Sun Java System Calendar Server 6.3 Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–4654–10
March, 2007

080625@20490

Contents

Tables

Examples

Preface

This guide explains how to administer Sun Java™ System Calendar Server 6.3 product version. Calendar Server provides a scalable, Web-based solution for centralized calendaring and scheduling for enterprises and service providers. Calendar Server supports personal calendars as well as group and resource scheduling.

Topics in this chapter include:

Who Should Use This Book

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

Before You Read This Book

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

- Basic administrative procedures for your platform operating system.
- Lightweight Directory Access Protocol (LDAP) — if you plan to use an LDAP directory server to store user information.

How This Book Is Organized

Part Title	Description
Part IPart 1, Overview	Provides a high-level overview of Calendar Server, including the components, architecture, interfaces, and protocols.
Part IIPart 2, Post Installation Configuration	Provides instructions for running the `csconfigurator.sh` program, and instructions for using the post installation database migration utilities such as `csmig`, `csvdmig`, `csmigrate`, and `commdirmig`. **Note –** The Directory Preparation Tool (comm_dssetup.pl) chapter has been moved to the *Sun Java System Communications Suite Installation and Configuration Guide.*
Part IIIPart 3, Customizing Your Calendar Server Configuration	Provides instructions on customizing various aspects of Calendar Server. It also describes how to configure CLD plug-in, how to set up a High Availability environment, set up and manage SSL, configure single sign-on through either Access Manager authentication, or through Messaging Server (circle of trust), configure `csstored` to take automatic backups, configure Calendar Server with multiple domains, and configure Calendar Server with multiple domains.
Part IVPart 4, Calendar Server Administration	Describes the general Calendar Server tasks such as starting and stopping services. It also explains how to create, modify, delete and list domains for a multiple domain environment, administer user and resource LDAP entries, administer Calendars, including access control, administer and maintain the Calendar Server databases and data, back up and restore Calendar Server data, manage the Delete Log database (`ics50deletelog.db`).
Part VPart 5, Appendixes	This part contains a worksheet for gathering the required information you will need when running the Calendar Server configuration script, `csconfiguator.sh`. The last two appendices provide reference for the Calendar Server command-line utilities and `ics.conf` parameters.

Related Books for Calendar Server Version 6.3

The following Calendar Server documents are available online in PDF and HTML formats:

- *Sun Java Communications Suite 5 Release Notes*
- *Sun Java Communications Suite 5 Upgrade Guide*
- *Sun Java Communications Suite 5 Documentation Center*
- *Sun Java Communications Suite 5 Deployment Planning Guide*
- *Sun Java System Calendar Server 6.3 Administration Guide* (this document)
- *Sun Java System Calendar Server 6.3 WCAP Developer's Guide*
- *Sun Java Communications Suite 5 Event Notification Service Guide*
- *Sun Java System Communications Services 6 2005Q4 Schema Reference*
- *Sun Java Communications Suite 5 Schema Migration Guide*
- *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*
- *Sun Java System Communications Express 6.3 Administration Guide*
- *Sun Java System Communications Express 6.3 Customization Guide*
- *Sun Java Enterprise System Technical Note: Sun Java System Calendar Frequently Asked Questions*
- *Sun Java Enterprise System Glossary*

In addition, the graphical user interfaces, Communications Express and Delegated Administration Console, have online help.

Calendar Server Version 6.3 Related Third-Party Web Site References

Third-party URL's 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 or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training for Calendar Server Version 6.3

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell	**`machine_name%`**
C shell for superuser	**`machine_name#`**
Bourne shell and Korn shell	**`$`**
Bourne shell and Korn shell for superuser	**`#`**

PART I

Overview of Calendar Server 6.3 Software

This part contains only the Overview chapter.

CHAPTER 1

Overview of Calendar Server 6.3 Software

Sun Java™ System Calendar Server 6.3 (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.

For information about basic configuration scenarios, see the *Sun Java Communications Suite 5 Deployment Planning Guide*.

This chapter covers the following topics:

Note – In this and subsequent chapters, when fully qualified directory paths are specified, they are for the Solaris platform. The default paths for Solaris are:

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

```
/var/opt/SUNWics5
```

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

The default paths for Linux® are:

```
/opt/sun/calendar
```

```
/var/opt/sun/
```

```
/etc/opt/sun
```

Linux users should substitute their default path in any command showing the Solaris default.

1.1 Calendar Server 6.3 Software Installation

The installation of Calendar Server 6.3 has significantly changed from earlier Calendar Server releases. You must use the Communications Suite installer to install Calendar Server 6.3 software. Do not use the Java Enterprise System installer. However, you may need to use the Java Enterprise System Installer to install other server products.

For more information about installing Calendar Server 6.3, see .*Sun Java Communications Suite 5 Installation Guide*

If you want to upgrade from an earlier version of Calendar Server, the upgrade process is described in the *Sun Java Communications Suite 5 Upgrade Guide*.

For information about migrating your calendar databases and LDAP database from older versions of Calendar Server to version 6.3, refer to the information found in Chapter 3, “Database Migration Utilities for Calendar Server 6.3.”

1.2 Post Installation Configuration for Calendar Server Version 6.3

After you install Calendar Server, you must configure it. The installer does not perform configuration as part of the installation process.

▼ High Level Task List for Post Installation Configuration for Calendar Server Version 6.3

1. **Run the Directory Server Setup script, `comm_dssetup.pl`, to configure Sun Java System Directory Server 5 (if the script has not already been run).**

   This script is located in the following directory: `/opt/SUNWcomds/sbin`.

   For information about running it, see the *Sun Java Communications Suite 5 Installation Guide*.

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 Appendix E, "Calendar Server Configuration Parameters."

   The configuration program is located in the following directory: `/opt/SUNWics5/sbin`

   For information about running `csconfigurator.sh`, see Chapter 2, "Initial Runtime Configuration Program for Calendar Server 6.3 software (csconfigurator.sh)."

3. **Customize your system by editing parameters in the `ics.conf` file.**

   The chapters in Part III describe how to customize your system by editing the `ics.conf` file.

Note – Its possible for the `ics.conf` to contain duplicate parameters with different values. The system reads the file sequentially, updating system settings as it goes along. With this method, the last value it finds for this parameter is the one that gets used.

As a best practice, add all of your `ics.conf` settings to the end of the file so you will know which ones you have set. But to improve efficiency, comment out the older instances of the parameter. This helps because the fewer parameters the system has to read, the faster it can process the file.

1.3 Special Accounts for Calendar Server Version 6.3

Calendar Server special accounts include the following:

- "1.3.1 Calendar Server Administrator (calmaster) Account in Calendar Server Version 6.3" on page 40
- "1.3.2 Calendar Server User and Group Accounts for Calendar Server Version 6.3" on page 41
- "1.3.3 Superuser (root)" on page 41
- "1.3.4 Non-root User (icsuser, icsgroup) for Calendar Server Version 6.3" on page 41

1.3.1 Calendar Server Administrator (calmaster) Account in Calendar Server Version 6.3

The Calendar Server administrator is a specific user name with its 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.siteadmin.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.

The following table describes the Calendar Server administrator configuration parameters in the `ics.conf` file.

TABLE 1–1 Calendar Server Administrator (calmaster) Configuration Parameters

Parameter	Description
service.siteadmin.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.siteadmin.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*.
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.siteadmin.userid*. The default is *yes*.

1.3.2 Calendar Server User and Group Accounts for Calendar Server Version 6.3

These special accounts are the user ID and group ID under which Calendar Server runs. Unless there are overriding reasons not to, 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.

1.3.3 Superuser (root)

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.

1.3.4 Non-root User (icsuser, icsgroup) for Calendar Server Version 6.3

Although you need *root* privileges to install Calendar Server, it is possible to run the services as a non-root user.

However, if you start the services as *root*, each process changes the effective UID to the runtime (non-root) user and group once the tasks that need the *root* privileges have been executed. Doing it this way allows the use of ports below 1024. Instead, when you start services as the non-root runtime user and group, the web server port must be set to a value greater than 1024 in order for the services to start successfully.

Note – The non-root user or group are created automatically at the time of configuration. The defaults are `icsuser`, and `icsgroup`.

1.4 Proxy Administrator Logins for Calendar Server Version 6.3

To allow administrators to administer user calendars, the following parameter in the `ics.conf` file is set by default as shown: *service.http.allowadminproxy="yes"*.

If you are using Communications Express, this parameter must be set to *"yes"*.

For more information on this parameter and on verifying that proxy logins are working, see "4.5 Configuring Logins and Authentication" on page 132.

1.5 End User Administration in Calendar Server Version 6.3

End users can connect to Calendar Server from client machines using a Web graphical user interface (GUI), Sun Java System Communications Express, or through the Connector for Microsoft Outlook, which allows end users to continue using Outlook on their desktop while still taking advantage of the Calendar Server back end. Users must have a unique entry in the LDAP directory. Each user can have one or more calendars and can belong to one or more groups.

Administrators, with the proper permissions, can add, delete or modify user LDAP entries, or resource LDAP entries, using the Delegated Administrator Utility (command-line) or Console (GUI).

For documentation on the Delegated Administrator Utility (*commadmin*), see *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

For documentation on the Delegated Administrator Console, see the Console's online help.

In addition, when necessary, you can use *ldapmodify* to modify LDAP entries directly. For information about *ldapmodify*, refer to the *Sun ONE Directory Server Resource Kit 5.2 Tools Reference*.

Caution – Utility programs used in pre-Java Enterprise System deployments, such as *csuser*, are still bundled with Calendar Server. If you are using Access Manager in your deployment, do not use these utilities for managing or creating user, domain or resource LDAP entries. There are some exceptions. Where these apply, this guide will direct you to the proper utility.

This section describes the following aspects of user and user calendar administration:

- "1.5.1 Choosing the Proper User Management Tool for Calendar Server Version 6.3" on page 43
- "1.5.2 Creating User LDAP Entries in Calendar Server Version 6.3" on page 44

- "1.5.3 Authenticating Users in Calendar Server Version 6.3" on page 44
- "1.5.4 Understanding User Preferences for Calendar Server Version 6.3" on page 44
- "1.5.7 Group Calendars Overview for Calendar Server Version 6.3" on page 46
- "1.5.6 Autoprovisioning: Automatic Creation of Calendars in Calendar Server Version 6.3" on page 45
- "1.5.8 Resources Overview for Calendar Server Version 6.3" on page 46

1.5.1 Choosing the Proper User Management Tool for Calendar Server Version 6.3

Calendar users, groups, and resources can be administered using one of the following user management tools:

- Delegated Administrator Console.

  Use this GUI to provision users, groups and resources in LDAP for Calendar Server. For information on using the GUI, see the Delegated Administrator Console online help.
- Delegated Administrator Utility (*commadmin*).

  Use these tools to provision users, groups and resources in LDAP for Calendar Server . For detailed instructions, see the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.
- Calendar Server utilities.

  Use these utilities to manage calendars. In addition, use them for user, group, and resource management if your configuration meets all of the following criteria:
  - You are not using Access Manager.
  - You have an earlier version of Calendar Server or Messaging Server installed using Sun LDAP Schema version 1.
  - You plan to continue using Schema version 1.

  See also the command-line utility reference in this guide, Appendix D, "Calendar Server Command-Line Utilities Reference."

Note – Delegated Administrator does not manage calendars. To create calendars for users, groups and resources, use the Calendar Server utilities *cscal* and *csresource*, or turn on autoprovisioning. With autoprovisioning turned on, the system creates a default calendar under two circumstances: if a user logs in without a default calendar, or a user, group or resource is issued an invitation before the default calendar exists.

1.5.2 Creating User LDAP Entries in Calendar Server Version 6.3

You can create users in LDAP using the following tools:

- For Schema version 1, create both the user and the calendar at the same time using the Calendar Server *csuser* utility.
- For Schema version 2, use Delegated Administrator Console to create the user with the Create New User wizard. Then create the users default calendar using the Calendar Server Utility *cscal*. See Appendix D, "Calendar Server Command-Line Utilities Reference."
- For Schema version 2, use Delegated Administrator Utility, *commadmin user create*. Then use the Calendar Server Utility *cscal*.

  For further instruction on adding users in this guide, see "14.1 Creating Calendar User LDAP Entries" on page 269.

  For information on using the Delegated Administrator Utility, see *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

1.5.3 Authenticating Users in Calendar Server Version 6.3

Calendar Server requires a LDAP directory server such Sun Java System Directory Server to authenticate users (and to store user preferences).

1.5.4 Understanding User Preferences for Calendar Server Version 6.3

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 Java System Calendar Server 6.3 WCAP Developer's Guide*.

1.5.5 LDAP Groups Overview in Calendar Server Version 6.3

Groups are named collections of users. Each group has an LDAP entry, similar to a user or resource entry. You can use the same group entry for all services, such as calendar and messaging.

The following are a few facts about Calendar Server LDAP groups:

- Calendar Server groups can be either static or dynamic.
- Groups with calendar service can have their own default calendar.
- Calendar Server groups can consist of individuals, resources, and other groups (nested).

For more information about group calendars, see the following section: “1.5.7 Group Calendars Overview for Calendar Server Version 6.3” on page 46.

1.5.6 Autoprovisioning: Automatic Creation of Calendars in Calendar Server Version 6.3

Calendar databases can be automatically populated by setting *local.autoprovision="yes"* in the `ics.conf` file. In addition, domains must be calendar enabled (have calendar service), meaning the domain LDAP entry must contain the `icsCalendar` object class.

There are two ways for default calendars to be created automatically:

- When a user logs in for the first time, if the user's LDAP entry is found, the system enables it for calendar services and creates a default calendar.
- When an LDAP user, group or resource is invited to an event before the default calendar has been created, the system creates a default calendar for that entity.

For example, suppose *tchang* exists in the directory server but is not yet enabled for calendaring (that is, does not have a default calendar). With autoprovisioning turned on, and with the domain calendar enabled:

- When *tchang* logs into Calendar Server for the first time, the system automatically enables *tchang* for calendaring and creates a default calendar with the *calid tchang@hisdomain.com*.
- Alternately, if someone invites *tchang* to an event before the default calendar has been created, the system will automatically create a default calendar for him, if *user.invite.autprovision="yes"* in the `ics.conf` file.

  For groups invited, the default group calendar is created if the following `ics.conf` parameter is set: *groupAutoprovisioning="yes"*.

  For resources, likewise, the default resource calendar is created if the following `ics.conf` parameter is set: *resource.invite.autoprovision="yes"*.

For more information about the configuration file parameters necessary for users, resources and groups, see “4.3 Configuring Calendar for LDAP Users, Groups and Resources” on page 123.

1.5.7 Group Calendars Overview for Calendar Server Version 6.3

A group calendar can be created for any calendar-enabled LDAP group. This calendar can be scheduled much like an individual's calendar. Invitations sent to the group are scheduled to the group calendar and all individual member calendars. If a group calendar does not yet exist at the time it is invited to an event, and autoprovisioning is turned on, the system creates a calendar with a default set of properties and ACLs.

