta.com`:

  ```
  csdomain -v -a icsTimezone=America/New_York add west.sesta.com
  ```

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 Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

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

Table 11-18 describes the commands available for the `csexport` utility.

Table 11-18 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 11-19 describes the `csexport` utility command options.

Table 11-19 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 Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

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

Table 11-20 describes the commands available for the `csimport` utility.

Table 11-20 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 11-21 describes the `csimport` utility command options.

Table 11-21 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 51.

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
  ```

csmonitor

The `csmonitor` utility is a script that performs these monitoring functions:

- Checks LDAP connectivity according to specific `ics.conf` parameters.
- Checks the transaction log files, and if more than one file exists, `csmonitor` sends a warning by email.
- Checks the available disk space for the calendar database.

Syntax

```
csmonitor [ -k | -c | -help]
```

Table 11-22 describes the commands for the `csmonitor` utility.

Table 11-22 csmonitor Utility Command Options

Parameter	Description
`-k`	Stops csmonitor.
`-c`	Backs up the csmonitor log and purges it.
`-help`	Displays the csmonitor usage.

Table 11-23 csmonitor.

Table 11-23 csmonitor Configuration Parameters

Parameter	Description
service.monitor.continuous	Specifies whether csmonitor should loop continuously: "0"–Do not loop continuously. "1"–Loop continuously. This option requires more system resources, but it can useful in debug mode. Default is "0".
service.monitor.loopsdelay	Specifies the delay in seconds between two monitoring loops. Default is "60".

Table 11-23 csmonitor Configuration Parameters *(Continued)*

Parameter	**Description**
service.monitor.emailaddress.from	Specifies the email address csmonitor sends messages from. Default is none.
service.monitor.emailaddress.to	Specifies the email address csmonitor should send messages to. Default is none.
service.monitor.csdb.logthreshold	Specifies a threshold value in percent of the total disk space for the maximum disk occupation considered to be normal. If the disk occupation where the calendar database (csdb directory) resides exceeds this value, csmonitor sends a warning email message. Default is "90".
logfile.monitor.logname	Specifies the csmonitor log file name. Default is "csmonitor.log".
logfile.monitor.maxlogfilesize	Specifies the maximum log file size. If the log file exceeds this size, csmonitor saves the log as `csmonitor.log.`*timestamp* and resets the log. Default is "2097152".
service.monitor.dbglevel	Specifies the debug level. The higher this value, `csmonitor` sends more precise and verbose messages. Default is "0".

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 11-25.)
- `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 Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

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


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

Table 11-24 describes the commands available for the `csplugin` utility.

Table 11-24 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 11-25.)
`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 11-25.)
`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 11-25.)
`version`	Display the version of the utility.

Table 11-25 describes the `csplugin` utility command options.

Table 11-25 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.

Table 11-25 csplugin Utility Command Options *(Continued)*

Option	Description
`-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
  ```

cspurge

The cspurge utility allows the manual purge of entries in the Delete Log database (`ics50deletelog.db`).

Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

```
cspurge [-q|-v] -e endtime -s starttime
```

Table 11-26 describes the `cspurge` utility command options.

Table 11-26 cspurge 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*	Specifies the ending time in GMT (also referred to as UTC or Zulu). This value is up to (less than) the specified time. The default is 0, which means to the end of time.

Table 11-26 cspurge Utility Command Options *(Continued)*

Option	Description
`-s` *starttime*	Specifies the starting time in GMT (also referred to as UTC or Zulu). This value includes (greater than or equal to) the specified time. The default is 0, which means from the beginning of time.

Examples

- Purge all entries in the Delete Log:

  ```
  cspurge -v -e 0 -s 0
  ```

- Purge all entries from July 1, 2003 through July 31, 2003:

  ```
  cspurge -v -e 20030731T235959Z -s 20030701T120000Z
  ```

- Purge all entries up to September 30, 2003:

  ```
  cspurge -v -e 20031030T235959Z -s 0
  ```

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 Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

Syntax

```
csresource [-q|-v] [-a aces] [-b basedn] [-d domain] [-t description] [-k yes|no]
[-o owner] [-y otherowners] -m email -c calid create name


csresource [-q|-v] [-b basedn] [-d domain]
delete|disable|enable [name]


csresource [-q|-v] [-b basedn] [-d domain] [-h host] list [name]
```

Table 11-27 describes the commands available for the `csresource` utility.

Table 11-27 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.
`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, display all resource calendars. If the -h *host* option is included, display the calendar attributes for the specified name (or all resource calendars) on that back-end server.

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

Table 11-28 describes the `csresource` utility command options.

Table 11-28 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.

Table 11-28 csresource Utility Command Options *(Continued)*

Option	Description
`-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".
`-b [`*basedn*`]`	LDAP base DN (distinguished name) to be used for the specified resource. The default is taken from the local.ugldapbasedn parameter 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 51.
`-d` *domain*	Specifies the name of a hosted (virtual) domain. Default is taken from the service.defaultdomain parameter in the ics.conf file.
`-t [`*description*`]`	Specifies a viewable comment about the purpose of the calendar. The default is no description.
`-h` *host*	Specifies the name of a back-end server where the resource calendar resides. This option applies only to the list command.
`-k yes|no`	Specifies whether double booking is allowed for a calendar associated with a resource such as a conference room. For example, yes means the resource calendar can have more than one event scheduled for the same time slot. If the -k option is omitted, the default is taken from the resource.allow.doublebook parameter in the ics.conf file. However, the resource.allow.doublebook parameter is used only when a calendar is created. After a calendar is created, Calendar Server checks the calendar properties (ics50calprops.db) to determine if double booking is allowed. If you need to change the calendar properties for a calendar to allow or disallow double booking, reissue csresource with the -k option.
-m *email*	Specifies the LDAP mail attribute (primary email address) for the resource.
`-o` *owner*	Primary owner. Default is taken from 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
  ```

- Display the LDAP attributes of the resource calendar with the viewable name `MeetingRoom100` on the back-end server `sesta`:

  ```
  csresource -v -h sesta list 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 6.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 Calendar Server is installed.
- If you are restoring the calendar database, Calendar Server must be stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

Syntax

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


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


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

Table 11-29 describes the commands available for the `csrestore` utility.

Table 11-29 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 11-30 describes the `csrestore` utility command options.

Table 11-30 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 default option. There is no default.
`-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.

Table 11-30 csrestore Utility Command Options *(Continued)*

Option	Description
`-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 51.

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 Calendar Server is installed.
- Calendar Server must be stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

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 11-31 describes the commands available for the `csschedule` utility.

Table 11-31 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 11-32 describes the `csschedule` utility command options.

Table 11-32 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.
`-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, recurrence ID 0, and 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 10:30:45, the first offset at time 10:30:45 on 12/31/2000, with the unique identifier 1111, recurrence ID 0, and 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 Calendar Server provides the `csstart` and `csstop` utilities only to provide compatibility with earlier releases.

It is recommended that you use the `start-cal` and `stop-cal` commands to start and stop Calendar Server. For more information, see "Starting and Stopping Calendar Server" on page 68.

The `csstart` utility starts 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.

Requirements

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

Syntax

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


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

Table 11-33 describes the commands available for the `csstart` utility.

Table 11-33 csstart 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 Sun ONE 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 HTTP service
`service [`*servicename*`]`	Start 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: 1. ens — a generic event registration and notification service that can be shared by other Sun ONE 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 HTTP service For more information about Calendar Server services, see the following sections: • "Calendar Server Access Control". • "Starting and Stopping Calendar Server" on page 68." • "Services Configuration" on page 259.
`version`	Display the version of the utility.

Table 11-34 describes the `csstart` utility command options.

Table 11-34 csstart Utility Command Options

Optio n	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). 2. Cleanup any database problems. 3. Start all services. **Note**: It is recommended that you use the stop-cal and start-cal utilities instead to stop and start 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 291.

Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

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

Table 11-35 describes the commands available for the `csstats` utility.

Table 11-35 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 11-36 describes the `csstats` utility command options.

Table 11-36 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: /opt/SUNWics5/cal/lib/counter/counter
`-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 Calendar Server provides the `csstop` and `csstart` utilities only to provide compatibility with earlier releases.

It is recommended that you use the `start-cal` and `stop-cal` utilities to start and stop Calendar Server. For more information, see "Starting and Stopping Calendar Server" on page 68.

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

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

Requirements

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

Syntax

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


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

Table 11-37 describes the commands available for the `csstop` utility..

Table 11-37 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 Sun ONE 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 HTTP 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 Sun ONE 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 HTTP service. For more information about Calendar Server services, see the following sections: • “Calendar Server Access Control”. • “Starting and Stopping Calendar Server” on page 68. • “Services Configuration” on page 259.
`version`	Display the version of the utility.

Table 11-38 describes the `csstop` utility command options.

Table 11-38 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 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
  ```

csstored.pl

The `csstored.pl` utility is a Perl script that performs the following online or “hot” archival operations for the calendar database and log files:

- Copies the database files and current log files to a backup archive directory and a “hot” backup directory.
- Copies the log files returned by the Berkeley database `db_archive` utility to a backup archive directory and a “hot” backup directory.
- Removes all log files returned by the Berkeley database `db_archive` utility, except the last file.
- Runs the Berkeley database `db_verify` on the database files in the “hot” backup directory.

For documentation about the Berkeley database utilities, refer to the Sleepycat Software Web site:

`http://www.sleepycat.com/docs/utility/index.html`

Requirements

- You must run the script from the following directory on the machine where Calendar Server is installed:

  cal_svr_base`/opt/SUNWics5/cal/sbin`

- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.
- Circular logging for Calendar Server must be disabled:

  ```
  caldb.berkeleydb.circularlogging = "no"
  ```

Syntax

```
perl csstored.pl
```

Table 11-39 describes the parameters in the ics.conf file used by the `csstored.pl` utility:

Table 11-39 Parameters in the ics.conf file Used by the `csstored.pl` Utility

Parameter	Description
caldb.berkeleydb.homedir.path	Specifies the path to the directory where the calendar database files are stored. The default is ".", which specifies this directory: *cal_svr_base*/var/opt/SUNWics5/csdb
caldb.berkeleydb.archive.path	Specifies the path for the calendar database archive directory. There is no default. To run csstored.pl, specify a path that is different from the calendar database directory and the "hot" backup directory.
caldb.berkeleydb.hotbackup.path	Specifies the path to the "hot" backup directory. There is no default. To run csstored.pl, specify a path that is different from the calendar database directory and the archive directory.
caldb.berkeleydb.archive.enable	Enables ("yes") or disables ("no") the archive option for the calendar database. The default is "no". To run csstored.pl, set this parameter to "yes".
caldb.berkeleydb.hotbackup.enable	Enables ("yes") or disables ("no") the "hot" backup option for the calendar database The default is "no". To run csstored.pl, set this parameter to "yes".
caldb.berkeleydb.circularlogging	Specifies ("yes" or "no") whether to remove database checkpoint files after their transactions are synchronized. The default is "yes". To run csstored.pl, set this parameter to "no".
caldb.berkeleydb.archive.interval	Specifies the interval in seconds for archiving the calendar database. The default is "120" seconds.

cstool

The cstool utility pings Calendar Server or either the cshttpd service or Event Notification Service (ENS). cstool also forces Calendar Server or a specific service to refresh its configuration. Commands are:

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

NOTE In the current release, do not use cstool refresh to refresh a configuration. Instead, use the stop-cal and start-cal utilities to stop and then restart Calendar Server.

For more information, see "Starting and Stopping Calendar Server" on page 68 and "Editing the ics.conf Configuration File" on page 249.

Requirements

- Calendar Server must be running.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

Syntax

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


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

Table 11-40 describes the commands available for the `cstool` utility.

Table 11-40 cstool Utility Commands

Command	Description
`ping [http|ens]`	Ping the cshttpd or ENS service.
`refresh [`*servicename*`]`	Force 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 (csadmind service) • dwp (csdwpd service) • http (cshttpd service) • notify (csnotifyd service)
`version`	Display the version of the utility.

Table 11-41 describes the `cstool` utility command options.

Table 11-41 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 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 ENS on the server with host name sesta:

  ```
  cstool -h sesta ping ens
  ```

- 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 Calendar Server.
- enable a user to log on to Calendar Server.
- list a user's calendar attributes.
- reset a user's calendar attributes to the default settings.

If you are using Sun ONE Directory Server, you can also use the ldapsearch and ldapmodify utilities. For information about these utilities, see the Sun ONE Directory Server documentation on the following Web site:

http://docs.sun.com/coll/S1_DirectoryServer_52

Requirements

- Calendar Server can be running or stopped.
- You must run the utility locally on the machine where Calendar Server is installed.
- The LDAP server that stores calendar user information must be running.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

Syntax

```
csuser [-q|-v] [-a aces] [-b basedn] -m email [-d domain] -f filename
-g givenname [-k yes|no] [-l langcode] -s surname -y userpassword
create userid


csuser [-q|-v] [-b basedn] [-d domain] [-h host] list [userid]


csuser [-q|-v] [-b basedn] [-d domain]
  [check|delete|disable|enable|reset] userid
```

Table 11-42 describes the commands available for the `csuser` utility.

Table 11-42 csuser Utility Commands

Command	Description
`check` *userid*	Check if the specified user ID is enabled for calendaring.
`create` *userid*	Create the specified user ID and enable this user to log into Calendar Server.
`delete` *userid*	Delete the specified user ID.
`disable` *userid*	Disable the specified user ID for calendaring. The utility defines http as the value of the nswcalDisallowAccess attribute.
`enable` *userid*	Enable the specified user ID for calendaring. (The utility adds the specified calendar ID to the nswcalCALID attribute.)
`list` [*userid*]	List the calendar attributes for the specified user ID. If user ID is not specified, list attributes for all enabled users. If the -h *server-name* option is included, list the calendar attributes for the specified user ID (or all enabled users) on that back-end server.
`reset` *userid*	Reset all calendar attributes for a user ID to their default settings. **Note**: After the calendar attributes for a user ID have been reset, all of the calendar attributes are removed from the user's LDAP entry, including icsCalendarUser (object class), icsSubscribed, icsCalendarOwned, icsCalendar, and icsDWPHost (if the user is in an LDAP CLD setup). A Calendar Server administrator then cannot create calendars on the user's behalf. These attributes are restored in the user's LDAP entry when the Calendar Server administrator issues a csuser enable command for the user.

Table 11-42 csuser Utility Commands *(Continued)*

Command	Description
`version`	Display the version of the utility.

Table 11-43 describes the `csuser` utility command options.

Table 11-43 csuser 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.
`-b` *basedn*	The base DN to be used for all LDAP users. The default value is taken from the setting `local.ugldapbasedn` defined in the `ics.conf` file.
`-d` *domain*	Specifies the name of a hosted (virtual) domain. Default is taken from the service.defaultdomain parameter in the ics.conf file.
`-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 (""). Default is: "@@o^a^r^g;@@o^c^wdeic^g;@^a^sf^g;@^c^^g;@^p^r^g" For details about the ACE format, see "Services Configuration" on page 259.
`-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 option is not valid for the create command. For more information, see "Calendar Identifiers (calids)" on page 51.
`-f` *filename*	File name to specify a password for options that require a password (-y parameter). If you are running csuser from a script, it is recommended for added security that you specify the password in filename.

Table 11-43 csuser Utility Command Options *(Continued)*

Option	Description
`-g` *givenname*	The user's LDAP given name (first name). This option is required. There is no default.
`-h` *host*	Specifies the name of a back-end server where the user's calendar resides. This option applies only to the list command.
`-p` *port*	The port number that LDAP server is listening to. The default value is taken from the setting local.ugldapport defined in the ics.conf file.
`-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.
-m *email*	Specifies the LDAP mail attribute (primary email address) for the user.
`-s` *surname*	The user's LDAP surname (last name). This option is required. There is no default.
`-u` *adminDN*	The LDAP distinguished name (DN) of the person with administration rights to the LDAP server that stores user authentication information accesses by this Calendar Server. The default value is taken from the configuration setting local.enduseradmindn defined in the ics.conf file.
`-w` *password*	The password of the LDAP administrator DN specified by the -u option. The default value is taken from the configuration setting local.enduseradmincred defined in the ics.conf file. This option is required if you use the -u option.
`-y` *userpassword*	The required calendar user's password. 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, email address jsmith@sesta.com, and the domain sesta.com:

  ```
  csuser -g John -s Smith -y password -m jsmith@sesta.com create
  JSmith -d sesta.com
  ```

- Delete the calendar user JSmith:

  ```
  csuser delete JSmith
  ```

- Disable the calendar user JSmith from logging in to Calendar Server:

  ```
  csuser disable JSmith
  ```

NOTE This command prevents JSmith from logging into Calendar Server to access calendar data, but it does not delete JSmith's data from the calendar database. If JSmith is currently logged into Calendar Server, JSmith retains access to calendar data until he logs off.

- Enable JSmith for calendaring (lets existing calendar user JSmith log in to 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
  ```

- List all calendar attributes for TChang on the back-end server `sesta`:

  ```
  csuser -v -h sesta list TChang
  ```

start-cal

The start-cal utility starts the Calendar Server services in this order:

- enpd — Event Notification Service (ENS)
- csnotifyd — Notification Service
- csadmind — Administration Service
- csdwpd — Database Wire Protocol (DWP) service, the distributed database service that is started only with a remote Calendar Server database configuration
- cshttpd — HTTP Service

Requirements

- You must run `start-cal` locally on the machine where Calendar Server is installed.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

```
start-cal
```

Example

```
./start-cal
```

For more information, see "Starting and Stopping Calendar Server" on page 68.

stop-cal

The stop-cal utility stops all Calendar Server services.

Requirements

- You must run `stop-cal` locally on the machine where Calendar Server is installed.
- On Solaris Systems, you must be logged in as the user and group under which Calendar Server is running (such as `icsuser` and `icsgroup`) that was specified during installation, or as `root`.

Syntax

```
stop-cal
```

Example

```
./stop-cal
```

For more information, see “Starting and Stopping Calendar Server” on page 68.

Chapter 12

Calendar Server Configuration Parameters

Sun ONE 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:

cal_svr_base`/etc/opt/SUNWics5/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` file only as described in Sun documentation or as directed by a customer support representative.

To edit the ics.conf file:

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

2. Change to the *cal_svr_base*`/etc/opt/SUNWics5/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 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 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 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 Calendar Server.

   ❍ If a parameter is not in the `ics.conf` file, add the parameter and its associated value to the file.

   ❍ 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 Calendar Server before you edit the `ics.conf` file.)

   For more information, see "Starting and Stopping Calendar Server" on page 68.

Configuration Parameters (ics.conf) File

Configuration parameters in the `ics.conf` file include:

- Local Configuration
- Calendar Store Configuration
- Calendar Log Information Configuration
- Services Configuration
- Hosted Domain Configuration
- Alarm Notification Configuration
- Calendar Lookup Database Configuration
- Single Sign-on (SSO) Configuration
  - Configuring SSO Through Identity Server
  - Configuring SSO Through Communications Servers Trusted Circle Technology
- Group Scheduling Engine (GSE) Configuration
- Database Configuration
- Calendar Database Parameters for ENS Messages
- Calendar Server API Configuration
- Event Notification Server (ENS) Configuration
- User Interface (UI) Configuration
- csmonitor Utility Configuration

Local Configuration

The following table shows the Local Configuration parameters with each parameter’s default value and description.

Table 12-1 Local Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
local.autoprovision	"yes"	Enables ("yes") or disables ("no") auto provisioning of Calendar Server users.
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.

Table 12-1 Local Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
local.hostname	" "	Host name of the machine on which Calendar Server is installed.
local.installeddir	" "	Directory path location where Calendar Server is installed. The default is *cal_svr_base*/opt/SUNWics5/cal.
local.instancedir	"."	Directory path location where programs and data for this instance of Calendar Server are installed. The default is *cal_svr_base*/opt/SUNWics5/cal/sbin.
local.plugindir.path	" "	Directory path location where CSAPI plug-ins for this instance of 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"	Group ID (GID) for Calendar Server files, such as counters and logs.
local.serveruid	"icsuser"	User ID (UID) for Calendar Server files, such as counters and logs.
local.sitelanguage	"en"	Default language for this instance of Calendar Server.
local.smtp.defaultdomain	" "	Name of the default domain used to lookup an attendee's calendar-id that corresponds to an email 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 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.
local.instance.lockdir.path	*cal_svr_base*/opt/SUNWics5/cal/lib/lock	Specifies the location where lock files for this server instance are stored.
local.instance.pidfile.path	*cal_svr_base*/etc/opt/SUNWics5/config	Specifies the location where PID files for this server instance are stored.
local.instance.counter.path	*cal_svr_base*/opt/SUNWics5/cal/lib/counter	Specifies the location where counter files for this server instance are stored.

Table 12-1 Local Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
local.instance.use.tmpfs	"false"	If "true" overlays a memory-based file system (tmpfs) for the session database based on the values of service.http.sessiondir.path and service.admin.sessiondir.path.
local.ugldapicsextendeduserprefs	"ceColorSet, ceFontFace, ceFontSizeDelta, ceDateOrder, ceDateSeparator, ceClock, ceDayHead, ceDayTail, ceInterval, ceToolText, ceToolImage, ceDefaultAlarmStart, ceSingleCalendarTZID, ceAllCalendarTZIDs, ceDefaultAlarmEmail, ceNotifyEmail, ceNotifyEnable, ceDefaultView, ceExcludeSatSun, ceGroupInviteAll"	Values for the Calendar Server 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.
local.domain.language	"en"	Language for the domain.

Calendar Store Configuration

The following table shows the Calendar Store Configuration parameters with each parameter's default value and description.