The following are some facts about group calendars:

- Group calendars do not have user interface preference like calendars for individuals because no one logs into a group calendar.
- Individuals need to subscribe to the group calendar to view it.
- The group's owner is responsible for setting the appropriate ACLs.
- Fetching free-busy information for a group calendar produces only the information for the group calendar, not the individual members' calendars.
- If a group calendar ACL does not allow invitation by an event organizer, the system returns an error. No group members are invited in this case.
- An organizer can invite a group using either its group calendar ID, or its mailing address.

For more information about Calendar Server users, see Chapter 14, “Administering Users, Groups, and Resources.”

1.5.8 Resources Overview for Calendar Server Version 6.3

A resource is anything that can be scheduled using a calendar, such as a conference room, or a projector. There is a separate resource LDAP entry for each such item. Create the LDAP entry and its associated calendar using the appropriate tools:

- For Schema version 2 - Use Delegated Administrator to create the resource LDAP entry, and the Calendar Server utility *resource* to create the calendar.
- For Schema version 1 - Use the `csresource create` command which creates both the resource LDAP entry and the calendar.

Note – It is not necessary to explicitly create resource calendars. With autoprovisioning enabled, the first time a resource is invited, the system will automatically create a resource calendar for it. This is the same behavior as for users and groups.

1.6 Data Formats and Standards Overview for Calendar Server Version 6.3

This section describes the following information about Calendar Server data:

- "1.6.1 Data Format for Calendar Server Version 6.3" on page 47
- "1.6.2 Import and Export of Calendar Data for Calendar Server Version 6.3" on page 47
- "1.6.3 Calendar Links for Data Exchange in Calendar Server Version 6.3" on page 47
- "1.6.4 Server Alarms in Calendar Server Version 6.3" on page 48
- "1.6.5 Support of ITIP/IMIP Standards in Calendar Server Version 6.3" on page 48

1.6.1 Data Format for Calendar Server Version 6.3

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

Calendar Server supports the following formats:

- XML (.xml) — The interface to Communications Express.
- iCalendar (.ical) — The default format.

1.6.2 Import and Export of Calendar Data for Calendar Server Version 6.3

Calendar data can be imported and exported in either iCalendar (`.ical`) or XML (`.xml`) format. Calendar Server administrators can import and export calendar data using the Calendar Server *csimport* and *csexport* utilities. End users can import and export calendar data using the Communications Express user interface.

1.6.3 Calendar Links for Data Exchange in Calendar Server Version 6.3

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

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

For information on how to link to a calendar, see "15.8 Linking to a Calendar" on page 313.

1.6.4 Server Alarms in Calendar Server Version 6.3

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.

1.6.5 Support of ITIP/IMIP Standards in Calendar Server Version 6.3

Calendar Server supports the ITIP/IMIP standards (RFC 2446 and RFC 2447), including ITIP methods `PUBLISH`, `REQUEST`, `REPLY`, and `CANCEL` for events.

1.7 LDAP Data Cache Option for Calendar Server Version 6.3

The LDAP data cache option 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.

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.

This section covers the following topics:

1.7.1 Considerations for Using the LDAP Data Cache for Calendar Server Version 6.3

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 you have deployed a "1.7.2 Master/Slave LDAP Configuration for Calendar Server Version 6.3" on page 49, where Calendar Server accesses the master LDAP directory through a slave LDAP directory server, there will be 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.

1.7.2 Master/Slave LDAP Configuration for Calendar Server Version 6.3

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 using an LDAP referral to the master directory server. The LDAP referral then 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.

  The following table lists operations and the LDAP attributes affected by such a delay:

Operation	LDAP Attributes
Autoprovisioning of calendars	`icsCalendar`, `icsSubscribed`, `icsCalendarOwned`, `icsDWPHost`
Calendar groups	`icsStatus`, `icsCalendar`
Calendar creation	`icsCalendarOwned`, `icsSubscribed`
Calendar subscription	`icsSubscribed`

Operation	LDAP Attributes
User options	`icsExtendedUserPrefs`, `icsFirstDay`, `icsTimeZone`, `icsFreeBusy`
Calendar searches	`icsCalendarOwned`

1.7.3 LDAP Data Cache for Calendar Server Version 6.3

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 `ldap_cache` database 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 multiprocessor system.
- Data persistence across user sessions The database is permanent and does not require refreshing.

1.7.4 LDAP Data Cache Limitations for Calendar Server Version 6.3

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.

1.8 Access Control for Calendar Server Version 6.3

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 section covers the following topics:

- "1.8.1 Secure Logins for Calendar Server Version 6.3" on page 51
- "1.8.2 Access Control by Users in Calendar Server Version 6.3" on page 52
- "1.8.3 Access Control Lists (ACLs) in Calendar Server Version 6.3" on page 52

1.8.1 Secure Logins for Calendar Server Version 6.3

When users log in to Calendar Server through Communications Express, by default the authentication process does not encrypt the login information, including user names and passwords. If you want secure logins at your site, configure Calendar Server to use the Secure Sockets Layer (SSL) protocol to encrypt the login data. For more information, see Chapter 7, "Configuring SSL," Configuring SSL.

1.8.2 Access Control by Users in Calendar Server Version 6.3

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 *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 “1.3 Special Accounts for Calendar Server Version 6.3” on page 39.

- 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 “15.8 Linking to a Calendar” on page 313.

1.8.3 Access Control Lists (ACLs) in Calendar Server Version 6.3

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 strings are case insensitive.

The following is a list of examples:

- `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 (`^`):

- "1.8.3.1 Who Element of Ace Strings in Calendar Server Version 6.3" on page 53 - The individual, user, domain, or type of user who the ACE applies to.
- "1.8.3.2 What Element of Ace Strings in Calendar Server Version 6.3" on page 54 - The target being accessed, such as a calendar or a calendar component such as an event, todo (task), or calendar property.
- "1.8.3.3 How Elements of Ace Strings in Calendar Server Version 6.3" on page 54 - The type of access control rights permitted, such as read, write, or delete.
- "1.8.3.4 Grant Element of Ace Strings in Calendar Server Version 6.3" on page 55 - 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`.

1.8.3.1 Who Element of Ace Strings in Calendar Server Version 6.3

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 1–2 "Who" Formats for Access Control Entry (ACE) Strings

Format	Description
user	Refers to a specific user. For example: jsmith.

TABLE 1–2 “Who” Formats for Access Control Entry (ACE) Strings *(Continued)*

Format	Description
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.
`@@{d|p|o|n}`	Refers to owners for the calendar: ■ `@@d` – domain of the primary owner ■ `@@p` – primary owner only ■ `@@o` – all owners, including the primary owner ■ `@@n` – not an owner

1.8.3.2 What Element of Ace Strings in Calendar Server Version 6.3

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

TABLE 1–3 “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

1.8.3.3 How Elements of Ace Strings in Calendar Server Version 6.3

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

TABLE 1–4 “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.

TABLE 1–4 “How” Types for Access Control Entry (ACE) Strings *(Continued)*

Type	Description
`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.
`z`	Self-administrating access - the authenticated user is granted the ability to add or remove an Access Control Entry. 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 Entry 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.

1.8.3.4 Grant Element of Ace Strings in Calendar Server Version 6.3

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

TABLE 1–5 Grant Values for Access Control Entry (ACE) Strings

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

1.8.3.5 Examples of ACEs for Calendar Server Version 6.3

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
  ```

1.8.3.6 Placing ACE's in an ACL for Calendar Server Version 6.3

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.

1.9 Internal Subsystems for Calendar Server Version 6.3

Sun Java System Calendar Server includes the following internal subsystems:

- "1.9.1 Protocol Subsystem" on page 57
- "1.9.2 Core Subsystem" on page 58
- "1.9.3 Database Subsystem" on page 58

The following graphic shows the logical flow through these subsystems.

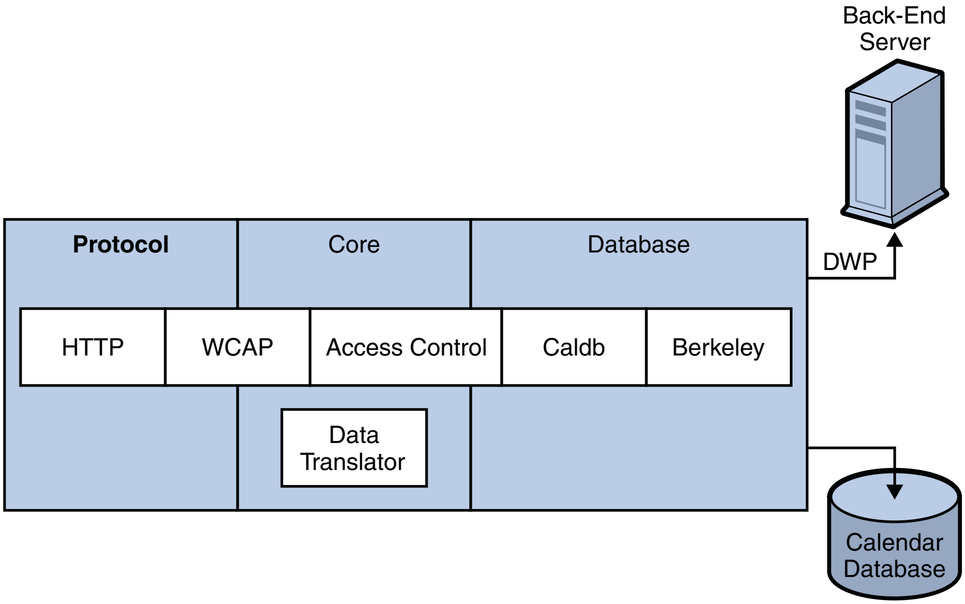


FIGURE 1–1 Calendar Server 6.3 Internal Subsystems Logical Flow

1.9.1 Protocol Subsystem

Clients retrieve calendar data by submitting requests using the HTTP protocol layer. This is a minimal HTTP server implementation, streamlined to support calendar requests. This is done by appending Web Calendar Access Protocol (WCAP) commands to the URL.

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 request output as XML or iCalendar wrapped in HTML.

For information about WCAP commands, see the *Sun Java System Calendar Server 6.3 WCAP Developer's Guide*.

1.9.2 Core Subsystem

The Core subsystem includes an access control component, a WCAP command interpretation component, and data translators to format data coming from the calendar database component. The Core subsystem processes calendar requests and generates XML and iCalendar output. The Core subsystem can also handle user authentication.

1.9.3 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 super set of the iCalendar standard.

The Database subsystem returns data in a low-level format, and the Core UI generator then translates the low-level data and sends it through WCAP.

For a distributed calendar database, Calendar Server uses the Distributed Wire Protocol (DWP) to provide a networking capability. For more information, see “1.10.5 Distributed Database Service: csdwpd in Calendar Server Version 6.3” on page 60.

For information about the calendar database, refer to Chapter 16, “Administering Calendar Server Databases with the csdb Utility.”

1.10 Services Running As Daemons in Calendar Server Version 6.3

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

- “1.10.1 Administration Service: csadmind in Calendar Server Version 6.3” on page 58
- “1.10.2 HTTP Service: cshttpd in Calendar Server Version 6.3” on page 59
- “1.10.3 Automatic Backup Service: csstored in Calendar Server Version 6.3” on page 59
- “1.10.4 Event Notification Service (ENS): csnotifyd and enpd in Calendar Server Version 6.3” on page 60
- “1.10.5 Distributed Database Service: csdwpd in Calendar Server Version 6.3” on page 60

1.10.1 Administration Service: csadmind in Calendar Server Version 6.3

The *csadmind* service manages alarm notifications, and group scheduling requests.

1.10.2 HTTP Service: cshttpd in Calendar Server Version 6.3

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 specified in the incoming WCAP command. Data can be formatted in standard RFC 2445 iCalendar format (`text/calendar`) or XML format (`text/xml`

1.10.3 Automatic Backup Service: csstored in Calendar Server Version 6.3

When properly configured, the `csstored` service creates automatic backups of the calendar database. You can configure Calendar Server for automatic backups when the `csconfigurator.sh` configuration program runs, or you can do it at a later time, as described in this guide.

If the service is started in the disabled state, it will send a message to the administrator every 24 hours stating that automatic backups are not enabled.

For instructions on how to configure this service to perform backups, see Chapter 9, “Configuring Automatic Backups (csstored).”

When configured properly, the service has the following functionality:

- Upon system start up and at 24 hour (default interval) intervals thereafter, takes a snapshot of the live Calendar Server calendar database. The interval is configurable. (If the service has been stopped and restarted, it does not take another snapshot unless the configured interval has elapsed since the last snapshot.)
- Verifies the database by running `csdb verify` against the backup copy.

  If the verify step fails (the database is corrupted), the service notifies the administrator. The administrator can put the live database in read-only mode, allowing you to troubleshoot the problem without having to shut down the databases. While in read-only mode, no modify or delete transactions are accepted (no logging). For more information about read-only mode, see “22.5.4 Preventing Service Interruptions When Your Database is Corrupted (Read-only Mode)” on page 373.

  Administrator intervention is required when a corruption is sensed. A notification is sent to the administrator.

If the verify succeeds, *csstored* performs the following additional tasks:

- Creates an archival backup consisting of the database snapshot and all the transaction log files that were applied to it since the previous snapshot.
- Creates a hot backup consisting of the database snapshot with the transaction log files applied to it.

  Should the live database become corrupted, a hot backup provides an immediate up to date backup of the database with a minimum of data loss and downtime.

  For information on how to restore an automatic backup copy, see "22.5.8 Restoring an Automatic Backup Copy" on page 381.

1.10.4 Event Notification Service (ENS): csnotifyd and enpd in Calendar Server Version 6.3

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 the subscriber. The default subscriber for the Calendar Server system is *csnotifyd*.

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

1.10.5 Distributed Database Service: csdwpd in Calendar Server Version 6.3

Using *csdwpd*, you can create a distributed calendar store. That is, use *csdwpd* to manage calendar databases spread across multiple back-end servers within the same Calendar Server configuration.

The *csdwpd* service runs in the background on back-end servers 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.

1.11 Public APIs for Calendar Server Version 6.3

Calendar Server includes the following APIs:

- "1.11.1 Web Calendar Access Protocol (WCAP) in Calendar Server Version 6.3" on page 61
- "1.11.2 Event Notification Service (ENS) API for Calendar Server Version 6.3" on page 62

1.11.1 Web Calendar Access Protocol (WCAP) in Calendar Server Version 6.3

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 RFC 2445 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 Java System Calendar Server 6.3 WCAP Developer's Guide*.

1.11.2 Event Notification Service (ENS) API for Calendar Server Version 6.3

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: Publisher API, Subscriber API, and Publish and Subscribe Dispatcher API.

For information about the ENS API, see the *Sun Java Communications Suite 5 Event Notification Service Guide*.

Note – The Calendar Server software also contains support for Java Message Queue for notification, but `csnotifyd` does not subscribe to it. Thus, it is not part of the default alarms and notification system. For more information, refer to the Sun Java System Java Message Queue documentation.

PART II

Post Installation Configuration for Calendar Server 6.3 Software

The chapters in this part describe the configuration and migration steps you must perform, after installation, before you can use Calendar Server.

This part includes the following chapters:

◆ ◆ ◆ CHAPTER 2

Initial Runtime Configuration Program for Calendar Server 6.3 software (csconfigurator.sh)

After you install Calendar Server, and before running it, you must configure it. It is important that you run the two configuration programs in the following order:

1. `comm_dssetup.pl`

   Configure the LDAP directory server as instructed in the *Sun Java System Communications Suite 5 Installation and Configuration Guide*.
2. `csconfigurator.sh`

   Configure Calendar Server as described in this chapter.

This chapter contains the following topics:

- "2.1 Gathering Your Configuration Information for Calendar Server 6.3 Software" on page 65
- "2.2 Running csconfigurator.sh" on page 69

Note – If you had an earlier version of Calendar Server or Messaging Server installed, you might need to migrate your LDAP directory entries from Schema version 1 to Schema version 2.

Do not run the configuration utility described in this chapter until you have read the *Sun Java Communications Suite 5 Schema Migration Guide*. It will instruct you on the timing and options for running the configuration utilities.

2.1 Gathering Your Configuration Information for Calendar Server 6.3 Software

The Calendar Server configuration program `csconfigurator.sh`, creates a new `ics.conf` configuration file in the following directory:

For Solaris: `/etc/opt/SUNWics5/config`

For Linux: `/etc/opt/sun/calendar/config`

The configuration program will ask you many questions for which you must enter specific information about your installation.

Before running the configuration program, you should gather the following configuration information:

- "2.1.1 LDAP Server Options" on page 66
- "2.1.2 Directory Manager Options" on page 66
- "2.1.3 Calendar Server Administrator" on page 67
- "2.1.4 Email and Email Alarms Options" on page 67
- "2.1.5 Runtime Configuration Options" on page 67
- "2.1.6 Calendar Server Startup" on page 68
- "2.1.7 Database, Logs, and Temporary Files Directories" on page 68

To help you keep track of the configuration information, use the worksheets in Appendix B, "Calendar Server Configuration Worksheet." (However, you should determine this information before you run the Java Enterprise System installer to avoid conflicts (such as port numbers) with other component products.)

2.1.1 LDAP Server Options

Calendar Server requires a directory server for user authentication and for the storage and retrieval of user preferences. The following table lists the options used to gather host and port information for the LDAP server.

TABLE 2–1 User Preferences Directory Options

Option	Description
LDAP Server Host Name	Host name of the LDAP directory server you are using for user authentication and user preferences. The default is the current host.
LDAP Server Port	Port number that the LDAP directory server listens on. The default is 389.

2.1.2 Directory Manager Options

The following table lists the options used to gather the name and password of the user that is designated the Directory Manager.

TABLE 2–2 Directory Manager Options

Option	Description
Directory Manager DN	User name that can make changes in the directory server schema. The default is `cn=Directory Manager`.
Directory Manager Password	Password of the Directory Manager DN. The password is not stored in plain text. There is no default.

2.1.3 Calendar Server Administrator

The Calendar Server Administrator is the user account that overrides any other Calendar Server ACLs. The Calendar Server Administrator user account must exist in your user authentication directory server. It is also used for proxy authentication. The following table lists the options used to gather the Calendar Server Administrator's user ID and password.

TABLE 2–3 Calendar Server Administrator Options

Option	Description
Administrator User ID	User ID of the Calendar Server Administrator; must be a user in the above LDAP directory server. The default is *calmaster*.
Administrator Password	Password of the Calendar Server Administrator. There is no default.

2.1.4 Email and Email Alarms Options

You can configure Calendar Server to send an email alarm message to a Calendar Server Administrator in case a server problem occurs. The following table lists the options used to gather email information.

TABLE 2–4 Email and Email Alarms Options

Option	Description
Email Alarms	Enables or disables email alarms. The default is Enabled.
Administrator Email Address	Email address of the Calendar Server Administrator who will receive the email alarm messages.
SMTP Host Name	Host name of the SMTP server used by the Calendar Server system to send email alarm messages. The default is the current host.

2.1.5 Runtime Configuration Options

You can configure the following Calendar Server runtime and system resource options.

TABLE 2–5 Runtime Configuration Options

Option	Description
Service Port	Port number that Calendar Server listens on to provide Web (HTTP) access to users. The default is `80`.
Maximum Sessions	Maximum number of Calendar Server sessions to allow concurrently. The default is `5000`.
Maximum Threads	Maximum number of Calendar Server threads to allow concurrently. The default is `20`.
Number of Server Processes	For Solaris: Maximum number of Calendar Server processes to run concurrently. The default is the number of CPUs on the server where you are installing Calendar Server. **For Linux:** Only one process can run at a time.
Runtime User ID	UNIX user name under which Calendar Server will run. This user name should not be *root*. If the account does not exist, the configuration program will create it. The default is *icsuser*.
Runtime Group ID	UNIX group under which Calendar Server will run. If the group does not exist, the configuration program will create it. The default is *icsgroup*.

2.1.6 Calendar Server Startup

You can configure the following options to automatically start Calendar Server.

TABLE 2–6 Calendar Server Startup Options

Option	Description
Start after successful installation	Whether to start Calendar Server automatically after a successful installation. The default is checked.
Start on system startup	Whether to start Calendar Server automatically after a system startup. The default is checked.

2.1.7 Database, Logs, and Temporary Files Directories

Calendar Server creates and stores information in calendar database files, log files, and temporary files in specific directories.

TABLE 2–7 Database, Logs, and Temporary Files Directories Options

Option	Description
Database Directory	Directory where the Calendar Server system creates and stores the calendar database (`*.db`) files. The default is: `/var/opt/SUNWics5/csdb`
Logs Directory	Directory where Calendar Server writes log files. The default is: `/var/opt/SUNWics5/logs`
Temporary Files Directory	Directory where the Calendar Server system writes temporary files. The default is: `/var/opt/SUNWics5/tmp`
Archive and hot backup Directories	Directory where the Calendar Server system writes archive backups. User defined directory for storing the daily snapshot and transactions logs. If both types of backups are desired, then place them in different directories. If no directory is specified, backups are stored in the current directory.
Attachment Store Directory	Directory where the Calendar Server system stores attachments to events and tasks.

Note – Do not change the location or names of the logs and temporary files directories.

2.2 Running csconfigurator.sh

You can run the configuration program from a graphical user interface (GUI), or from the command line.

If you run the program remotely, you must set your `DISPLAY` environment variable properly and allow X-Windows connections from the server to display on your computer. For example, to use the *xhost* utility, execute the following command on your computer:

```
# xhost +
```

This section contains the following topics:

- “To Run the Configuration Program from the Command Line” on page 70
- “To Run the Configuration Program from the GUI” on page 70

▼ To Run the Configuration Program from the Command Line

1. **Login as or become superuser (*root*).**

2. **Change to the `/opt/SUNWics5/cal/sbin` directory.**

3. **Run the script using the options chosen from the following table:**

Option	Description
`-nodisplay`	Run the configuration script in text-only mode (non-GUI).
`-noconsole`	Do not display text output. Use this with `-nodisplay` to run the configuration script in silent mode.
`-novalidate`	Do not validate input field text.
`-saveState [statefile]`	Save the answers that you input in response to configuration questions to a state file (text file). Unless you specify a fully qualified path for the state file, it will be saved in the default directory: `/opt/SUNWics5/cal/jconfigure`.
`-state [statefile]`	Use the state file for setting input values. This option must be used in conjunction with *-novalidate* and *-noconsole*.

For example, to run the configuration script in command-line mode, issue the following command:

```
./csconfigurator.sh -nodisplay
```

The command-line version asks for the same information and in the same order as the GUI. Default values are indicated in square brackets, `[]`. To accept a default value, press Enter on your keyboard.

Note – For the text of the information contained in the various questions presented by the script, see the text in the GUI screens shown in the sections that follow.

▼ To Run the Configuration Program from the GUI

1. **Login as or become superuser (*root*).**

2. **Change to the `/opt/SUNWics5/cal/sbin` directory.**

3 **Run the command:**

```
./csconfigurator.sh
```

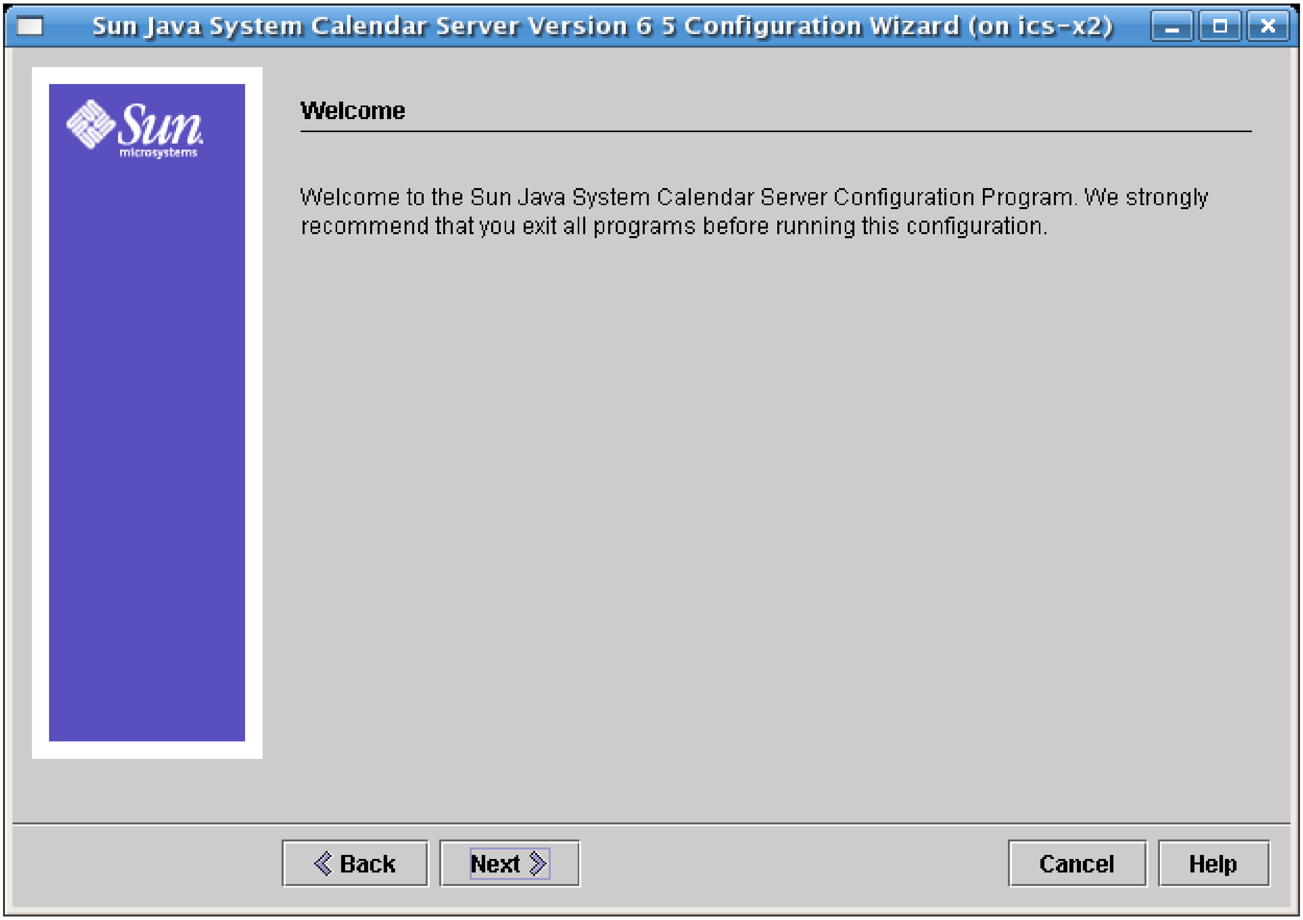
The configuration program displays the following series of screens:

- "2.3 Welcome Screen" on page 71
- "2.4 Administration, User Preferences and Authentication Screen" on page 72
- "2.6 Email and Email Alarms Screen for Calendar Server 6.3" on page 76
- "2.7 Runtime Configuration Screen for Calendar Server 6.3" on page 77
- "2.9 Directories to Store Configuration and Data Files Screen for Calendar Server 6.3" on page 82
- "2.10 Archive and Hot Backup Configuration Screen for Calendar Server 6.3" on page 84
- "2.11 Ready to Configure Screen for Calendar Server 6.3" on page 86
- "2.13 Configuration Summary Screen for Calendar Server 6.3" on page 89

Caution – The configuration program only configures a single domain. If you plan to use multiple domains, you need to add the domains using Delegated Administrator.

Note – The title bars for all screens are incorrect. The version is 6.3, not 6.5 as shown.

2.3 Welcome Screen



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)
Sun microsystems
Back
Next
Cancel
Help

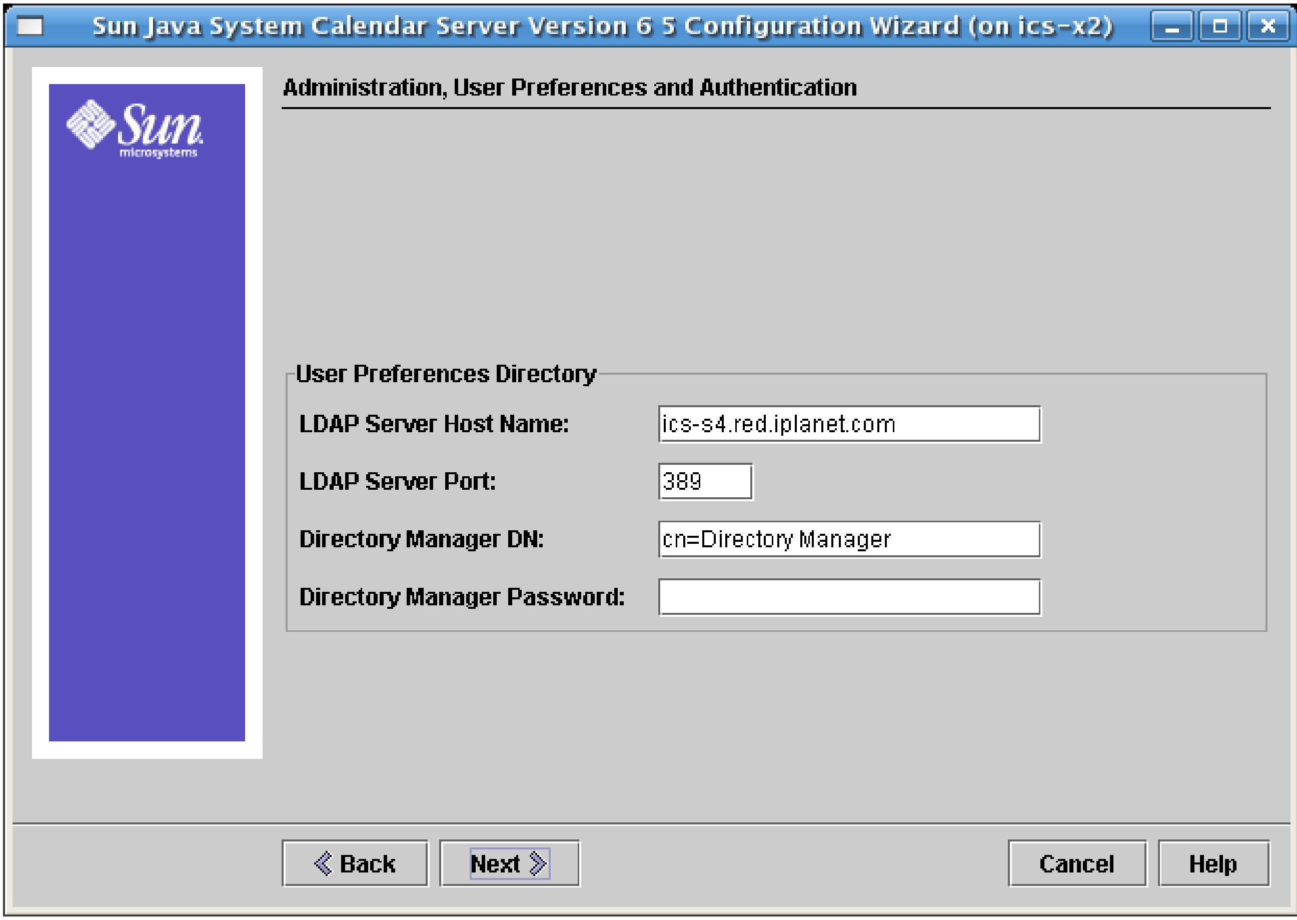
Welcome

Welcome to the Sun Java System Calendar Server Configuration Program. We strongly recommend that you exit all programs before running this configuration.

FIGURE 2–1 Calendar Server Configuration Program Welcome Screen

Click Next to continue or Cancel to exit.

2.4 Administration, User Preferences and Authentication Screen



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

Administration, User Preferences and Authentication

User Preferences Directory

LDAP Server Host Name: ics-s4.red.iplanet.com

LDAP Server Port: 389

Directory Manager DN: cn=Directory Manager

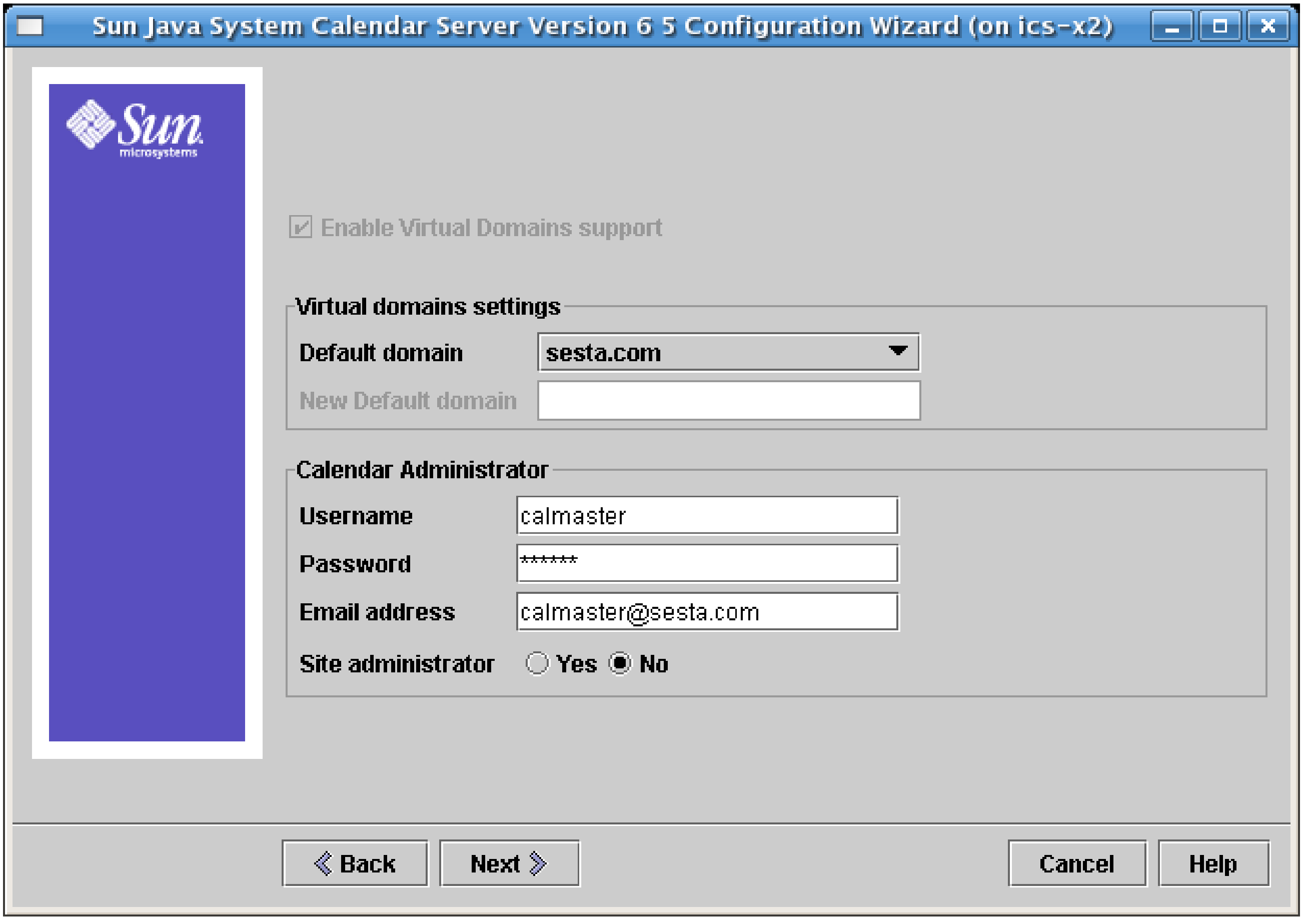
Directory Manager Password:

FIGURE 2–2 Administration, User Preferences and Authentication Configuration Screen

2.4.1 User Preferences Directory Options

LDAP Server Host Name	Host name of the LDAP directory server you are using for user authentication. Default: current host
LDAP Server Port	Port number that the LDAP server listens on. Default: 389
Directory Manager DN	User name that can make changes in the directory server schema. Default: `cn=Directory Manager`
Directory Manager Password	Password of the Directory Manager. This will not be stored in plain text. Default: None

2.5 Virtual Domains and Calendar Administrator Screen



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

☑ Enable Virtual Domains support

Virtual domains settings

Default domain: sesta.com

New Default domain:

Calendar Administrator

Username: calmaster

Password: ******

Email address: calmaster@sesta.com

Site administrator: ○ Yes ● No

FIGURE 2–3 Virtual Domains and Calendar Administrator Screen

Note – Virtual domains, hosted domains and multiple domains are all names for the same ability to have more than one LDAP domain with its corresponding user and group records.

If you are upgrading from a non-virtual domain environment, the Enable Virtual Domains Support checkbox must be selected. If you already have a multiple domain environment, the checkbox is greyed out. Virtual domains support is now the default behavior of Calendar Server, and is not optional.

2.5.1 Virtual Domains Settings for Calendar Server 6.3

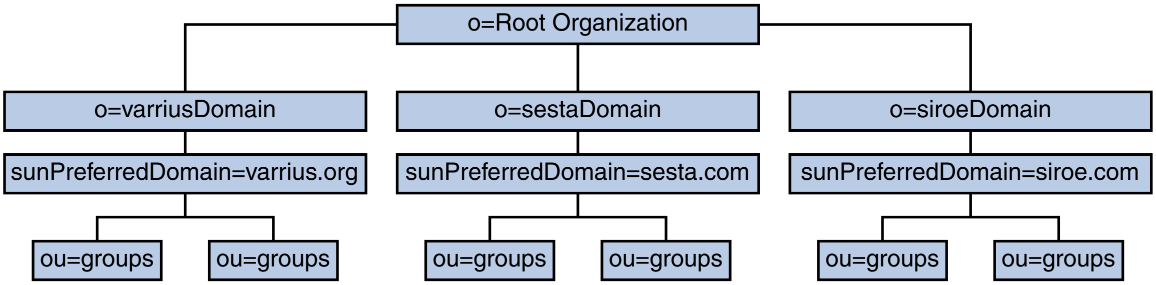


FIGURE 2–4 Virtual Domain Structure

Virtual domains support is now the default behavior for Calendar Server for fresh installations. Using the configuration program graphical user interface, enter a default domain name in the New Default domain input box. The configuration program then creates the domain for you.

Choose your default domain from one of those showing in the Default domain box. If you already used multiple domains in the previous version of Calendar Server, and you do not want to use the domain showing in the Default domain box, click the box to see the list of domains you can choose from and select a new default domain.

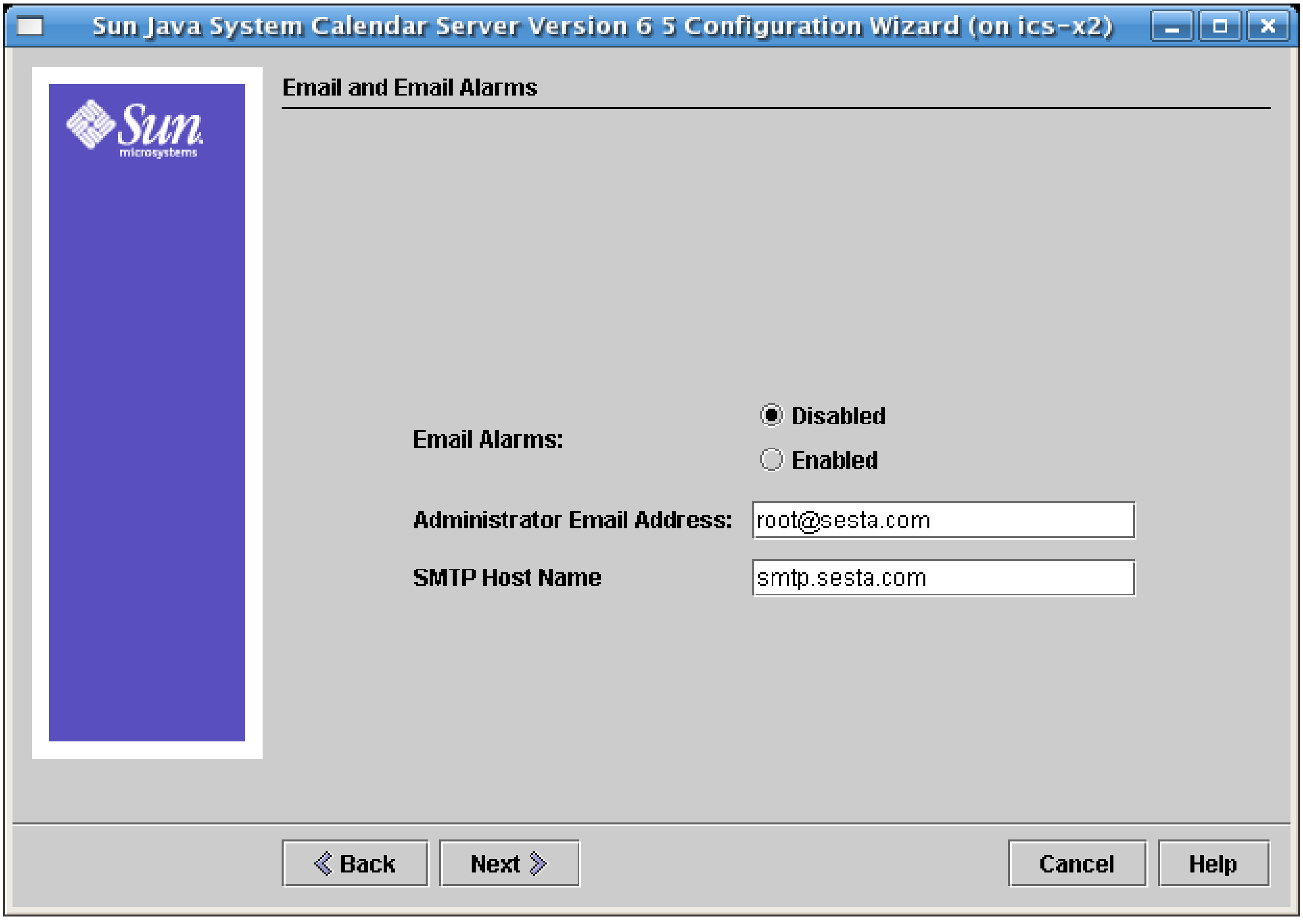
2.5.2 Calendar Administrator Name and Password for Calendar Server 6.3

Username	Username of the Calendar Server Administrator. Default: *calmaster*
Administrator Password	Password of the Calendar Server Administrator. Default: *None*
Email Address	Email address for the Calendar Server Administrator.
Site Administrator	The Site Administrator is the user that has proxy authentication rights across domains.

Click the appropriate response: **Yes** if the Calendar Administrator is also the Site Administrator. **No** if the Calendar Administrator is not the Site Administrator.

Click Next to continue, Back to return to the previous screen, or Cancel to exit.

2.6 Email and Email Alarms Screen for Calendar Server 6.3


Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)
Sun microsystems
Email and Email Alarms
Back
Next
Cancel
Help

Email Alarms: ● Disabled ○ Enabled

Administrator Email Address: root@sesta.com

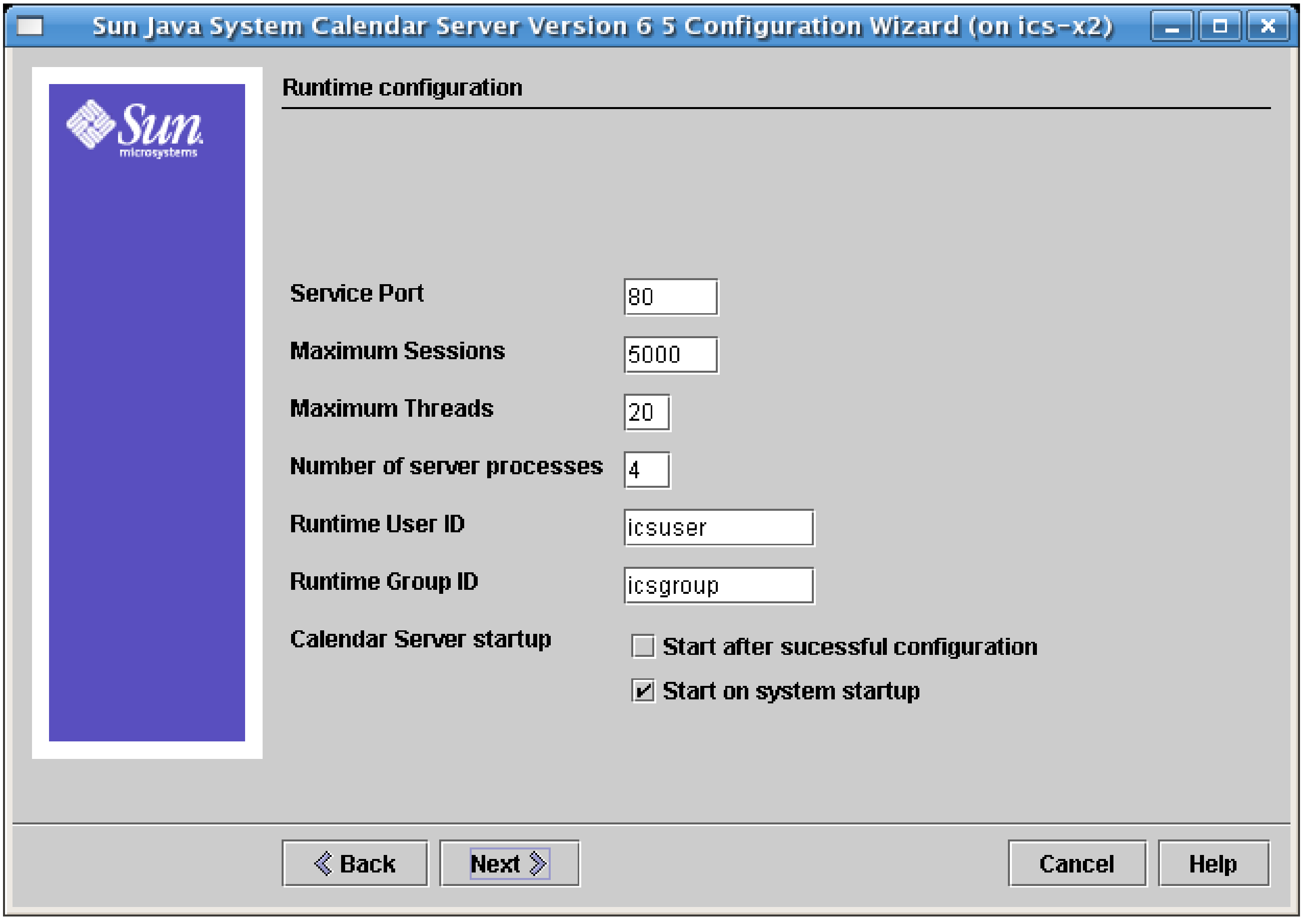
SMTP Host Name smtp.sesta.com

FIGURE 2–5 Email and Email Alarms Configuration Screen

Email Alarms	Specifies whether Calendar Server should send an email alarm message to a Calendar Server administrator in case a server problem occurs. Default: *disabled*. If you choose *Disabled*, no administrator receives email alarms for server problems.
Administrator Email Address	Email address of the Calendar Server Administrator who will receive the email alarm messages. Default: *None*
SMTP Host Name	Host name of the SMTP server where used to send alarm messages. Default: *current host*.

Click Next to continue, Back to return to the previous screen, or Cancel to exit.

2.7 Runtime Configuration Screen for Calendar Server 6.3



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

Runtime configuration

Service Port: 80

Maximum Sessions: 5000

Maximum Threads: 20

Number of server processes: 4

Runtime User ID: icsuser

Runtime Group ID: icsgroup

Calendar Server startup:

□ Start after sucessful configuration

☑ Start on system startup

FIGURE 2–6 Runtime Configuration Screen

Service Port	Port number that Calendar Server listens on to provide Web (HTTP) access to users. Default: `80`.
Maximum Sessions	Maximum number of concurrent Calendar Server sessions. Default: `5000`
Maximum Threads	Maximum number of concurrent Calendar Server threads. Default: `20`
Number of Server Processes	Maximum number of Calendar Server processes to run on the server. Default: Number of CPUs on the server where you are installing Calendar Server.

Runtime User ID

UNIX user name under which Calendar Server will run. If the account does not exist, the configuration program will create it. Default: *icsuser*

Caution – Do not use *root* as the Runtime User ID.

Runtime Group ID

UNIX group under which Calendar Server will run. If the group does not exist, the configuration program will create it. Default: *icsgroup*

Calendar Server Startup Options

Select one or both options by clicking in the check box.

- Start after successful configuration

  Specifies whether to start Calendar Server automatically after this configuration program successfully finishes running.

- Start on system startup

  Specifies whether to start Calendar Server automatically after a system startup.

Note – By default, only the Start on system startup checkbox is selected.

Click Next to continue, Back to return to the previous screen, or Cancel to exit.

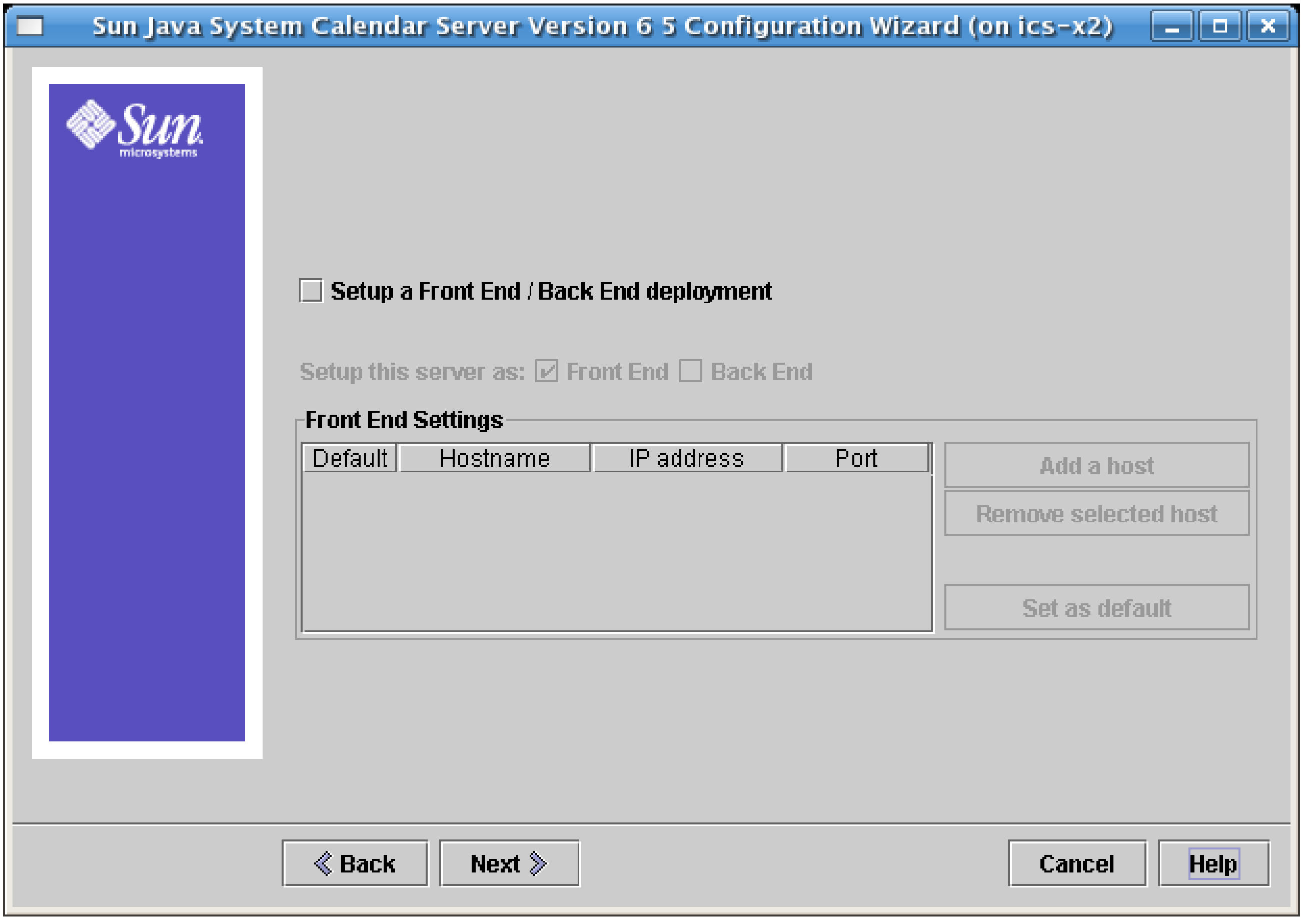
2.8 Set Up a Front End-Back End Deployment Screen for Calendar Server 6.3

Choose whether to configure this server as a single server deployment, or a front-end, back-end deployment. If you choose to have a single server instance of Calendar Server, then do not select the checkbox on this screen. If you want to put your Calendar Server databases on one or more servers, while keeping the processes that communicate with the client on a different server, select the checkbox.