Table 12-2 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 calendar can have more than one event scheduled for the same time slot when the calendar is created: • `"no"` prevents double booking. • `"yes"` allows double booking. This parameter is used only when a user calendar is created. After a user calendar is created, Calendar Server checks the calendar properties (ics50calprops.db) to determine if double booking is allowed. If you need to change the calendar properties for a user calendar to allow or disallow double booking, use cscal with the -k option.
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 184.
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 12-2 Calendar Store Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
calstore.calendar.create.lowercase	"no"	Specifies whether Calendar Server should convert a calendar ID (`calid`) to lowercase when creating a new calendar or when looking up a calendar using the LDAP CLD plug-in.
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 Calendar Server filters (recognizes) Private and Time and Date Only (confidential) events and tasks. If `"no"`, 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 Calendar Server is in limited virtual domain mode ("y" or "n").

Table 12-2 Calendar Store Configuration Parameters in the ics.conf File *(Continued)*

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

Calendar Log Information Configuration

The following table shows the Calendar Log Configuration parameters with each parameter's default value and description.

Table 12-3 Calendar Log Configuration 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, Calendar 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 (in bytes) that must be available for logging. When this value is reached, Calendar Server attempts to free disk space by expiring old log files. Logging is paused if space cannot 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

The following table shows the Services Configuration parameters with each parameter's default value and description.

Table 12-4 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 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 when the calendar is created: • "no" prevents double booking. • "yes" allows double booking. This parameter is used only when a resource calendar is created. After a resource calendar is created, Calendar Server checks the calendar properties (ics50calprops.db) to determine if double booking is allowed. If you need to change the calendar properties for a resource calendar to allow or disallow double booking, use csresource with the -k option.
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 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.

Table 12-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.admin.checkpoint	"yes"	If "yes", start the csadmind database checkpoint thread.
service.admin.dbcachesize	"8388608"	Maximum cache size (in bytes) for Berkeley Database for administration sessions.
service.admin.deadlock	"yes"	If "yes", start the csadmind database deadlock detection thread.
service.admin.diskusage	"no"	If "yes", start the csadmind low disk space monitor thread.
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.numprocesses		Maximum number of a concurrent administration processes allowed.
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.
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.

Table 12-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.dcroot	"o=internet"	Root suffix of the DC tree in the directory. Required for hosted (virtual) domain mode support. See also "Hosted Domain Configuration Parameters" on page 145.
service.domainname	" "	Domain name for this installation.
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 Calendar Server is running, such as mycal@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).
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.

Table 12-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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 concurrently running HTTP service (`csdwpd`) processes that should run on a server. For a server that has multiple CPUs, see "Using Load Balancing Across Multiple CPUs" on page 323.
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.
service.http.tmpdir	"/var/opt/SUNWics5/tmp"	Directory relative to executable where all URL references to files are stored.
service.http.uidir.path	"html"	Directory that contains the default calendar client. If allowing only WCAP access, set to "".
service.http.renderhtml	"no"	Enables or disables rendering of HTML and JavaScript in the Event Title and Description fields for events in Calendar Express. Rendering does not apply to any other text fields for events or to any text fields for tasks.
service.ldapmemcache	"no"	If "yes", use cache in LDAP SDK.

Table 12-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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"	On Solaris Systems, 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.format	"text/calendar"	Specifies the default output format for commands currently applied only for freebusy.
service.wcap.freebusybegin	"30"	Specifies the default offset from the current time in days for get_freebusy for beginning of the range.
service.wcap.freebusyend	"30"	Specifies the default offset from the current time in days for get_freebusy for end of the range.
service.wcap.allowcreatecalendars	"yes"	If "yes", allow calendars to be created.
service.wcap.allowdeletecalendars	"yes"	If "yes", allow calendars to be deleted.
service.wcap.allowchangepassword	"no"	If "yes", allow users to change their passwords via this server.
service.wcap.allowpublicwritablecalendars	"yes"	If "yes", allow users to have publicly writable calendars.
service.wcap.allowsetprefs.cn	"no"	If "yes", allow the set_userprefs.wcap attribute to modify the user preference "cn" (LDAP user's common name).
service.wcap.allowsetprefs.givenname	"no"	If "yes", allow the set_userprefs.wcap attribute to modify the user preference "givenname" (LDAP user's given name).
service.wcap.allowsetprefs.icsCalendar	"no"	If "yes", allow the set_userprefs.wcap attribute to modify the user preference “icsCalendar" (a user's default calendar identifier).

Table 12-4 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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.userprefs.ldapproxyauth	"no"	If "yes", enables LDAP proxy authorization for get_userprefs.wcap command. If "no", anonymous LDAP search is performed.
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).
service.wcap.version	"3.0"	WCAP version.

SSL Configuration

The following table shows the SSL Configuration parameters with each parameter’s default value and description.

Table 12-5 Configuration Parameters for SSL

Parameter	Default Value	Description
service.http.ssl.cachedir	"."	Specifies the physical path location for the SSL cache.
service.http.ssl.usessl	"no"	Specifies whether the cshttpd process should use the SSL subsystem.
service.http.ssl.port.enable	"no"	Specifies whether the cshttpd process should listen only on the SSL port.
service.http.ssl.port	"443"	Specifies the port number for the cshttpd process where HTTPS requests from Calendar Server users are received.
service.http.ssl.securelogin	"yes"	Specifies whether to encrypt the login.
service.http.ssl.securesession	"yes"	Specifies whether to encrypt the entire session.
service.http.ssl.certdb.path	"alias"	Specifies the physical path location of the SSL Certificate Database.
service.http.ssl.certdb.password	"*password*"	Specifies the SSL Certificate Database access password.
service.http.ssl.sourceurl	"https://*localhost*:443"	Specifies the SSL host name and port number for the originating source URL.
service.http.ssl.ssl2.ciphers	""	Specifies ciphers for SSL2.
service.http.ssl.ssl2.sessiontimeout	"0"	Specifies the session timeout for SSL2.
service.http.ssl.ssl3.ciphers		Specifies a list of supported or valid SSL ciphers. Default is: "rsa_rc4_40_md5,rsa_rc2_40_md5,rsa_des_sha,rsa_rc4_128_md5,rsa_3des_sha"
service.http.ssl.ssl3.sessiontimeout	"0"	Specifies the timeout value for the SSL session.
service.http.ssl.cachesize	"10000"	Specifies the maximum size of the SSL cache database.
encryption.rsa.nssslactivation	"on"	Enables the RSA Cypher Encryption Family Services for SSL.
encryption.rsa.nsssltoken	"internal"	Specifies the location of the RSA Cypher Encryption Family token.
encryption.rsa.nssslpersonalityssl	"SampleSSLServerCert"	Specifies the certificate name for the RSA Cypher Encryption Family.

Hosted Domain Configuration

The following table shows the hosted (virtual) domain configuration parameters with each parameter's default value and description.

Table 12-6 Configuration Parameters for Hosted Domain Support

Parameter	Default Value	Description
service.virtualdomain.support	"n"	Enables ("y") or disables ("n") support for hosted (virtual) domains.
local.schemaversion	"1"	Specifies the version of the LDAP schema: • "1" = Sun ONE LDAP Schema, v.1. See also service.dcroot. • "2" = Sun ONE LDAP Schema, v.2. See also service.schema2root.
service.dcroot	" "	Specifies the root suffix of the DC tree in the LDAP directory, if local.schemaversion = "1". For example: "o=internet"
service.schema2root	" "	Specifies the root suffix underneath which all domains are found, if local.schemaversion = "2". For example: "o=sesta.com"
service.defaultdomain	" "	Specifies the default domain for this instance of Calendar Server. Used when a domain name is not supplied during a login. For example: "sesta.com".
service.loginseparator	"@+"	Specifies a string of separators used for the *login-separator* when Calendar Server parses "userid[*login-separator*]domain". Calendar Server tries each separator in turn.
service.siteadmin.userid	" "	Specifies the user ID of the domain administrator.

Alarm Notification Configuration

The following table shows the Alarm Notification Server Configuration parameters with each parameter's default value and description.

Table 12-7 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.
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.

Table 12-7 Alarm Notification Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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 Lookup Database Configuration

The following table shows the Calendar Lookup Database (CLD) parameters with each parameter's default value and description.

Table 12-8 Calendar Lookup Database (CLD) 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 ("*"), Calendar Server loads all plug-ins.
caldb.cld.type	"local"	Specifies the type of CLD plug-in to use: •"local" specifies that all calendars are stored on the local server where Calendar Server is running and no plug-in is loaded. •"directory" specifies the LDAP CLD plug-in, which uses a calendar owner's icsDWPHost LDAP attribute to determine the server where a specified calendar is stored.
caldb.dwp.server.default= "server-name"	" "	Specifies the fully qualified default DWP server name used by Calendar Server if a user or resource calendar entry in the LDAP server database does not have an `icsDWPHost` attribute. If a user who logs into Calendar Server through Calendar Express or using WCAP commands does not have an `icsDWPHost` attribute, Calendar Server uses this parameter to auto-provision the attribute. If a user already has an `icsDWPHost` attribute, `caldb.dwp.server.default` is not used. This name must be resolvable by your Domain Name Service (DNS) into a valid IP address.
caldb.cld.cache.enable	"yes"	Enables ("yes") or disables ("no") the Calendar Lookup Database (CLD) cache option. For optimum performance for the LDAP CLD plug-in, set to "yes".
caldb.cld.cache.logfilesizemb	"10"	Specifies the maximum size in megabytes of the checkpoint file for the CLD cache.
caldb.cld.cache.mempoolsizemb	"4"	Specifies the size in megabytes of shared for the CLD cache option.
caldb.cld.cache.maxthread	"1000"	Specifies the maximum number of database threads for the CLD cache option.

Table 12-8 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
caldb.cld.cache.homedir.path	"."	Specifies the location of database event, task, and alarm files for the CLD cache option. The default value of "." specifies that these files are stored in the *cal_svr_base*/var/opt/SUNWics5/csdb/cld_cache directory.
caldb.cld.cache.checkpointinterval	"60"	Specifies the number of seconds between checkpointing for the CLD cache option.
caldb.cld.cache.circularlogging	"yes"	Specifies whether to remove the checkpoint files after they are synchronized for the CLD cache option.
`caldb.dwp.server.`*host-name*`.ip = "`*host-name*`"`	" "	Specifies the host name of a server that is storing a calendar database. The server must be running the DWP (csdwpd) service. This name must be resolvable by your Domain Name Service (DNS) into a valid IP address. This parameter is used by the LDAP CLD plug-in. **Note**: In each part of the parameter, *host-name* must be identical and fully qualified. For example: caldb.dwp.server.sesta.com.ip ="sesta.com"
caldb.dwp.connthreshold	"1"	Maximum number of backlogged requests before the server obtains a new network connection.
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 (DWP) service at the specified machine’s hostname.

Table 12-8 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
caldb.dwp.server.*hostname.*port	"9779"	Specifies the port number of the server using the Database Wire Protocol (DWP) service at the specified machine's hostname.
caldb.dwp.server.*back-end-server*.admin	" "	On a front-end server, specifies the user ID that is used for authentication for a DWP connection to a back-end server, where *back-end-server* is the name of the server.
caldb.dwp.server.*back-end-server*.cred	" "	On a front-end server, specifies the password that is used for authentication for a DWP connection to a back-end server, where *back-end-server* is the name of the server.
caldb.dwp.stacksize	"65536"	Stack size for Database Wire Protocol service threads.
caldb.cld.directory.ldapbasedn		Base DN to authenticate against if LDAP plugin is used for the calendar locate mechanism.
caldb.cld.directory.ldaphost		Host name of the LDAP server to access if an LDAP plugin is used for the calendar locate mechanism.
caldb.cld.directory.ldapbindcred		Bind credentials (password) for the user specified in the setting `local.authldapbinddn` if an LDAP plugin is used for the calendar locate mechanism.
caldb.cld.directory.ldapbinddn		DN used to bind to for authentication to search for user's DN if an LDAP plugin is used for the calendar locate mechanism.
caldb.cld.directory.ldapport	"389"	Port number of the LDAP server to access if an LDAP plugin is used for the calendar locate mechanism.
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.
logfile.dwp.buffersize	"0"	Size of Database Wire Protocol service log buffers (in bytes).
logfile.dwp.expirytime	"604800"	Number of seconds before the Database Wire Protocol service log files expire.
logfile.dwp.flushinterval	"60"	Number of seconds between flushing buffers to the Database Wire Protocol service log files.

Table 12-8 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
logfile.dwp.logdir	"logs"	Directory location of the Database Wire Protocol service log files.
logfile.dwp.loglevel	"Notice"	Determines the level of detail the server will log for the Database Wire Protocol service. Each Database Wire Protocol log entry is assigned one of the following levels (starting with the most severe): Critical, Error, Warning, Notice, Information, and Debug. If you set this preference to Critical, the server will log the least amount of detail. If you want the server to log the most amount of detail, specify Debug. For example, if you specify Warning, only Critical, Error, and Warning level log entries are logged.
logfile.dwp.maxlogfiles	"10"	Maximum number of Database Wire Protocol related log files in log directory.
logfile.dwp.maxlogfilesize	"2097152"	Maximum size of each Database Wire Protocol log file (in bytes).
logfile.dwp.maxlogsize	"20971520"	Maximum disk space for all Database Wire Protocol log files (in bytes).
logfile.dwp.minfreediskspace	"5242880"	Minimum free disk space that must be available for logging Database Wire Protocol service activity (in bytes). When this value is reached, the server will attempt to free disk space by expiring old log files. All logging will be paused if no space can be freed up.
logfile.dwp.rollovertime	"86400"	Number of seconds before Database Wire Protocol service log files are rotated.
service.dwp.admin.userid	" "	On a back-end server, specifies the user ID that is used to authenticate a DWP connection. This parameter is optional. If a back-end server does not specify a user ID, no authentication is performed.
service`.dwp.admin.cred`	" "	On a back-end server, specifies the password that is used to authenticate a DWP connection. This parameter is optional. If a back-end server does not specify a password, no authentication is performed.
service.dwp.calendarhostname	"localhost"	The hostname of the machine on which the Database Wire Protocol service is running.
service.dwp.maxthreads	"1000"	Maximum number of concurrently running Database Wire Protocol service threads.

Table 12-8 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.dwp.numprocesses	"1"	Maximum number of concurrently running Database Wire Protocol (DWP) service (csdwpd) processes that should run on a server. For a server that has multiple CPUs, see "Using Load Balancing Across Multiple CPUs" on page 323.
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	"59779"	Port number that the Database Wire Protocol service listens to. This value is the default port for the LDAP CLD plug-in.
service.dwp.ldap.enable	"yes"	Enable (`"yes"`) or disable (`"no"`) LDAP for remote user authentication for the Database Wire Protocol (`csdwpd`) service.
service.calendarsearch.ldap	"yes"	Specifies whether Calendar Server searches the LDAP directory and then the calendar database ("yes") or only the calendar database ("no").

Single Sign-on (SSO) Configuration

- Configuring SSO Through Identity Server
- Configuring SSO Through Communications Servers Trusted Circle Technology

Configuring SSO Through Identity Server

The following table shows the SSO configuration parameters with each parameter's default value and description when you are using Identity Server.

Table 12-9 SSO Configuration Parameters in the ics.conf File (Through Identity Server)

Parameter	Default	Description
local.calendar.sso.singlesignoff	"yes"	Enables ("yes") or disables ("no") SSO for Calendar Server.
local.calendar.sso.amcookiename	"iPlanetDirectoryPro"	Specifies the name of the Identity Server SSO cookie.
local.calendar.sso.amnamingurl	"http://*IdentityServer*:*port*/amserver/namingservice"	Specifies the URL of the Identity Server SSO naming service.
local.calendar.sso.amloglevel	"3"	Specifies the log level for Identity Server SSO. Range is from 1 (quiet) to 5 (verbose).
local.calendar.sso.logname	"am_sso.log"	Specifies the name of the Identity Server SSO API log file.

Configuring SSO Through Communications Servers Trusted Circle Technology

The following table shows the SSO configuration parameters with each parameter's default value and description when the Communications Servers trusted circle technology.

Table 12-10 SSO Configuration Parameters in the ics.conf File Using Communications Servers Trusted Circle Technology

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="ics50"
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 Calendar Server. The application prefix must not end with a hyphen (-), because Calendar Server appends a hyphen to the value. For example: sso.appprefix="ssogrp1"
sso.*appid*.url		Verification URL for the value specified for sso.appid. For example: sso.ics50.url="http://siroe.com:80/default.html"
sso.*nnn*.ip	sso.*appid*.ip	IP address of the value specified for sso.appid. For example: sso.ics50.ip="123.12.456.123"
sso.cookiedomain	"."	Causes the browser to send a cookie only to servers in the specified domain. 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 parameter is missing from `ics.conf`, 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.

Table 12-10 SSO Configuration Parameters in the ics.conf File Using Communications Servers Trusted Circle Technology *(Continued)*

Parameter	Default Value	Description
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 Calendar Server on sesta.com with port number 8883: sso.ics50.url = "http://sesta.com:8883/VerifySSO?"

Group Scheduling Engine (GSE) Configuration

The following table shows the Group Scheduling Engine (GSE) configuration parameters with each parameter's default value and description.

Table 12-11 Group Scheduling Engine (GSE) 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

The following table shows the Database configuration parameters with each parameter’s default value and description.

Table 12-12 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 stored. The default is “.”, which specifies: *cal_svr_base*/var/opt/SUNWics5/csdb
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.
caldb.smtpmsgfmtdir	"en"	Specifies the directory under *cal_svr_base*/etc/opt/SUNWics5/config that contains the localized version of the files used to format email notifications. For example, "en" specfies the directory for the English localized version, and "fr" specifies the directory for the French localized version.
caldb.smtpport	"25"	Port for SMTP host.

Calendar Database Parameters for ENS Messages

Table 12-13 describes the parameter, default value, and description. If you require a value other than the default, you must add the parameter to the `ics.conf` file.

Table 12-13 Calendar Database Parameters for ENS Messages

Parameter	Default Value	Description
caldb.serveralarms.url	"enp:///ics/alarm"	Specifies the URL for the ENS message.
caldb.serveralarms.contenttype	""	Specifies the content type of the message data. Value can be "text/xml" or"text/calendar".
caldb.berkeleydb.ensmsg.createcal	"no"	Creates an ENS message when a calendar is created.
caldb.berkeleydb.ensmsg.createcal.url	"enp:///ics/calendarcreate"	Specifies the URL for the ENS message.
caldb.berkeleydb.ensmsg.createcal.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.deletecal	"no"	Creates an ENS message when a calendar is deleted.
caldb.berkeleydb.ensmsg.deletecal.url	"enp:///ics/calendardelete"	Specifies the URL for the ENS message.
caldb.berkeleydb.ensmsg.deletecal.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.modifycal	"no"	Creates an ENS message when a calendar is modified.
caldb.berkeleydb.ensmsg.modifycal.url	"enp:///ics/calendarmodify"	URL for the ENS message.
caldb.berleleydb.ensmsg.modifycal.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.createevent	"no"	Creates an ENS message when an event is created.
caldb.berkeleydb.ensmsg.createevent.url	"enp:///ics/caleventcreate"	Specifies the URL for the ENS message.

Table 12-13 Calendar Database Parameters for ENS Messages *(Continued)*

Parameter	Default Value	Description
caldb.berleleydb.ensmsg.createevent.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.modifyevent	"no"	Creates an ENS message when an event is modified.
caldb.berkeleydb.ensmsg.modifyevent.url	"enp:///ics/caleventmodify"	Specifies the URL for the ENS message.
caldb.berleleydb.ensmsg.modifyevent.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.deleteevent	"no"	Creates an ENS message when an event is deleted.
caldb.berkeleydb.ensmsg.deleteevent.url	"enp:///ics/caleventdelete"	Specifies the URL for the ENS message.
caldb.berkeleydb.ensmsg.deleteevent.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.createtodo	"no"	Creates an ENS message when a todo is created.
caldb.berkeleydb.ensmsg.createtodo.url	"enp:///ics/caltodocreate"	Specifies the URL for the ENS message.
caldb.berleleydb.ensmsg.createtodo.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.modifytodo	"no"	Creates an ENS message when a todo is modified.
caldb.berkeleydb.ensmsg.modifytodo.url	"enp:///ics/caltodomodify"	Specifies the URL for the ENS message.
caldb.berleleydb.ensmsg.modifytodo.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".
caldb.berkeleydb.ensmsg.deletetodo	"no"	Creates an ENS message when a todo is deleted.
caldb.berkeleydb.ensmsg.deletetodo.url	"enp:///ics/caltododelete"	Specifies the URL for the ENS message.
caldb.berkeleydb.ensmsg.deletetodo.contenttype	"text/xml"	Specifies the content type of the message data: "text/xml" (default) or "text/calendar".

Calendar Server API Configuration

Table 12-14 shows the Calendar Server API (CSAPI) configuration parameters with each parameter's default value and description.

Table 12-14 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.
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.

Table 12-14 CSAPI Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
csapi.plugin.dbtranslator.name	"*"	If the setting csapi.plugin.dbtranslator is "y", then either: load all the database-to-output format 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 Solaris 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.
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

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. 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 12-15 shows the Event Notification Server (ENS) configuration parameters in `ics.conf`. with each parameter's default value and description.

Table 12-15 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	"57997"	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.
service.notify.enable	"yes"	If "yes", start the csnotifyd service when starting all services and stop csnotifyd when stopping all services.
service.notify.maxretrytime	"-1"	How many times csnotifyd will consecutively retry and fail to contact the Event Notification Server. "-1" causes the alarm thread to retry indefinitely.
service.notify.retryinterval	"3"	Number (in seconds) that `csnotifyd` waits before attempting to re-contact the Event Notification Server after a connection failure.