This section covers the following topics:

- "2.8.1 Single Server Deployment for Calendar Server 6.3" on page 80
- "2.8.2 Front-End and Back-End Deployment for Calendar Server 6.3" on page 81

2.8.1 Single Server Deployment for Calendar Server 6.3



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

□ Setup a Front End / Back End deployment

Setup this server as: ☑ Front End □ Back End

Front End Settings

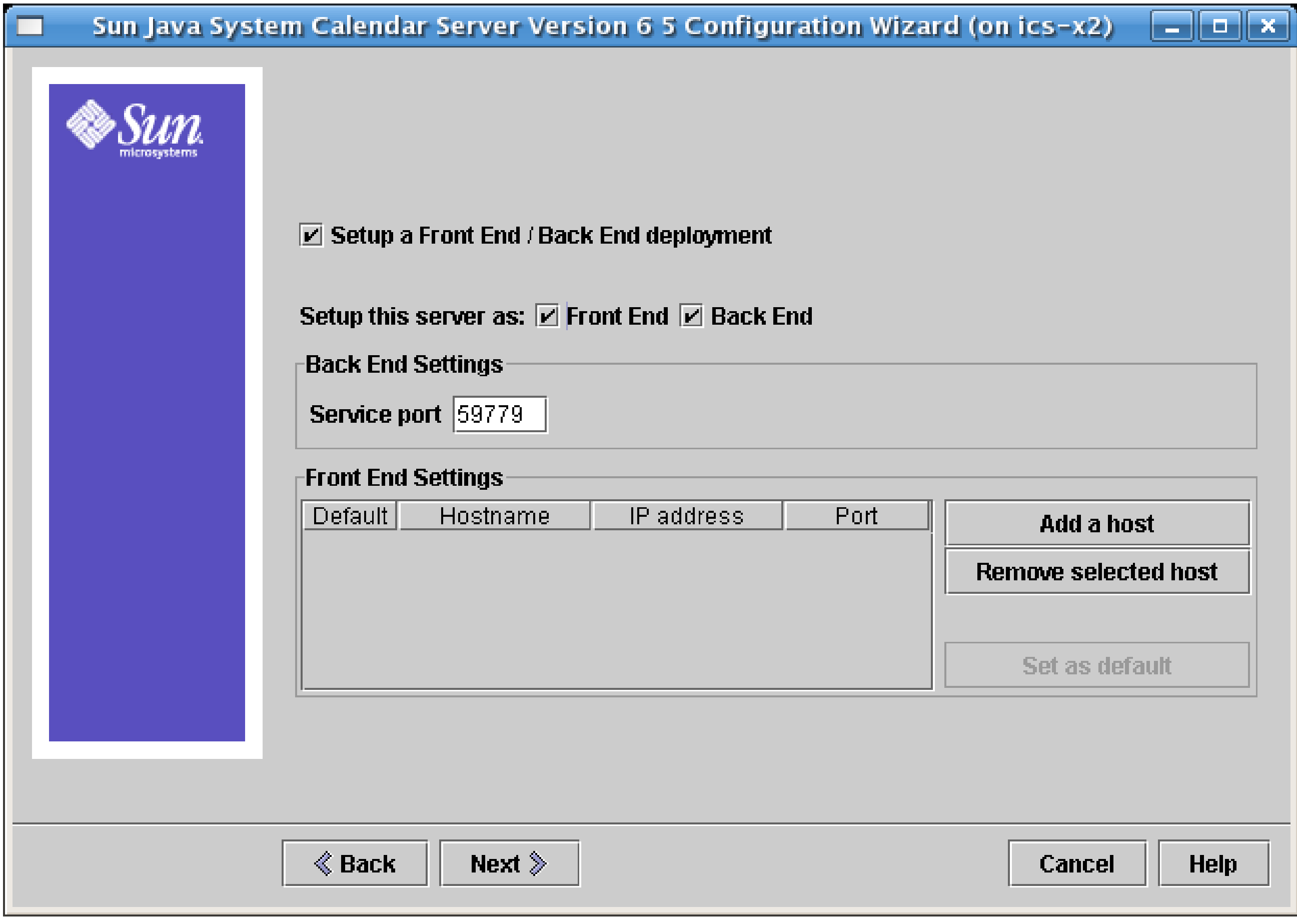
Default	Hostname	IP address	Port

FIGURE 2–7 Single Server Deployment

Do not change any part of this screen if you want a single server deployment where both the administrative processes and the databases reside on one server. Click Next to continue.

If you wish to deploy separate Front End and Back End machines, click the checkbox labeled: Setup a Front End/Back End deployment. The screen will change and you will be allowed to configure the front-end and back-end servers separately, as shown in the following two screen shots.

2.8.2 Front-End and Back-End Deployment for Calendar Server 6.3



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

☑ Setup a Front End / Back End deployment

Setup this server as: ☑ Front End ☑ Back End

Back End Settings

Service port 59779

Front End Settings

Default	Hostname	IP address	Port

FIGURE 2–8 Set Up a Front-End and Back-End Server

To complete this screen, perform the following steps:

1. To configure the back-end server, that is, the server on which to store calendar databases, you need only specify the service port.

   The service port entry box is pre-filled with the port named in the `ics.conf` parameters *service.dwp.server.hostname.port* and *service.dwp.port*.

If you want to change the port number, enter the desired port number in the Service Port entry box.

2. To configure the front-end server, click Add a Host and then enter the host name and IP address of the server you are configuring.

Note – Add only the server you are currently configuring to the list. If you plan to configure other front-end servers, add them at the time you configure them. (You must run the configuration program on each server you add to your configuration.)

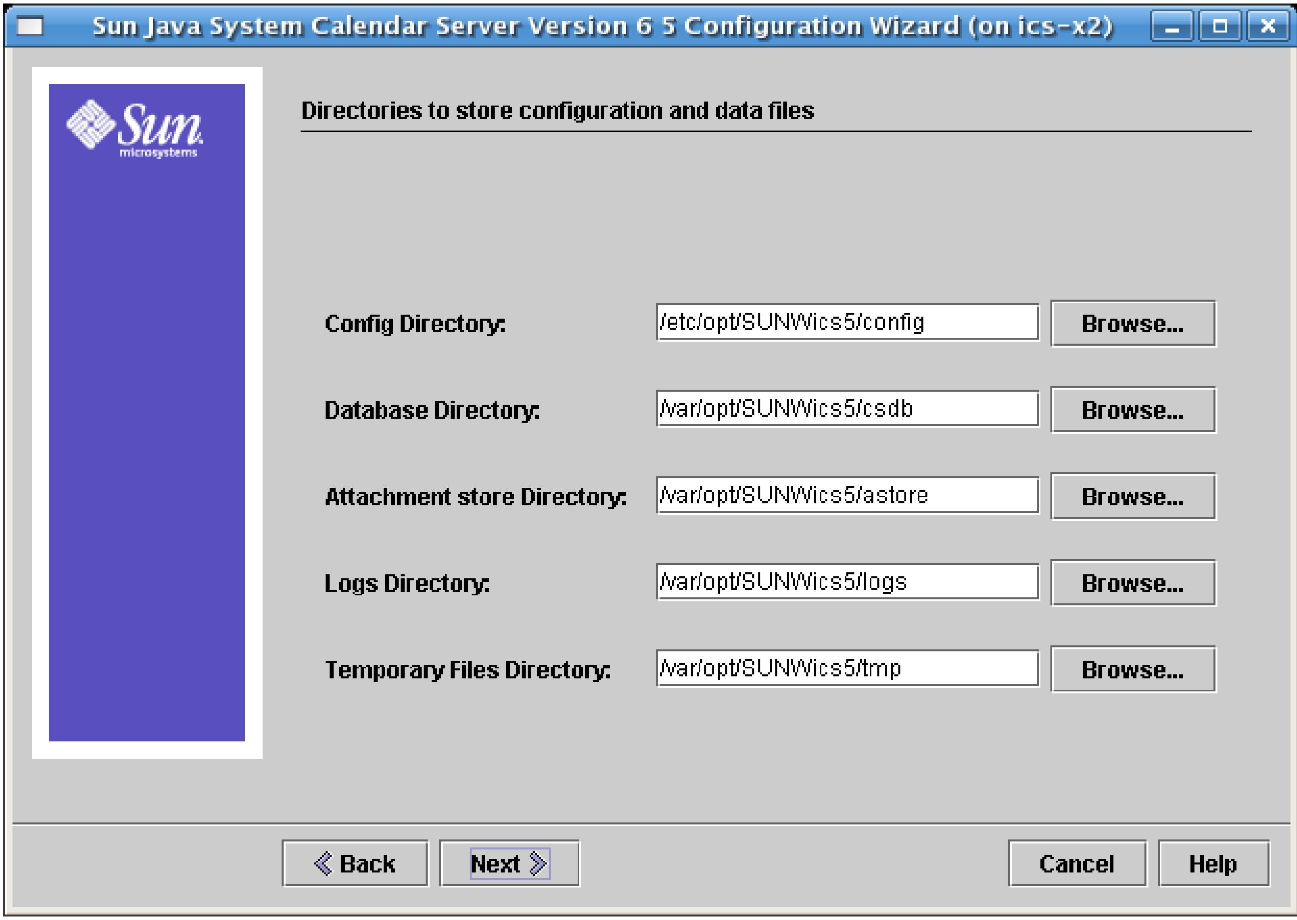
---

3. If this server is the default front-end server, select the Default checkbox.
4. Click Next.

Note – You may also remove servers from this list by clicking Remove Selected Host.

2.9 Directories to Store Configuration and Data Files Screen for Calendar Server 6.3

Accept the default directories on this screen. While you are allowed to choose the store configuration and data files directories, it is not advised.



Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

Directories to store configuration and data files

Config Directory: /etc/opt/SUNWics5/config

Database Directory: /var/opt/SUNWics5/csdb

Attachment store Directory: /var/opt/SUNWics5/astore

Logs Directory: /var/opt/SUNWics5/logs

Temporary Files Directory: /var/opt/SUNWics5/tmp

FIGURE 2–9 Select Directories Configuration Screen

Config Directory

Directory where the configuration file (`ics.conf`) resides.

Database Directory

Directory where Calendar Server creates and stores the calendar database files. Default: `/var/opt/SUNWics5/csdb`

Attachment Store Directory

Directory where the attachment store resides. Default: `/var/opt/SUNWics5/astore`

Logs Directory

Directory where Calendar Server writes log files. Default: `/var/opt/SUNWics5/logs`

Temporary Files Directory

Directory where the Calendar Server writes temporary files. Default: `/var/opt/SUNWics5/tmp`

Then, Click Next to continue, Back to return to the previous screen, or Cancel to exit.

Note – If any of these directories do not already exist, a pop-up window appears for each missing directory. Click the appropriate button to choose whether to have the configuration program create the new directory, or to return you to the screen where you can choose a different directory.

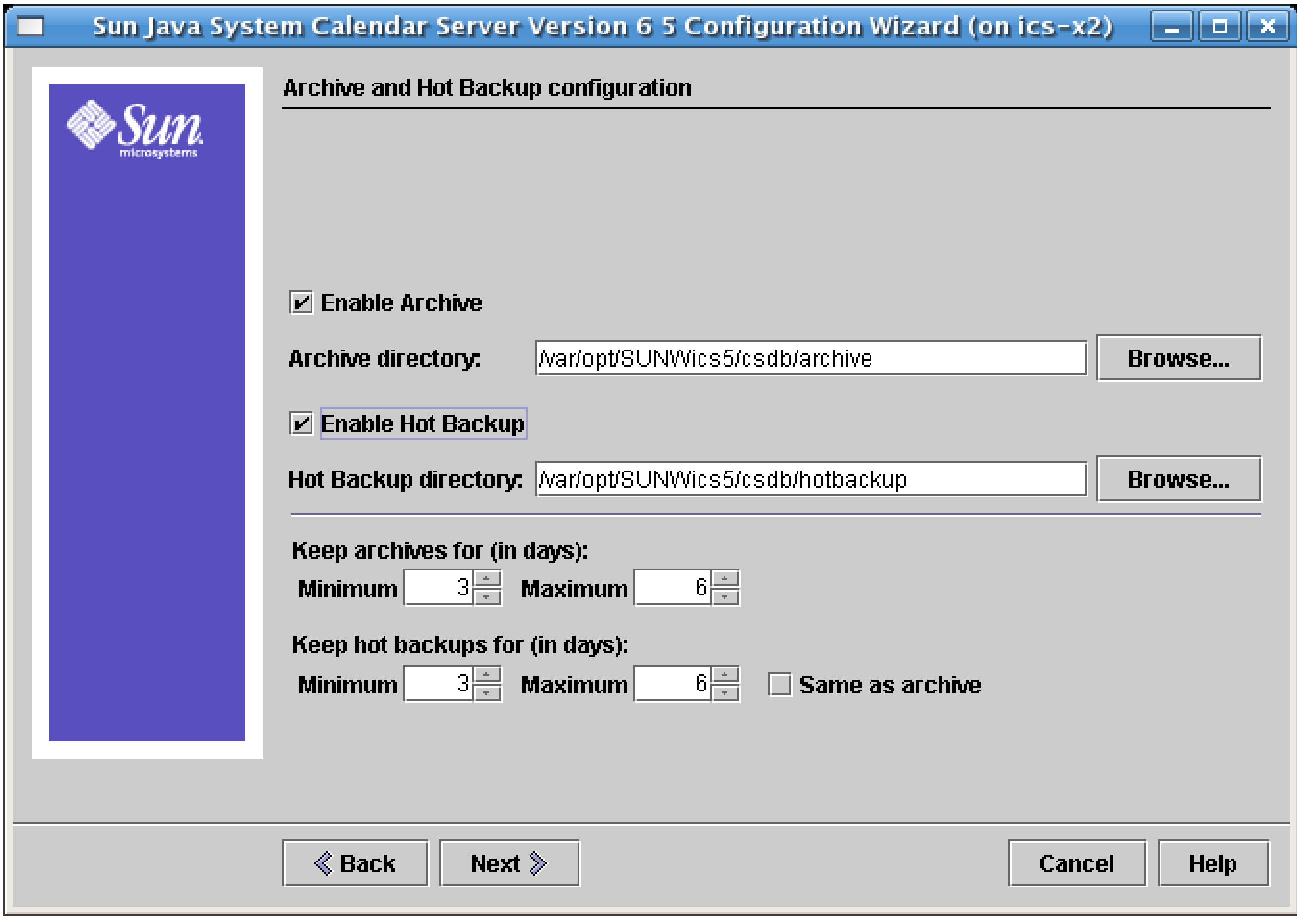
For any directory that already exists but is not empty, a pop-up window appears with two choices. Click the appropriate button to accept the directory anyway, or to return to the screen where you can choose a different directory.

2.10 Archive and Hot Backup Configuration Screen for Calendar Server 6.3

This screen allows you to select both automatic backup types, or either one of the two, or none. Select or deselect the boxes appropriately. Using both archive backups and hot backups is strongly recommended.

Tip – Prevent the catastrophic loss of all your database copies due to an equipment failure. Keep your automatic backup copies on a disk or disk system other than the one where your live databases reside.

For information on automatic backups, see Chapter 9, “Configuring Automatic Backups (csstored).”


Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)

Archive and Hot Backup configuration

☑ Enable Archive

Archive directory: /var/opt/SUNWics5/csdb/archive

☑ Enable Hot Backup

Hot Backup directory: /var/opt/SUNWics5/csdb/hotbackup

Keep archives for (in days):

Minimum 3 Maximum 6

Keep hot backups for (in days):

Minimum 3 Maximum 6 □ Same as archive

FIGURE 2–10 Archive and hot backup Configuration Screen

Enable Archive

When this box is selected (default), *csstored* will take a snapshot of your calendar databases every 24 hours. Throughout the day, at regular intervals, it stores the transaction log files for that day with the snapshot in the archive backup directory.

If this box is not checked, the Archive Directory input field is greyed out.

Archive Directory

Choose the backup directory by clicking Browse, or accept the default.

Enable Hot Backup

When this box is selected (default), *csstored* takes a snapshot of your calendar databases every 24 hours, then applies the transaction logs to the snapshot at a set interval (default is two minutes), throughout the day, ensuring a nearly complete duplicate of your live database.

If this box is not checked, the Hot Backup Directory input field is greyed out.

Hot Backup Directory

Choose the backup directory by clicking Browse, or accept the default.

Keep Archives for (in days)

This field is only active if the Enable Archive box is selected; otherwise, it is greyed out.

Click the up or down arrows in the Minimum and Maximum fields to select range of days of archival backups to keep in the backup directory.

Keep Hot Backups for (in days)

This field is only active if the Enable Hot Backup box is selected; otherwise, it is greyed out.

You can set the number of hot backups to keep in two ways:

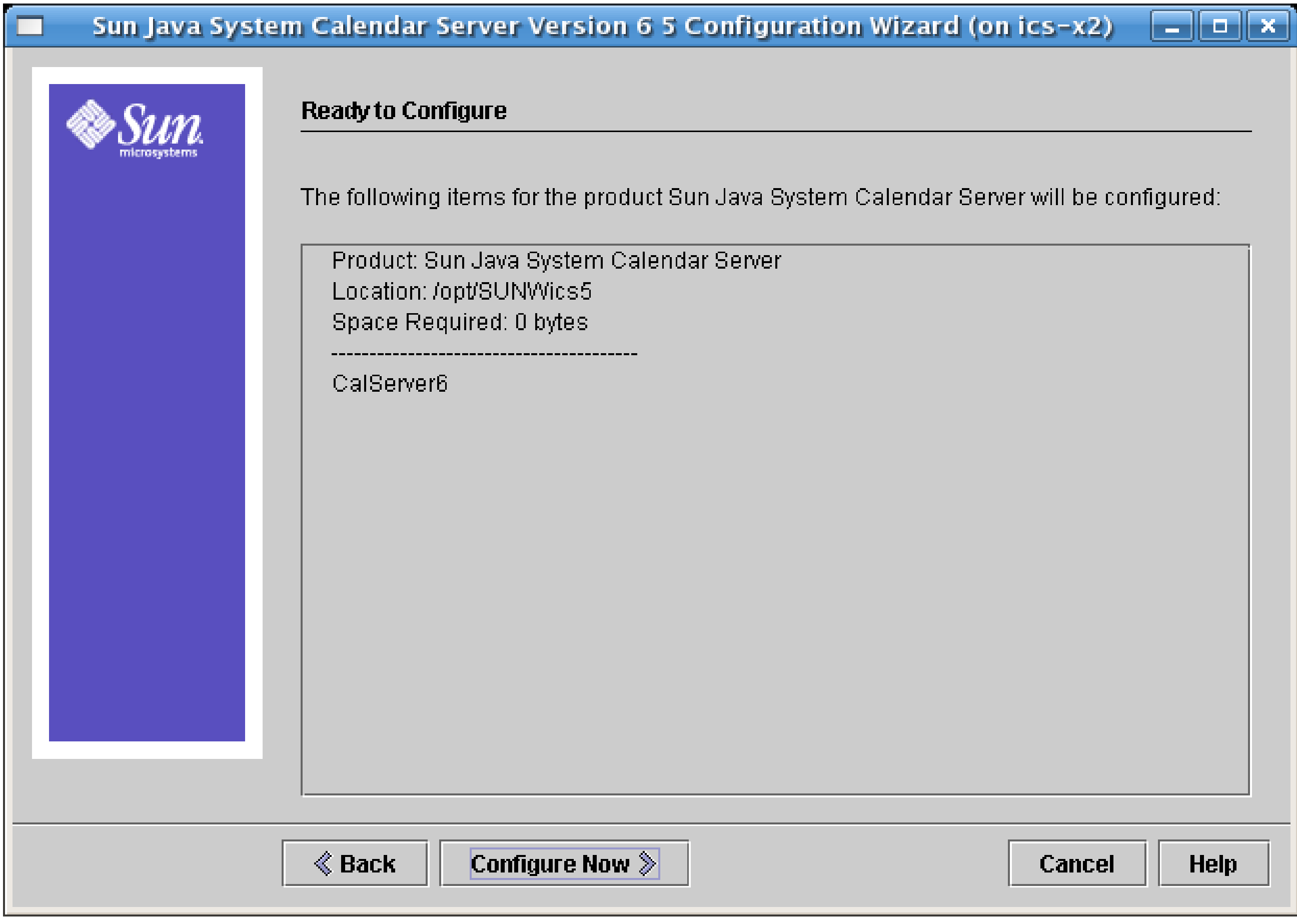
- Click the up or down arrows in the Minimum and Maximum fields to select the range of days of hot backups to keep in the directory.

  The number of copies actually stored at any one time depends on the size of the files and the size of the directory. When either the size limits, or maximum number of copies exceeds the limit, the oldest copies are purged down to the minimum number specified on this configuration screen.
- If you want the same settings for Hot Backups as for Archival Backups, you can check the Same As Archive box.

Click Next to continue, Back to return to the previous screen, or Cancel to quit the configuration program.

2.11 Ready to Configure Screen for Calendar Server 6.3

Up to now the screens have been gathering data needed for the configuration and performing some validity checking. You can go back and redo the configuration information at this point, or start the configuration.


Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)
Sun microsystems
Back
Configure Now
Cancel
Help

Ready to Configure

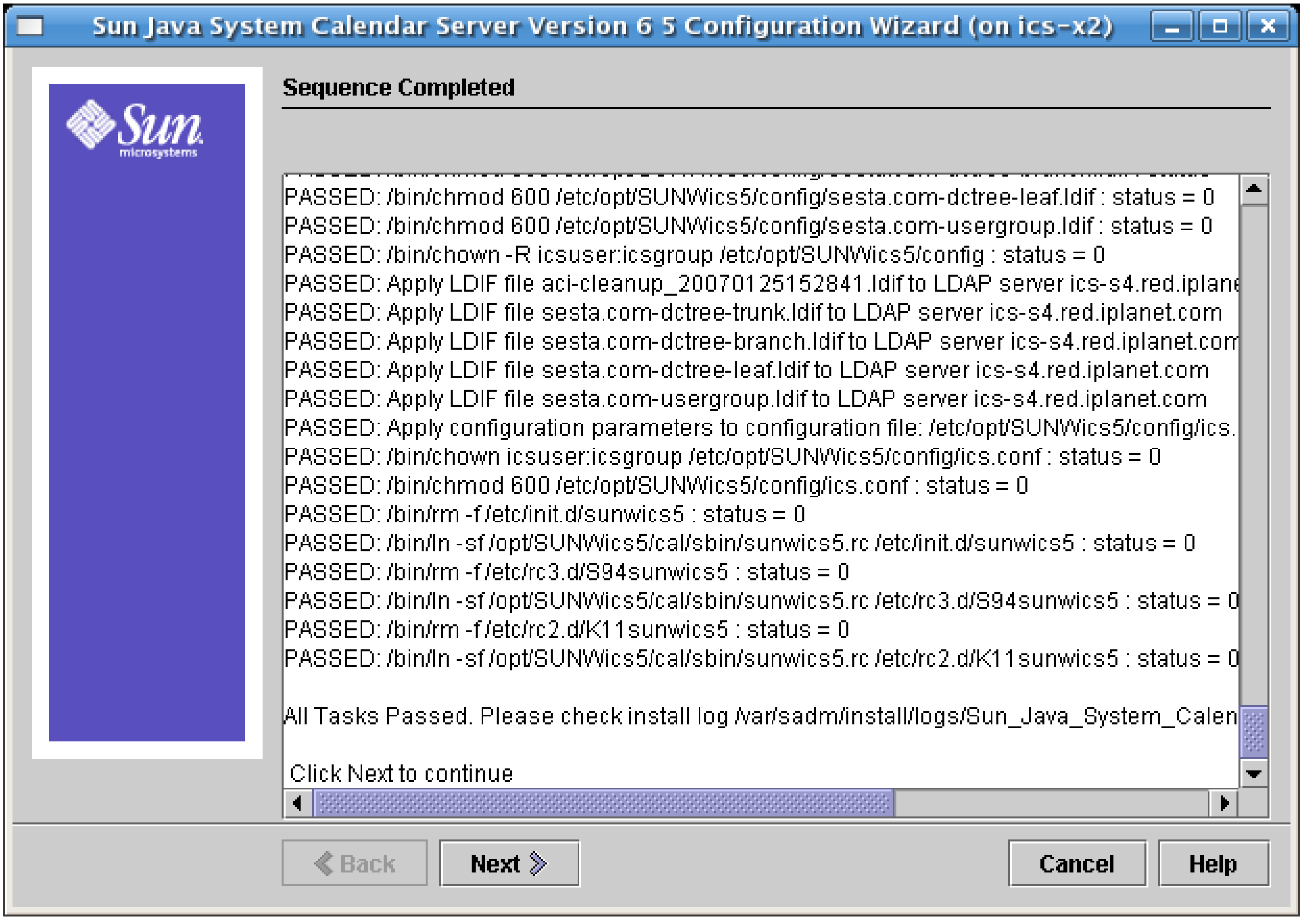
The following items for the product Sun Java System Calendar Server will be configured:

```
Product: Sun Java System Calendar Server
Location: /opt/SUNWics5
Space Required: 0 bytes
----------------------------------------
CalServer6
```

FIGURE 2–11 Ready to Configure Screen

Click Configure Now to configure Calendar Server, Back to return to the previous screen, or Cancel to exit.

2.12 Sequence Completed Screen for Calendar Server 6.3


Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)
Sun microsystems
Sequence Completed
Back
Next
Cancel
Help

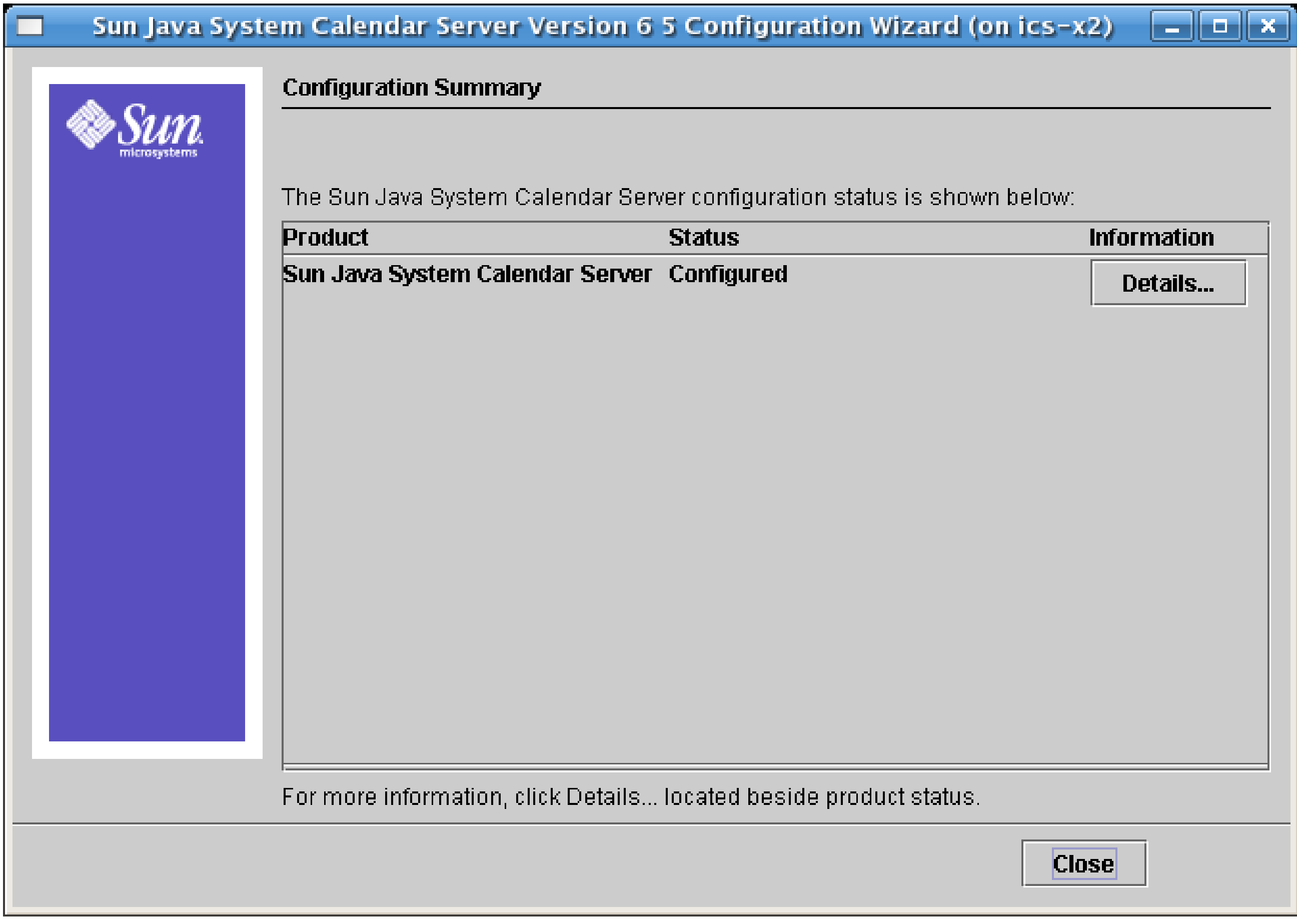
```
PASSED: /bin/chmod 600 /etc/opt/SUNWics5/config/sesta.com-dctree-leaf.ldif : status = 0
PASSED: /bin/chmod 600 /etc/opt/SUNWics5/config/sesta.com-usergroup.ldif : status = 0
PASSED: /bin/chown -R icsuser:icsgroup /etc/opt/SUNWics5/config : status = 0
PASSED: Apply LDIF file aci-cleanup_20070125152841.ldif to LDAP server ics-s4.red.iplane
PASSED: Apply LDIF file sesta.com-dctree-trunk.ldif to LDAP server ics-s4.red.iplanet.com
PASSED: Apply LDIF file sesta.com-dctree-branch.ldif to LDAP server ics-s4.red.iplanet.com
PASSED: Apply LDIF file sesta.com-dctree-leaf.ldif to LDAP server ics-s4.red.iplanet.com
PASSED: Apply LDIF file sesta.com-usergroup.ldif to LDAP server ics-s4.red.iplanet.com
PASSED: Apply configuration parameters to configuration file: /etc/opt/SUNWics5/config/ics.
PASSED: /bin/chown icsuser:icsgroup /etc/opt/SUNWics5/config/ics.conf : status = 0
PASSED: /bin/chmod 600 /etc/opt/SUNWics5/config/ics.conf : status = 0
PASSED: /bin/rm -f /etc/init.d/sunwics5 : status = 0
PASSED: /bin/ln -sf /opt/SUNWics5/cal/sbin/sunwics5.rc /etc/init.d/sunwics5 : status = 0
PASSED: /bin/rm -f /etc/rc3.d/S94sunwics5 : status = 0
PASSED: /bin/ln -sf /opt/SUNWics5/cal/sbin/sunwics5.rc /etc/rc3.d/S94sunwics5 : status = 0
PASSED: /bin/rm -f /etc/rc2.d/K11sunwics5 : status = 0
PASSED: /bin/ln -sf /opt/SUNWics5/cal/sbin/sunwics5.rc /etc/rc2.d/K11sunwics5 : status = 0