Table 12-15 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.notify.startupretrytime	"0"	Total number of seconds Calendar Server keeps trying to contact the Event Notification Server before it stops. This setting is similar to caldb.serveralarms.maxretrytime except that it applies only when the alarm thread is first starting. Once the alarm thread has successfully started, caldb.serveralarms.maxretrytime is used. "0" tells the alarm thread to exit immediately if it fails to connnect to the Event Notification Server at startup.
ens.startlistener	"0"	Acceptable values: • "1" • "0"
caldb.berkeleydb.alarmretrytime	"300"	Retry time in seconds after a recoverable alarm delivery error.
caldb.berkeleydb.ensmsg.createcal	"no"	If "yes", create an Event Notification Service message when a calendar is created using the following format: enp://ics/createcal?calid=cal
caldb.berkeleydb.ensmsg.modifycal	"no"	If "yes", create an Event Notification Service message when a calendar is modified using the following format: enp://ics/modifycal?calid=cal
caldb.berkeleydb.ensmsg.deletecal	"no"	If "yes", create an Event Notification Service message when a calendar is deleted using the following format: enp://ics/deletecal?calid=cal

Table 12-15 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
caldb.berkeleydb.ensmsg.advancedtopics	"no"	Specifies how modify event notifications are published: • "yes"–Modify event notifications differentiate between reply, refresh, or modify transaction by publishing to the respective topic: caldb.berkeleydb.ensmsg.replyevent caldb.berkeleydb.ensmsg.refreshevent or caldb.berkeleydb.ensmsg.modifyevent • "no"–Modify event notifications are published to the caldb.berkeleydb.ensmsg.modifyevent topic, regardless of whether the transaction is a reply, refresh, or modify.
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.refreshevent	"no"	Specifies ("yes" or "no") whether Calendar Server should create an ENS message when an event is refreshed.
caldb.berkeleydb.ensmsg.refreshevent.contenttype	"text/xml"	Specifies the content type of the message data for the refresh of an event. Values can be "text/xml" or "text/calendar".
caldb.berkeleydb.ensmsg.refreshevent.url	"enp:///ics/caleventrefresh"	Specifies the URL for the ENS message for the refresh of an event.
caldb.berkeleydb.ensmsg.replyevent	"no"	Specifies ("yes" or"no") whether Calendar Server should create an ENS message for a reply to an event.
caldb.berkeleydb.ensmsg.replyevent.contenttype	"text/xml"	Specifies the content type of the message data for a reply to an event. Values can be "text/xml" or "text/calendar".
caldb.berkeleydb.ensmsg.replyevent.url	"enp:///ics/caleventreply"	Specifies the URL for the ENS message for a reply to an event.
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

Table 12-15 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

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

Table 12-15 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
caldb.serveralarms.maxretrytime	"-1"	How many times the alarm thread will consecutively retry and fail to contact the Event Notification Server. "-1" causes the alarm thread to retry indefinitely.
caldb.serveralarms.maxthreads	"10"	Maximum number of threads for the Event Notification Server.
caldb.serveralarms.retryinterval	"5"	Number (in seconds) that the alarm thread (in `csadmind`) waits before attempting to recontact the Event Notification Server.
caldb.serveralarms.stacksize	"65536"	Stack frame size for Event Notification Server threads.
caldb.serveralarms.startupretrytime	"0"	Total number of seconds 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 (UI) Configuration

The following table shows the user interface (UI) configuration parameters with each parameter's default value and description.

Table 12-16 User Interface Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
ui.config.file	" "	Calendar Server can read an optional (xml based) configuration file at startup that can hide parts of the user interface. Calendar Server allows only one configuration file, and the value of this setting determines the name of the file to use. Calendar Server looks for the file in the data directory where the user interface xml and xslt files are: *cal_svr_base*/opt/SUNWics5/cal/data 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.
ui.eventdialog.inform.enable	"no"	Specifies whether Calendar Express should display the "Inform About Event" option when users create or edit an event.

Table 12-16 User Interface Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
ui.toolbar.repainting.enable	"yes"	Enables ("yes") or disables ("no") repainting (or refreshing) of the Calendar Express tool bar. Setting to "no" can improve performance, because does not perform the XML and XSLT transformation for the tool bar. If set to "no", clicking refresh on any view takes the Calendar Express user back to the default view. If the browser cache option (next parameter) is enabled, the tool bar repainting option is not used.
browser.cache.enable	"no"	Enables ("yes") or disables ("no") the browser cache option. For more information, see "Using the Refresh View Option" on page 324.
render.xslonclient.enable	"yes"	Enables ("yes") or disables ("no") client-side rendering by downloading the XSLT processing to the end user's browser. For more information, see XSL Rendering in the Client Browser.

csmonitor Utility Configuration

The following table shows the `csmonitor` utility configuration parameters with each parameter's default value and description.

Table 12-17 csmonitor Utility Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
service.monitor.continuous	"0"	Specifies whether csmonitor should loop continuously: "0"–Do not loop continuously. "1"–Loop continuously. This option requires more system resources, but it can useful in debug mode.
service.monitor.loopsdelay	"60"	Specifies the delay in seconds between two monitoring loops.
service.monitor.emailaddress.from	none	Specifies the email address csmonitor sends messages from.
service.monitor.emailaddress.to	none	Specifies the email address csmonitor should send messages to.
service.monitor.csdb.logthreshold	"90"	Specifies a threshold value in percent of the total disk space for the maximum disk occupation considered to be normal. If the disk occupation where the calendar database (csdb directory) resides exceeds this value, csmonitor sends a warning email message.
logfile.monitor.logname	"csmonitor.log"	Specifies the csmonitor log file name.
logfile.monitor.maxlogfilesize	"2097152"	Specifies the maximum log file size. If the log file exceeds this size, csmonitor saves the log as csmonitor.log.*timestamp* and resets the log.
service.monitor.dbglevel	"0"	Specifies the debug level. The higher this value, csmonitor sends more precise and verbose messages.

Counters Configuration (counter.conf) File

Calendar Server counters (statistics) configuration parameters are in the following file:

cal_svr_base/etc/opt/SUNWics5/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 11, "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

The following table shows each alarm counter's name, type, size, and description.

Table 12-18 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

The following table shows each disk usage counter's name, type, size, and description.

Table 12-19 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

The following table shows each HTTP counter's name, type, size, and description.

Table 12-20 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 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.maxSessions	COUNTER	4	Maximum number of WCAP sessions served.
httpstat.numConnections	COUNTER	4	Total number of connections served.
httpstat.numCurrentConnections	GAUGE	4	Current number of active connections.
httpstat.numCurrentSessions	GAUGE	4	Current number of WCAP sessions.
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

The following table shows each Group Scheduling Engine (GSE) counter's name, type, size, and description.

Table 12-21 Group Scheduling Engine (GSE) 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

The following table shows each Authentication counter's name, type, size, and description.

Table 12-22 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

The following table shows each WCAP counter's name, type, size, and description.

Table 12-23 WCAP (wcapstat) Counters in the counter.conf File

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

Database Counters

The following table shows each Database counter's name, type, size, and description.

Table 12-24 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

The following table shows each Server Response counter's name, type, size, and description.

Table 12-25 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

The following table shows each Session Status counter's name, type, size, and description.

Table 12-26 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

Calendar Server sends the types of email messages described in Table 12-27. 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:

cal_svr_base`/etc/opt/SUNWics5/config`

For example, the English version of the task alarm message format is specified in the file:

cal_svr_base`/etc/opt/SUNWics5/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 Notifications
- Special Character Sequences for Dates
- Simple Event Reminder Example
- Complex Event Reminder Example

Calendar Server Mail Parameters

The following table shows the message type, parameter name, default format file description, and recipient for each Calendar Server Mail parameter.

Table 12-27 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

Table 12-27 Calendar Server Email Formats in the ics.conf File *(Continued)*

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

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 Notifications
- Special Character Sequences for Dates

Special Character Sequences for Events

The following table shows the format code and meaning for Special Character Sequences for Event Notifications.

Table 12-28 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

Table 12-28 Special Character Sequences for Event Notifications *(Continued)*

Format Code	Meaning
%L	Location
%l	Recurrence rules in iCalendar format
%M	Modify time
%N	New line
%n	The current time stamp used with `DTSTAMP`
%P	Priority
%r	Recurrence id (blank if this event does not recur)
%S	Event sequence number
%s	Summary
%t	Event status
%U	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 299.)

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 303). 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 Notifications

The following table shows the format code and meaning for Special Character Sequences for Task Notifications.

Table 12-29 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
%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

Table 12-29 Special Character Sequences for Task Notifications *(Continued)*

Format Code	Meaning
%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)

Special Character Sequences for Dates

The following table shows the format code and meaning for Special Character Sequences for dates.

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

Table 12-30 Special Character Sequences for Dates

Format Code	Meaning
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for locale
%d	Day of month as decimal number (01 - 31)
%H	Hour in 24-hour format (00 - 23)
%I	Hour in 12-hour format (01 - 12)
%j	Day of year as decimal number (001 - 366)

Table 12-30 Special Character Sequences for Dates *(Continued)*

Format Code	Meaning
%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
```

The description of each line in this example is:

- Line 1 is the message subject.
- 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.
- 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.
- Line 4 is the last MIME line, %x is the content transfer encoding string needed for this message.
- Line 5 lists the event summary and calls out the event summary with %s.
- 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.
- Line 7 lists the event end time.
- Line 8 lists the location of the event.
- 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. This example has a 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
- Solaris Operating System 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 systems. 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.

Solaris Operating System Tools

The following table shows the monitoring tools available for the Solaris Operating System.

Table A-2 Monitoring Tools for the Solaris Operating System

Tool	Description
`lockstat`	Provides information on OS and application locking. Available on Solaris 2.6 systems 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.

Appendix B

Calendar Server Time Zones

This appendix describes how Sun™ ONE Calendar Server defines and processes time zones, including:

- Overview of Calendar Server Time Zones
- Managing Calendar Server Time Zones
  - Adding a New Time Zone
  - Modifying an Existing Time Zone

For more information about time-zone properties and parameters, refer to the RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar):

`http://www.ietf.org/rfc/rfc2445.txt`

Overview of Calendar Server Time Zones

The timezones.ics file contains the representation of the time zones supported by Calendar Server. On Solaris Systems, the file is located in the following directory:

cal_svr_base/opt/SUNWics5/cal/data

At startup, Calendar Server reads the timezones.ics file, generates time-zone data, and then stores the data in memory. Thus, time-zone data is kept in memory while Calendar Server is running. Consequently, if you add a new time zone or modify an existing one, you must stop and restart Calendar Server for the change to take effect.

Time zones in the timezones.ics file are identified by the TZID parameter. For example, Calendar Server identifies the Pacific Standard Time (PST/PDT) zone using the America/Los_Angeles TZID, as shown in Figure B-1. The TZNAME property is an abbreviated representation of the time zone, such as PST (Pacific Standard Time) for the America/Los_Angeles time zone.

Time zones such as America/Los_Angeles that recognize daylight savings time (DST) contain two sub-components: STANDARD for stand time and DAYLIGHT for DST. The X-NSCP-TZCROSS list contains a series of dates that indicate when the time zone changes to and from DST (DAYLIGHT) and standard (STANDARD) time.

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 DST to standard or standard to DST change occurs. The Calendar Express user interface uses the dates in X-NSCP-TZCROSS to determine when to display a change in the time zone.

A WCAP command that includes the time zone ID (tzid) parameter should refer to a valid time zone defined in the timezones.ics file. Calendar Server then returns data using that time zone. If a WCAP command specifies an unrecognized time zone, Calendar Server returns data in the GMT time zone by default. For more information about WCAP, refer to the *Sun ONE Calendar Server 6.0 Programmer's Manual*.

Figure B-1 shows the America/Los_Angeles time-zone representation in the timezones.ics file.

Figure B-1 America / Los_Angeles Time-Zone Representation in the timezones.ics File

```
BEGIN:VTIMEZONE
TZID:America/Los_Angeles
BEGIN:STANDARD
DTSTART:19671025T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0700
TZOFFSETTO:-0800
TZNAME:PST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZOFFSETFROM:-0800
TZOFFSETTO:-0700
TZNAME:PDT
END:DAYLIGHT
X-NSCP-TZCROSS:19880403T100000Z;19881030T090000Z;19890402T100000Z;19891029T090000Z;
  19900401T100000Z;19901028T090000Z;19910407T100000Z;19911027T090000Z;
  19920405T100000Z;19921025T090000Z;19930404T100000Z;19931031T090000Z;
  19940403T100000Z;19941030T090000Z;19950402T100000Z;19951029T090000Z;
  19960407T100000Z;19961027T090000Z;19970406T100000Z;19971026T090000Z;
  19980405T100000Z;19981025T090000Z;19990404T100000Z;19991031T090000Z;
  20000402T100000Z;20001029T090000Z;20010401T100000Z;20011028T090000Z;
  20020407T100000Z;20021027T090000Z;20030406T100000Z;20031026T090000Z;
  20040404T100000Z;20041031T090000Z;20050403T100000Z;20051030T090000Z;
  20060402T100000Z;20061029T090000Z;20070401T100000Z;20071028T090000Z;
  20080406T100000Z;20081026T090000Z;20090405T100000Z;20091025T090000Z;
  20100404T100000Z;20101031T090000Z;20110403T100000Z;20111030T090000Z;
  20120401T100000Z;20121028T090000Z;20130407T100000Z;20131027T090000Z;
  20140406T100000Z;20141026T090000Z;20150405T100000Z;20151025T090000Z;
  20160403T100000Z;20161030T090000Z;20170402T100000Z;20171029T090000Z;
  20180401T100000Z;20181028T090000Z;20190407T100000Z;20191027T090000Z;
  20200405T100000Z;20201025T090000Z;20210404T100000Z;20211031T090000Z;
  20220403T100000Z;20221030T090000Z;20230402T100000Z;20231029T090000Z;
  20240407T100000Z;20241027T090000Z;20250406T100000Z;20251026T090000Z;
  20260405T100000Z;20261025T090000Z;20270404T100000Z;20271031T090000Z;
  20280402T100000Z;20281029T090000Z;20290401T100000Z;20291028T090000Z;
  20300407T100000Z;20301027T090000Z;20310406T100000Z;20311026T090000Z;
  20320404T100000Z;20321031T090000Z;20330403T100000Z;20331030T090000Z;
  20340402T100000Z;20341029T090000Z;20350401T100000Z;20351028T090000Z;
  20360406T100000Z;20361026T090000Z;20370405T100000Z;20371025T090000Z;
  20360406T120000Z;20361026T110000Z;20370405T120000Z;20371025T110000Z
END:VTIMEZONE
```

Managing Calendar Server Time Zones

This section describes these topics:

- Adding a New Time Zone
- Modifying an Existing Time Zone

Adding a New Time Zone

This section describes how to add a new time zone to Calendar Server, so that it is available in the Calendar Express user interface. For example, you might want to add a new time zone for America / Miami.

TIP The simplest way to add a new time zone is to copy and edit time-zone entries that are similar to the time zone you want to add in each of the files described in the following steps. For example, if you want to add a time zone for America / Miami, copy and edit the time-zone entries in each file for America / New_York.

To Add a New Time Zone

1. Add a time-zone block for the new time zone in the following file:

   cal_svr_base`/opt/SUNWics5/cal/data/timezones.ics`

   Again, the simplest way to add a new time-zone block is to copy an existing block that is similar, including any daylight savings time (DST) offsets, to the time zone you want to add. Then, edit the new time-zone block, making any changes for the new time zone. If your new time zone has Daylight Savings Time (DST), try to find a similar

2. Modify the `getDisplayNameofTZID` template in the following file:

   cal_svr_base`/opt/SUNWics5/cal/html/`*language*`/i18n.xsl file`

   where *language* specifies the directory for the language your site is using. For example: `en` for English, or `fr` for French.

   Add the new entry in the `i18n.xsl` file as:

   `<xsl:when test="$tzid='`*TimeZoneArea*`/`*TimeZoneName*`'"`
   TimeZoneArea`/`*TimeZoneName*`</xsl:when>`

where:

TimeZoneArea is one of the geographic divisions: Africa, America, Asia, Atlantic, Australia, Europe, or Pacific.

TimeZoneName is the name of your new time zone.

For example:

```
<xsl:when test="$tzid='America/Miami'">America/Miami</xsl:when>
```

3. Modify the following XML files:

cal_svr_base`/opt/SUNWics5/cal/html/change_timezone.xml`
cal_svr_base`/opt/SUNWics5/cal/html/new_cal.xml`
cal_svr_base`/opt/SUNWics5/cal/html/new_group.xml`

In each of these files, add the following line:

```
<timezone type="TimeZoneType" tzid="TimeZoneArea/TimeZoneName"
offset="offset"/>
```

where:

TimeZoneType is `"americas"`,`"europeAfrica"`, or `"asiaPacific"`.

TimeZoneArea and *TimeZoneName* are defined in Step 2.

offset is the number of hours that your new time zone is ahead (+) or behind (-) GMT. For example, if your new time zone is four hours behind GMT, the offset would be `"-04:00"`.

For example:

```
<timezone type="americas" tzid="America/Miami" offset="-05:00"
daylightOffset="-04:00"/>
```

4. If you want the new time zone to be the default time zone for user preferences, modify the “`icsTimeZone`” entry in the following file:

cal_svr_base`/opt/SUNWics5/cal/html/default_user_prefs.xml`

5. To include a reference to your new time zone in the Calendar Express online help, modify the following file:

cal_svr_base`/opt/SUNWics5/cal/html/`*language*`/chcncpt.html`

6. Stop (if necessary) and then restart Calendar Server for your new time zone to take effect.

Modifying an Existing Time Zone

This section describes how to modify an existing time zone. For example, you might want to change the name of a time zone, such as "America/Phoenix" to "US/Arizona".

To Modify an Existing Time Zone

1. Modify the time-zone block for the time zone you want to change in the following file:

   cal_svr_base/`opt/SUNWics5/cal/data/timezones.ics`

   If you are changing a time-zone name, change the TZID entry to the new name.

2. Modify the `getDisplayNameofTZID` template in the following file:

   cal_svr_base/`opt/SUNWics5/cal/html/`*language*/`i18n.xsl file`

   where: *language* specifies the directory for the language your site is using. For example: `en` for English or `fr` for French.

   If you are changing the name, change the existing time-zone name to the new name.

3. Modify the following XML files for changes to the time zone:

   cal_svr_base/`opt/SUNWics5/cal/html/change_timezone.xml`
   cal_svr_base/`opt/SUNWics5/cal/html/new_cal.xml`
   cal_svr_base/`opt/SUNWics5/cal/html/new_group.xml`

   For information about the entries in these files, see Step 2 and Step 3 in To Add a New Time Zone.

4. If the change affects the default time zone for user preferences, modify the "`icsTimeZone`" entry in the following file:

   cal_svr_base/`opt/SUNWics5/cal/html/default_user_prefs.xml`

5. If the change affects the Calendar Express online help, modify the following file:

   cal_svr_base/`opt/SUNWics5/cal/html/`*language*/`chcncpt.html`

6. Stop (if necessary) and then restart Calendar Server for your time-zone changes to take effect.

Appendix C

Calender Server Performance Tuning

To improve the performance of Sun ONE Calendar Server, consider the following options:

- Indexing the LDAP Directory Server
- Using Calendar Searches of the LDAP Directory Server
    - Indexing the icsCalendarOwned Attribute
    - Setting the nsSizeLimit and nsLookthroughLimit Parameters
- Using the LDAP Data Cache Option
- Using the CLD Cache Option
- Using a Memory-Based File System for the Session Database
- Using Load Balancing Across Multiple CPUs
- Setting the gse.belowthresholdtimeout Parameter
- Using the Refresh View Option
- Disabling the Calendar Express Tool Bar Repainting Option
- XSL Rendering in the Client Browser

Indexing the LDAP Directory Server

To improve performance when Calendar Server accesses the LDAP directory server, add indexes to the LDAP configuration file for the following attributes.

- `icsCalendar` attribute is used for searching for the default calendar for a calendar user or resource. Specify presence (pres), equality (eq), and substring (sub) index types.
- `icsCalendarOwned` is used for searching for a subscribe operation when the LDAP CLD plug-in is enabled. Specify presence (pres), equality (eq), and substring (sub) index types. See also "Using Calendar Searches of the LDAP Directory Server" on page 320.
- `mail` and `mailAlternateAddress` specify a user's primary and alternate email addresses. See also "Required LDAP mail Attribute for Calendar Server Users" on page 49 and "Email Alias (mailalternateaddress Attribute)" on page 51.

Note If you run the Directory Server Setup (`comm_dssetup.pl`) script to configure Sun ONE Directory Server 5.x, the script adds indexes for the `icsCalendar` and `icsCalendarOwned` attributes.

For information about adding directory server indexes, refer to the *Sun ONE Directory Server Configuration, Command, and File Reference* on the following website:

`http://docs.sun.com/db/coll/S1_ipDirectoryServer_51`

Using Calendar Searches of the LDAP Directory Server

Calendar searches of the LDAP directory server are enabled by the following parameter in the `ics.conf` file:

```
service.calendarsearch.ldap = "yes"
```

If you are using calendar searches of the LDAP directory, you can improve performance by:

- Indexing the icsCalendarOwned Attribute
- Setting the nsSizeLimit and nsLookthroughLimit Parameters

Indexing the icsCalendarOwned Attribute

To determine if the calendar search performance of the LDAP directory server can be improved, try the following LDAP command:

```
ldapsearch -b "base"
"(&(icscalendarowned=*user*)(objectclass=icsCalendarUser))"
```

where *base* is the LDAP base DN of the directory server where the user and resource data for Calendar Server is located, and *user* is the value that an end user can enter in the Calendar Express Subscribe->Calendar Search dialog.

Tests have shown that with 60,000 entries, the above search took about 50-55 seconds without indexing `icsCalendarOwned`. After indexing, the above search took only about 1-2 seconds.

On Sun ONE Directory Server, index the `icsCalendarOwned` attribute using the following command on Solaris Systems:

```
server5/bin/slapd db2index -D slapd-serverID
-t icsCalendarOwned: eq,pres,sub:2.16.840.1.113730.3.3.2.11.1
```

where *slapd-serverID* is the full path to the `slapd-serverID` directory.

Setting the nsSizeLimit and nsLookthroughLimit Parameters

The `nsSizeLimit` and `nsLookthroughLimit` parameters in your LDAP directory server configuration must be large enough so that searches complete properly.

To determine if these parameters are set to appropriate values, try the following command:

```
ldapsearch -b "base"
"(&(icscalendarowned=*user*)(objectclass=icsCalendarUser))"
```

where *base* is the LDAP base DN of the directory server where the user and resource data for Calendar Server is located, and *user* is the value that an end user can enter in the Calendar Express Subscribe->Calendar Search dialog.

If the LDAP server returns an error, the `nsSizeLimit` or the `nsLookthroughLimit` parameter might not be large enough. Follow these guidelines to set these parameters:

- Ensure that the value for the nsSizeLimit parameter is large enough to return all the desired results; otherwise, truncation can occur, and no results will be displayed.
- Ensure that the value for the nsLookthroughLimit parameter is large enough to complete a search of all the users and resources in the LDAP directory. If possible set nsLookthroughLimit to -1, which causes no limit to be used.

Using the LDAP Data Cache Option

The LDAP data cache option ensures that LDAP data is available immediately after it has been committed, even there is a delay in the availability of committed data.

For example, if your site has deployed a master/slave LDAP configuration where Calendar Server accesses the master LDAP directory through a slave LDAP directory server, which in turn introduces a delay in the availability of committed LDAP data, the LDAP data cache can ensure that your Calendar Server clients have accurate LDAP data.

For information, refer to Appendix D, "Using the LDAP Data Cache."

Using the CLD Cache Option

If you are using the with the LDAP CLD plug-in, make sure the following configuration parameters in the ics.conf are set to "yes" (which is the default value for each parameter):

```
caldb.cld.cache.enable = "yes"
```

caldb.cld.cache.enable enables the CLD cache option. This cache stores the DWP host server information (icsDWPHost LDAP attribute) for calendar users and thus reduces calls to the LDAP directory server.

```
service.calendarsearch.ldap = "yes"
```

service.calendarsearch.ldap specifies that calendar searches are performed using the LDAP CLD plug-in or a user preference plug-in.

Using a Memory-Based File System for the Session Database

To improve performance on Solaris systems, you can configure a memory-based file system (`tmpfs`) for the session database by setting the following parameter in the `ics.conf` file:

```
local.instance.use.tmpfs to "true"
```

The `tmpfs` file system is then overlaid based on the values of the `service.http.sessiondir.path` and `service.admin.sessiondir.path` parameters.

For more information, see the `tmpfs(7FS)` and `mount_tmpfs(1M)` man pages in the Solaris documentation:

http://docs.sun.com/db/prod/solaris

Using Load Balancing Across Multiple CPUs

If a server has multiple CPUs, by default Calendar Server distributes the HTTP Service (`cshttpd` processes) and Distributed Database Service (`csdwpd` processes) across the CPUs.

The `service.http.numprocesses` and `service.dwp.numprocesses` parameters determine the actual number of processes that run for each service. By default, these parameters are set to the number of CPUs for the server during installation, but you can reset these values. For example, if a server has 8 CPUs, but you want a `cshttpd` and `csdwpd` process to run in only 4 CPUs, set the parameters as:

```
service.http.numprocesses="4"
service.dwp.numprocesses="4"
```

To disable load balancing, add the `service.loadbalancing` parameter to the `ics.conf` file and set it to “no”. Then restart Calendar Server for the change to take effect.

Setting the gse.belowthresholdtimeout Parameter

The following parameter in the ics.conf file specifies the time in seconds to wait before Calendar Server scans the Group Scheduling Engine (GSE) queue for incoming jobs:

```
gse.belowthresholdtimeout = "3"
```

If there are more jobs in the queue than the maximum threads allocated, the last thread always scan the queue again. Therefore, this setting only takes effect only when the number of jobs is below the maximum threads allocated.

The default is “3”. Increasing this number reduces the frequency the server scans the queue and can improve overall performance.

Using the Refresh View Option

For Calendar Express end users, the Refresh View option improves performance by using calendar data in the browser cache to refresh a view rather than requiring an update from the Calendar Server database.

To enable the Refresh View option, the following parameter in the ics.conf file must be set to “yes”:

```
browser.cache.enable = "yes"
```

If you reset this parameter, you must stop and then restart Calendar Server for the new value to take effect.

When the Refresh View option is configured for a site, Calendar Express displays Refresh View on all calendar views on the View tab.

When a user clicks Refresh View, Calendar Express first checks whether the calendar data in the view has changed before requesting an update from the calendar database. If the data has not changed, Calendar Express uses information in the browser cache to refresh the view. Unnecessary requests to the calendar database are avoided, which is especially useful if a calendar has a large number of events or tasks.

If an event or task has changed, Calendar Express requests an update from the calendar database to refresh the view. Thus, users can also use Refresh View to ensure that Calendar Express always displays the latest calendar data.

Disabling the Calendar Express Tool Bar Repainting Option

The tool bar repainting option causes a Calendar Express view to be repainted (refreshed) when a user clicks refresh. Sometimes, however, this option can cause performance problems because Calendar Server refreshes a view by performing XML and XSLT transformation for the tool bar.

To disable the tool bar repainting option, set the following parameter in the `ics.conf` file to “no”:

```
ui.toolbar.repainting.enable="no"
```

If `ui.toolbar.repainting.enable` is set to “no”, clicking refresh on any view takes the Calendar Express user back to the default view.

Setting `ui.toolbar.repainting.enable` to “no” can improve performance, because Calendar Express does not perform the XML and XSLT transformation for the tool bar.

If the browser cache option (`browser.cache.enable` parameter) is set to “yes”, the tool bar repainting option is not used.

XSL Rendering in the Client Browser

Calendar Server performs client-side rendering by downloading the XSLT processing to the end user’s browser, which in turn reduces the processing that must be done by Calendar Server. Calendar Server downloads the XSLT processing only if the browser is capable of rendering the XSLT processing. In the current release, this applies only to Internet Explorer 6.0.

Tests have shown that client-side rendering can improve the interface (UI) scalability by 4 to 6 times, which means that Calendar Server can support 4 to 6 times as many concurrent end users without maxing out a CPU.

The following parameter in the `ics.conf` file controls client-side rendering (currently for Internet Explorer 6.0 or later only):

```
render.xslonclient.enable="yes"
```

By default this parameter is set to “yes”. To turn off client-side rendering, set the parameter to “no” and then restart Calender Server.

Appendix D

Using the LDAP Data Cache

This appendix describes the Sun ONE Calendar Server LDAP data cache, which ensures that LDAP data is available immediately after it has been committed, even if the LDAP directory server is configured to include a delay in the availability of committed data. Topics in this appendix include:

Considerations for Using the LDAP Data Cache

Use these guidelines to determine if your site should configure the LDAP data cache:

- If Calendar Server at your site accesses your master (or root) LDAP directory server directly with no delays in the availability of committed LDAP data, you don't need to configure the LDAP data cache. Make sure that the `local.ldap.cache.enable` parameter is set to "no" (which is the default).
- If your site has deployed a Master/Slave LDAP Configuration where Calendar Server accesses the master LDAP directory through a slave LDAP directory server, which in turn introduces a delay in the availability of committed LDAP data, configure the LDAP data cache to ensure that your end users have the most recent data.

Master/Slave LDAP Configuration

A Master/Slave LDAP configuration includes a master (root) directory server and one or more slave (consumer or replica) directory servers. Calendar Server can access the master LDAP directory server either directly or through a slave directory server:

- If Calendar Server accesses the master LDAP directory server directly, the LDAP should be accurate, and you don't need to configure the LDAP data cache.
- If Calendar Server accesses the master LDAP directory server through a slave directory server, LDAP data changes are usually written transparently via an LDAP referral to the master directory server, which in turn replicates the data back to each slave directory server.

In this second type of configuration, problems with inaccurate LDAP data can occur because of the delay in the availability of committed LDAP data to the slave directory servers.

For example, Calendar Server commits an LDAP data change, but the new data is not available for a specific amount of time because of the delay in the master directory server updating each slave directory server. A subsequent Calendar Server client operation uses the old LDAP data and presents an out-of-date view.

If the delay in updating the slave directory servers is short (only a few seconds), clients might not experience a problem. However, if the delay is longer (minutes or hours), clients will display inaccurate LDAP data for the length of the delay.

Table D-1 lists the LDAP attributes that are affected by a delay in a master/slave LDAP server configuration where Calendar Server accesses the master LDAP directory server through a slave LDAP directory server.

Table D-1 Calendar Server LDAP Attributes Affected by Delays

Operation	LDAP Attributes Affected
Auto provisioning	icsCalendar, icsSubscribed, icsCalendarOwned, icsDWPHost
Calendar groups	icsSet
Calendar creation	icsCalendarOwned, icsSubscribed
Calendar subscription	icsSubscribed
User options	icsExtendedUserPrefs, icsFirstDay, icsTimeZone, icsFreeBusy
Calendar searches	icsCalendarOwned

To insure that your end uses have the most recent LDAP data, configure the LDAP data cache as described in the following sections: LDAP Data Cache and LDAP Data Cache Configuration Parameters.

LDAP Data Cache

The LDAP data cache resolves the master / slave LDAP configuration problem by providing Calendar Server clients with the most recent LDAP data, even when the master directory server has not updated each slave directory server.

If the LDAP data cache is enabled, Calendar Server writes committed LDAP data to the cache database (`ldapcache.db` file). By default, the LDAP cache database is located in the *cal_svr_base*`/var/opt/SUNWics5/csdb/ldap_cache` directory, but you can configure a different location if you prefer.

When a client makes a change to the LDAP data for a single user, Calendar Server writes the revised data to the LDAP cache database (as well as to the slave directory server). A subsequent client operation retrieves the LDAP data from the cache database. This data retrieval applies to the following operations for a single user:

- User’s attributes at login
- User’s options (such as color scheme or time zone)
- User’s calendar groups
- User’s subscribed list of calendars

Thus, the LDAP data cache database provides for:

- Data consistency across processes on a single system–The database is available to all Calendar Server processes on a multi-processor system.
- Data persistence across user sessions–The database is permanent and does not require refreshing. You can configure the time to live (TTL) for an LDAP data cache entry and the interval between each database cleanup. See LDAP Data Cache Configuration Parameters for more information.

Limitations

The LDAP data cache does not provide for:

- Reading the cache for searches where a list of entries is expected. For example, searching for attendees for a meeting. This type of search is subject to any LDAP delay. For instance, a newly created calendar will not appear in a calendar search if the LDAP search option is active and the search is performed within the delay period following the creation of a new calendar.
- Reading and writing of the cache across multiple front-end servers. Each front-end server has its own cache, which is not aware of data in other caches.
- The capability to handle a user who doesn't always log into the same server. Such a user will generate different LDAP data in the cache on each server.

LDAP Data Cache Configuration Parameters

Table D-2 describes the configuration parameters in the `ics.conf` file for the LDAP data cache.

Table D-2 LDAP Data Cache Configuration Parameters

Parameter	Description
local.ldap.cache.enable	Enables ("yes") or disables ("no") the LDAP data cache. The default is "no".
local.ldap.cache.checkpointinterval	Specifies the number of seconds for the checkpoint thread to sleep. The default time is 60 seconds.
local.ldap.cache.circularlogging	Specifies whether or not to remove the database log files after they have been processed. The default is "yes".
local.ldap.cache.homedir.path	Specifies the physical location of LDAP data cache database. The default is *cal_svr_base*/var/opt/SUNWics5/csdb/ldap_cache.
local.ldap.cache.logfilesizemb	Specifies the maximum size in megabytes of the checkpoint file. The default is 10 megabytes.
local.ldap.cache.maxthreads	Specifies the maximum number of threads for the LDAP data cache database. The default is "1000".
local.ldap.cache.mempoolsizemb	Specifies the number of megabytes of shared memory. The default is 4 megabytes.
local.ldap.cache.entryttl	Specifies the time to live (TTL) in seconds for an LDAP data cache entry. The default is 3600 seconds (1 hour).
local.ldap.cache.stat.enable	Specifies whether or not to log the access to the LDAP data cache and to print statistics in the log file. The default is "no". **Note** This parameter applies only to debug mode.
local.ldap.cache.stat.interval	Specifies the interval in seconds when each statistics report is written to the log file. The default is 1800 seconds (30 minutes).
local.ldap.cache.cleanup.interval	Specifies the interval in seconds between each database cleanup. The default is 1800 seconds (30 minutes).

CAUTION If Calendar Server or the server where Calendar Server is running is not properly shut down, it is recommended that you manually delete all files in the `ldap_cache` directory to avoid any database corruption that might cause problems during a subsequent restart.

Appendix E

High Availability (HA) Configuration Worksheets

To plan a high availability (HA) configuration for Sun ONE Calendar Server, use the following worksheets:

- Calendar Server HA Configuration Worksheet
- Calendar Server Installation Worksheet
- Calendar Server Configuration Worksheet

TIP If you are viewing the HTML version of these worksheets and you want to print them, consider printing the PDF version for better control of page breaks.

Calendar Server HA Configuration Worksheet

Table E-1 Sun ONE Calendar Serve HA Configuration Worksheet

Component	Value and Comments
File-system mount point	For example: /global/cal Your value: ________________________________
Logical Host Name	For example: cal-logical-host Your value: ________________________________
Logical Host Name IP Address	Your value: ________________________________
Nodes in the Cluster	For example: cal-node-1 and cal-node-2 Your value: ________________________________
Calendar Resource Group Name	For example: cal-resource-group Your value: ________________________________
Calendar Server Storage Resource	For example: cal-resource-group-store Your value: ________________________________
Calendar Server Resource	For example: cal-resource Your value: ________________________________

Calendar Server Installation Worksheet

Table E-2 lists the values you set when you install Calendar Server using the Sun Java Enterprise System installer.

Table E-2 Sun ONE Calendar Server Installation Worksheet

Component	Description and Comments
Calendar Server Installation Directory	Default value: opt/ For example: global/cal/opt/r Your value: ______________________________

Calendar Server Configuration Worksheet

Table E-3 lists the values you set when you run the Calendar Server configuration program (`csconfigurator.sh`).

Table E-3 Sun ONE Calendar Server Configuration Worksheet

Component	Description and Comments
LDAP Server Host Name	For example: `ldaphost.sesta.com` Your value: ______________________________
LDAP Server Port	Port number that the LDAP server listens on. Default: 389. Your value: ______________________________
Directory Manager DN	User name that can make changes in the directory server schema. Default: `cn=Directory Manager`. Your value: ______________________________
Directory Manager Password	Password of the Directory Manager DN. Default: None Your value: ______________________________

Table E-3 Sun ONE Calendar Server Configuration Worksheet *(Continued)*

Component	Description and Comments
Administrator User ID	User ID of the Calendar Server administrator. This user must be a user in the above LDAP directory server. Default: `calmaster`. Your value: ______________________________
Administrator Password	Password of the Calendar Server administrator. Default: None Your value: ______________________________
Email Alarms	Specifies whether Calendar Server should send an email alarm message to a Calendar Server administrator in case a server problem occurs. Default: Enabled. Your value: ______________________________
Administrator Email Address	Email address of the Calendar Server administrator who will receive the email alarm messages. Default: None. Your value: ______________________________
SMTP Host Name	Host name of the SMTP server where email alarm messages should be sent. Default: Current host. Your value: ______________________________
Service Port	Port number that Calendar Server listens on to provide Web (HTTP) access to users. Default: 80. Your value: ______________________________
Maximum Sessions	Maximum number of Calendar Server sessions. Default: 5000. Your value: ______________________________
Maximum Threads	Maximum number of Calendar Server threads. Default: 20. Your value: ______________________________
Number of Server Processes	Maximum number of Calendar Server processes. Default: Number of CPUs on the server where you are installing Calendar Server. Your value: ______________________________

Table E-3 Sun ONE Calendar Server Configuration Worksheet *(Continued)*

Component	Description and Comments
Runtime User ID	Default value: icsuser For an HA configuration, add to /etc/passwd on all nodes in the cluster. Your value: ______________________________
Runtime Group ID	Default value: icsgroup For an HA configuration, add to /etc/group on all nodes in the cluster. Your value: ______________________________
Calendar Server Startup	Start after successful installation. Default: Checked. Your value: For an HA configuration, do **not** check this option. Start on system startup. Default: Checked. Your value: For an HA configuration, do **not** check this option.
Database Directory	Default: var/opt/SUNWics5/csdb For example: /global/cal/var/opt/SUNWics5/csdb Your value: ______________________________
Logs Directory	Default: var/opt/SUNWics5/logs For example: /global/cal/var/opt/SUNWics5/logs Your value: ______________________________
Temporary Files Directory	Default: var/opt/SUNWics5/tmp For example: /global/cal/var/opt/SUNWics5/tmp Your value: ______________________________

Appendix F

Using Sun ONE Instant Messaging Pop-up Reminders

Sun ONE Calendar Server 5.1.1 (or later) is integrated with Sun ONE Instant Messaging 6.0 (or later) to provide automatic pop-up reminders for both calendar events and tasks on the Instant Messenger desktop.

This appendix describes:

- Pop-up Reminders Operation
- Configuring Instant Messaging Pop-ups
    - Instant Messaging Server Configuration
    - Calendar Server Configuration
    - Instant Messenger Configuration

Pop-up Reminders Operation

Pop-up reminders operate based on email reminders. An alarm causes Calendar Server to send an email notification and Instant Messaging to display a pop-up reminder, depending on the options are configured by both the end user and the Calendar Server administrator:

- End users configure email reminders in Calendar Express (as described in the Calendar Express online Help) and pop-up reminders in Instant Messenger, as described in "Instant Messenger Configuration" on page 342.

- A Calendar Server administrator can choose to configure both email and pop-up reminders for end users or either option. For example, to turn email reminders off, set the following parameter in the `ics.conf` file:

  ```
  caldb.serveralarms.binary.enable= "no"
  ```

Pop-up Reminders Architectural Flow

If configured, Instant Messaging pop-up reminders follow this architectural flow:

1. The Instant Messaging JMS subscriber subscribes to Calendar Server events and notifications in the Event Notification Service (ENS).
2. Calendar Server publishes an event or notification in `text/xml` or `text/calendar` format to ENS.
3. The Instant Messaging JMS subscriber receives the calendar event or notification and then generates a message in `text/calendar` format.
4. The Instant Messaging server sends the message to the calendar owner, if the end user is online.
5. Instant Messenger generates an HTML pop-up reminder on the end user's desktop based on the message.

Configuring Instant Messaging Pop-ups

This section includes the following configuration examples:

- Instant Messaging Server Configuration
- Calendar Server Configuration
- Instant Messenger Configuration

In these examples, Instant Messenger users will receive pop-up reminders for both Calendar Server events and tasks. The server-side installation is as follows:

- Calendar Server 5.1.1 (or later) is installed on `cal.example.com`.
- Instant Messaging 6.0 (or later) server is installed on `im.example.com`.

You will need to edit the configuration parameters for the servers used at your specific site.

Instant Messaging Server Configuration

To configure Instant Messaging server:

1. Add the following parameters to the `iim.conf` file:

```
! JMS Consumers
jms.consumers=cal_reminder
jms.consumer.cal_reminder.destination=enp:///ics/customalarm
jms.consumer.cal_reminder.provider=ens
jms.consumer.cal_reminder.type=topic
jms.consumer.cal_reminder.param="eventtype=calendar.alarm"
jms.consumer.cal_reminder.factory=com.iplanet.im.server.JMSCalendarMessageListener

! JMS providers
jms.providers=ens
jms.provider.ens.broker=cal.example.com:7997
jms.provider.ens.factory=com.iplanet.ens.jms.EnsTopicConnFactory
```

2. Restart the Instant Messaging server for the configuration changes to take effect:

```
cd /opt/SUNWiim/sbin/
imadmin refresh
```

Calendar Server Configuration

To configure Calendar Server, perform these steps:

1. Set (or add) the following parameters in the `ics.conf` file:

```
caldb.serveralarms = "yes"
caldb.serveralarms.contenttype = "text/xml"
caldb.serveralarms.dispatch = "yes"
caldb.serveralarms.dispatchtype = "ens"
caldb.serveralarms.url = "enp:///ics/customalarm"
```

2. Restart Calendar Server for the configuration changes to take effect:

```
cd /opt/SUNWics5/cal/sbin/
stop-cal
start-cal
```

Instant Messenger Configuration

To receive pop-up reminders for both Calendar Server events and tasks, users must configure their Instant Messenger as follows:

1. On the **`Main`** window, select **`Settings`** from the **`Tools`** menu, or click the **`Settings`** icon.
2. On the **`Messenger Settings`** window, click the **`Alerts`** tab.
3. Check the **Show calendar reminder alerts** option.
4. Click **`OK`**.

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. 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. 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 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 (CLD) A plug-in that determines the physical location of a calendar when the calendar database is distributed over two or more back-end servers. Calendar Server provides the LDAP CLD plug-in.

Calendar Server Application Programming Interface (CSAPI) A programmatic interface that provides the capability to modify or enhance the feature set of Calendar Server. CSAPI modules are plug-ins that are loaded from the `cal/bin/plugins` directory when 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 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. Calendar Server requires that a calendar user be stored in a directory server such as an LDAP server. 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. 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. 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. 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 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 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. 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. 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. Calendar Server uses the Sun ONE 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, 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. 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. 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. 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 Calendar Server.

Zulu time A military designation for GMT and UTC (Coordinated Universal Time).

Index