All Tasks Passed. Please check install log /var/sadm/install/logs/Sun_Java_System_Calen

 Click Next to continue
```

FIGURE 2–12 Sequence Completed Screen

This panel provides a running update of all the tasks and the disposition (passed or failed). When the message "All Tasks Passed", the configuration has finished. Check the log files indicated to see if there are any error messages.

Click Next when the configuration program completes.

2.13 Configuration Summary Screen for Calendar Server 6.3


Sun Java System Calendar Server Version 6 5 Configuration Wizard (on ics-x2)
Sun microsystems
Close

Configuration Summary

The Sun Java System Calendar Server configuration status is shown below:

Product	Status	Information
Sun Java System Calendar Server	Configured	Details...

For more information, click Details... located beside product status.

FIGURE 2–13 Configuration Summary Screen

Click Details to view the details of the configuration log or Close to exit the configuration program.

CHAPTER 3

Database Migration Utilities for Calendar Server 6.3

This chapter describes the various database migration utilities available for migrating your calendar databases and LDAP database after you have installed and configured Calendar Server 6.3 software.

This chapter contains the following sections:

Tip – If you are migrating from Calendar Server 6.0, 6.1, or 6.2 versions, run the utility called "3.3 csmigrate Utility" on page 94. If you did not already run `cs5migrate` for recurring events and tasks in your previous deployment, you must run `cs5migrate` on your existing calendar databases before running `csmigrate`.

If you are migrating from Calendar Server 5.1.1, migrate the calendar databases and the LDAP database using the migration utilities as explained in "3.2 Choosing the Right Calendar Server Utilities" on page 93.

If you had an even earlier version of Calendar Server installed, call technical support for assistance with migration of your data.

3.1 An Overview of Calendar Server Database Migration Utilities

This section describes each of the migration utilities. Use only the migration utilities you need, depending on which version of Calendar Server you previously had installed. These utilities are found in the `sbin` directory.

Tip – If you have ever run the `cs5migrate` utility against your databases, but did not use the *-r*option, you must run it again with the *-r* option before running any of the other utilities.

The migration utilities are as follows:

"3.5 csmig Utility" on page 96	Assigns an owner to each calendar in the Calendar Server 6 database and maps each calendar ID (`calid`) to an owner, if needed, which allows support for multiple domains and the LDAP Calendar Lookup Database (CLD) plug-in. Run this utility before `csvdmig`.
"3.6 csvdmig" on page 106	Upgrades a Calendar Server 6 site to use multiple domains by adding the calendar's domain (`@`*domainname*) to each `calid`. For example, in the domain `sesta.com`, the *jdoe* `calid`would now be `jdoe@sesta.com`. This utility is packaged with Calendar Server. Run this utility after`csmig` and before`cs5migrate`.
"3.4 cs5migrate Utility" on page 96	Migrates your calendar databases from Calendar Server version 5 to version 6.2 format. You must run this utility against your databases specifying the *-r* option. If you migrated from Calendar Server version 5.1.1 to version 6.2 prior to this time, but you did not run the `cs5migrate` utility with the *-r* option, you must run it with that option before running the `csmigrate` utility. Run this utility after `csmig` and `csvdmig` and before `csmigrate`.
"3.3 csmigrate Utility" on page 94	Migrates your calendar databases for upgrading from Calendar Server version 6.0, 6.1, or 6.2 to Calendar Server 6.3 version. If you need to run `cs5migrate` with the *-r* option, run it before this utility.

"3.7 commdirmig" on page 108	Migrates LDAP data from Schema version 1 to Schema version 2 in preparation for use with Access Manager (in Legacy mode).

3.2 Choosing the Right Calendar Server Utilities

This section helps you decide which utilities you need to run to have all calendar databases and your LDAP database at the Calendar Server 6.3 software level.

Use the following table to find the correct collection of utilities to run:

Note – Run the utilities in the order given.

TABLE 3–1 Choosing the Right Utilities

Calendar Server Version You Are Migrating From	Condition of Your Database Files	Utilities to Use
Calendar Server 6.0, 61. 6.2	You are using recurring events and tasks and have previously run `cs5migrate` in the past. You already use Schema version 2.	Run `csmigrate`
Calendar Server 6.0, 61. 6.2	You are using recurring events and tasks and have previously run `cs5migrate` in the past. You did not use Schema version 2 before, but need to now.	Run `csmigrate` and `commdirmig`.
Calendar Server 6.0, 61. 6.2	You have never run `cs5migrate` against your files. You already use Schema version 2, or you are on Schema version 1 and plan to stay with it.	Run `cs5migrate` and `csmigrate`.
Calendar Server 6.0, 61. 6.2	You have never run `cs5migrate` against your files. You did not use Schema version 2 before, but need to now.	Run `cs5migrate`, `csmigrate`, and `commdirmig`.
Calendar Server 5.1.1	You did not use multiple domains in the past.	Run `csmig`, `csvdmig`, `cs5migrate`, `csmigrate`, and `commdirmig`.

TABLE 3–1 Choosing the Right Utilities *(Continued)*

Calendar Server Version You Are Migrating From	Condition of Your Database Files	Utilities to Use
Earlier than Calendar Server 5.1.1	Your files do not support multiple domains or the LDAP CLD. Your LDAP database is using Schema version 1.	Call technical support for help getting your database and LDAP files to Calendar Server 5.1.1 level.
Earlier than Calendar Server 5.1.1	Your system is configured for limited virtual domains, or you have multiple instances of Calendar Server software installed on an operating system that predates Solaris 10.	Contact the sales account representative for an evaluation of your migration requirements.

3.3 csmigrate Utility

The `csmigrate` utility is used to migrate Calendar Server 6.0, 6.1 or 6.2 databases to Calendar Server 6.3 databases. You can find the `csmigrate` utility in the `sbin` directory of the Calendar Server product along with other administrative tools.

This section contains the following topics:

- "3.3.1 csmigrate Utility Syntax" on page 94
- "3.3.2 csmigrate Example" on page 95
- "How to Run the Calendar Server csmigrate Utility" on page 95

3.3.1 csmigrate Utility Syntax

The syntax for the `csmigrate` command is:

```
csmigrate [-q] [-d] [-l min|max] [-b backup_dir] source_dbdir target_dbdir
```

The options and their usage are as follows:

`-q` (optional) Specifies quiet mode and no print instructions.

`-d` (optional) Specifies dry run mode and no new database written.

`-l min|max` (optional) Specifies log level. The migration logs are written to `csmigrate.log` and errors are written to `csmigrateError.log` in the default logs directory.

`-b backup_dir` (optional) Specifies the directory to backup source database. The program backs up the source database to this directory and works on that copy to prevent any damage to the source databases. Default location is the `backup` under the source database directory.

`-source_dbdir` (mandatory) The directory where pre-migration database files are located.

`-target_dbdir` (mandatory) The directory where post-migration files are created.

`-V` (other supported option) To print the version information of the tool.

`-?` (other supported option) To print the usage information of the tool.

Note – The exit codes for the program are 255 on failure and 0 on success.

3.3.2 csmigrate Example

Examples of using the options in `csmigrate` command are:

```
csmigrate -b /var/opt/SUNWics5/tmpdb /var/opt/SUNWics5/old_db /var/opt/SUNWics5/new_db
csmigrate -q /var/opt/SUNWics5/old_db /var/opt/SUNWics5/new_db
csmigrate -l min old_db /var/opt/SUNWics5/new_db
csmigrate -l max old_db /var/opt/SUNWics5/new_db
```

▼ How to Run the Calendar Server csmigrate Utility

1 **Log in with root privileges.**

2 **Stop all services.**

For example, issue the following command:

```
stop-cal
```

3 **Move your current databases to a temporary directory.**

For example, move the entire `csdb` directory to the `oldcsdb`.

```
mv cal-svr-base/SUNWics5/csdb/* cal-svr-base/SUNWics5/oldcsdb
```

4 **Make sure both the new directory and old files in that directory are owned by the default administrator (icsuser, icsgroup).**

If the ownership is not correct, change ownership using the following command:

```
chown -R icsuser:icsgroup /cal-svr-base/SUNWics5/oldcsdb/
```

5. **Run the migration tool.**

   Migrate from your new backup copy (`oldcsdb`) to the `csdb` directory as shown in the following example:

```
cd cal-svr-base/SUNWics5/cal/sbin/
./csmigrate -l max /cal-svr-base/SUNWics5/oldcsdb cal-svr-base/SUNWics5/csdb
```

6. **Restart calendar services.**

   For example, use the following command:

```
stop-cal
```

3.4 cs5migrate Utility

The `cs5migrate` utility is used to migrate the Calendar Server 5.1.1 databases to Calendar Server 6.3 level. In addition, run this utility if you are migrating from one of the earlier Calendar Server 6 versions, and you did not use the recurring option.

The `cs5migrate` utility performs the following tasks:

Note – In the past, if you did not plan to use the Connector for Microsoft Outlook, you could choose to run this utility without doing the recurring data conversion. However, starting with Calendar Server 6.3, you must convert your recurring data to the new format.

This utility can be found in the `sbin` directory along with other administrative tools, after you have upgraded to Calendar Server 6.3 software.

3.5 csmig Utility

The `csmig` utility assigns an owner to each calendar in the calendar database and maps each calendar ID (`calid`) to an owner, if needed.

The `csmig` utility supports multiple domains and the LDAP Calendar Lookup Database (CLD) plug-in. Calendars in the migrated database are accessible using the LDAP CLD plug-in. For information about the LDAP CLD plug-in, see Chapter 5, “Configuring Calendar Database Distribution Across Multiple Machines in Calendar Server Version 6.3.”

This section describes the following topics:

- “3.5.1 csmig Utility Functions” on page 97
- “3.5.2 csmig Utility Requirements” on page 98
- “3.5.3 csmig Syntax” on page 98

3.5.1 csmig Utility Functions

The csmig migration utility performs the following functions:

3.5.1.1 Migrates Calendars

csmig migrates both user and resource calendars in the current calendar database (*.db files) specified by the caldb.berkeleydb.homedir.path parameter. In the new destination target database, csmig updates entries required by the LDAP CLD plug-in in the calendar properties (calprops), events, todos (tasks), and group scheduling engine (GSE) database files.

csmig writes only to the destination target database; it does not update your existing calendar database.

3.5.1.2 Assigns Owners to Calendars

csmig assigns an owner to each calendar in the calendar database and maps each calendar ID (calid) to an owner, if needed. All default calids are kept as is, and no changes are made.

Other calendars are mapped as follows:

- User calendars that don't have valid owners will be owned by the user passed to csmig by the -c option. For example, if calendar ID jsmith doesn't have an owner, it will be converted to orphan:jsmith, where orphan is specified as the -c option.
- Resource calendars that don't have an owner will be owned by the resource user passed to csmig by the -r option.
- If a resource calendar has any colons (:) in the name, the colons are converted to underscores, so that the migrated name has only one colon.

  For example, a calendar named football with owner bkamdar will be converted to bkamdar:football. A calendar named tchang:soccer with the owner bkamdar will be converted to bkamdar:tchang_soccer. A resource calendar named auditorium:room1 with an owner admin1 will be converted to admin1:auditorium_room1.

3.5.1.3 Updates LDAP Attributes

csmig updates LDAP attributes for all relevant LDAP entries, including icsSubscribed, icsCalendar, icsCalendarOwned, icsFreeBusy, icsSet, and for resource calendars, uid. csmig creates the icsDWPHost attribute for each calendar in the LDAP directory server database. icsDWPHost specifies the host name of the back-end server where a calendar resides.

3.5.2 csmig Utility Requirements

The requirements for using csmig are:

- The calendar database must not be corrupted. Use the csdb check command to check your calendar database, and if necessary, run the csdb rebuild command to rebuild the database. For information about these commands, Appendix D, "Calendar Server Command-Line Utilities Reference."
- You must have sufficient disk space for the new destination target database and if applicable, your backup database.
- To run csmig, log in as icsuser (or as the Calendar Server runtime user ID specified during configuration). If you run csmig as superuser (root), you might need to reset the permissions for the migrated files.

  You must also have privileges to manage the attributes of calendar users in the LDAP directory server that stores user preferences.
- Calendar Server must be stopped.

3.5.3 csmig Syntax

The csmig utility has the following syntax:

```
csmig [-t DestinationDB]
      [-b Backend-DWPHost]
      [-o OutputFile]
      [-e ErrorFile]
      [-m MappingFile]
      [-c calendarOwner]
      [-r resourceOwner]
      { migrate|dryrun }
```

The following table lists the utility options, gives a description of each, and gives the default value.

csmig Options	Description and Default Value
`-t` *DestinationDB*	Specifies the destination target database that `csmig` generates. The default is `MigratedDB`.
`-b` *Backend-DWPHost*	Specifies the name of the DWP back-end host server. This name must match the DWP back-end host server name specified in the `ics.conf` file.
`-o` *OutputFile*	Specifies an output file that captures the `csmig` output to the screen as well as any errors that occur. The default is `MigrateOut`.
`-e` *ErrorFile*	The file where `csmig` writes any errors or database entries that cannot be resolved. If database entries cannot be resolved, they are not written to the destination database. The default is `MigrateError`.
`-m` *MappingFile*	Specifies an output mapping file generated in `dryrun` mode that lists entries in the LDAP schema that need to be changed. For example: Old: `calid=jsmith` New: `calid=jsmith:basketball` The mapping file provides only a list of changes to make to the LDAP schema. csmig does not actually make the changes to the schema The mapping file is not used in `migrate` mode.
`-c` *calendarOwner*	Specifies the owner for user calendars that don't have owners.
`-r` *resourceOwner*	Specifies the owner for resource calendars that don't have owners.
`migrate|dryrun`	Specifies which mode the utility is running in. Use `migrate` mode to perform the migration. Use `dryrun` mode to generate the output mapping file before you actually migrate.

3.5.4 csmig Utility Migration Steps

If you had a version of Calendar Server predating version 5.1.1, after you install and configure Calendar Server 6.3, run `csmig` to migrate your existing Calendar Server and LDAP databases. Migration of the LDAP data is required for the LDAP CLD plug-in to work properly. Use these steps to migrate calendar data using `csmig`:

▼ High Level Steps for Using csmig

1. **Configure your Directory Server using** `comm_dssetup.pl`**.**

   If you have not already indexed LDAP attributes using `comm_dssetup.pl`, do so at this time. This will greatly help performance of the LDAP data migration.

2 **Using a staging server (not your production server), perform a test dry run.**

A dry run reports what csmig would do during an actual migration but does not migrate any data. After the dry run, and before you actually migrate, correct any errors and determine a plan to handle any unresolved calendars.

For instructions on how to perform a test dry run, see "3.5.4 csmig Utility Migration Steps" on page 99.

3 **Migrate Your Production Data**

During a production run, csmig migrates the calendar database (.db files) and LDAP data (user and group preferences data), icsSubscribed, icsCalendar, icsCalendarOwned, icsFreeBusy, icsSet, and uid (for resource calendars). After the migration, all calendar resources will have an LDAP entry created.

For instructions on how to migrate your production data, see "3.5.4 csmig Utility Migration Steps" on page 99.

▼ To Perform a Test Dry Run

1 **Install Calendar Server 6.3 (if necessary) on the staging server.**

2 **Copy a snapshot of your calendar database to the staging server.**

3 **Mimic your production LDAP environment on the staging server by performing the following tasks:**

- Install Directory Server.
- Install a snapshot of the LDAP database on this server.

4 **Run comm_dssetup.pl to configure the staging Directory Server.**

5 **Run csconfigurator.sh to configure the staging Calendar Server.**

6 **Log in as icsuser (or, if its different, log in as the Calendar Server runtime user ID specified during configuration). If you run csmig as superuser (root), you might need to reset the permissions for the migrated files.**

7 **Change to the *cal-svr-base*/SUNWics5/cal/sbin directory.**

8 **Run the csdb check command to check your database for corruption. If corruption is indicated, run csdb rebuild to rebuild the database.**

9 **Consider creating a catchall calid for user calendars that don't have an owner. For example, the following command creates a user with the calid of orphan:**

```
./csuser -g orphan -s adminuser -y password -l en -c orphan create orphan
```

10 Stop the Calendar Server using the `stop-cal` command (if necessary).

cal-svr-base/SUNWics5/cal/sbin/stop-cal

11 Run `csmig` with the `-dryrun` option. For example, you might enter:

```
./csmig -b sesta.com -o csmig.out -e csmig.errors
 -m csmig.map -c orphan -r calmaster dryrun
```

This command assigns user calendars without an owner (orphan calendars) to the owner `orphan` and resource calendars without an owner to the owner `calmaster`.

12 Check the output mapping file (`csmig.map`). The mapping file lists entries that need to be updated in the LDAP schema.

13 Check the output, mapping, and error files. Resolve any LDAP issues or errors that you find. Determine how you will handle any unresolved calendars before the actual migration.

Several options are:

- Delete any unneeded calendars before you migrate.
- Assign owners to any unresolved calendars.
- Allow `csmig` to assign owners to the calendars during migration using the `-c` and `-r` options.

14 Run `csmig` to migrate your staging calendar database.

For example, the following command migrates the calendar database to the `/var/opt/SUNWics5/testcsdb/` directory:

```
./csmig -t /var/opt/SUNWics5/testcsdb/ -b sesta.com
-o csmig.out -e csmig.errors -m csmig.map -c orphan
-r calmaster migrate
```

15 After the test migration is finished, perform these steps to check out the newly migrated calendar database.

a. **Copy the migrated database to the `/csdb` directory specified by the** *caldb.berkeleydb.homedir.path* **parameter. Or, edit this parameter to point to the new location of the migrated database.**

b. **Run `csdb check` on the new calendar database. The number of events and todos in the migrated database should match the pre-migration totals.**

c. **Search for `icsCalendarOwned` entries and make sure that the entries match the pre-migration number of calendars.**

d. **Log in to Communications Express and verify some of the calendars in the migrated database.**

   If the test migration is successful, you are ready to migrate your production database.

▼ To Migrate Your Production Data

1 **Log in as `icsuser` (or as the Calendar Server runtime user ID specified during configuration). If you run `csmig` as superuser (*root*), you might need to reset the permissions for the migrated files.**

2 **Change to the *cal-svr-base*`/SUNWics5/cal/sbin` directory.**

3 **Stop the Calendar Server using the `stop-cal` command.**

cal-svr-base`/SUNWics5/cal/sbin/stop-cal`

4 **Backup the following data:**

- Calendar database (`.db` files).
- LDAP data: `slapd` database directory and LDAP database.
- `ics.conf` file. This step is not actually required, but it can be useful if you need to revert to your original configuration.

5 **Run `csmig` with the `-migrate` option.**

For example, the following command migrates the calendar database to the `/var/opt/SUNWics5/newcsdb/` directory:

```
./csmig -t /var/opt/SUNWics5/newcsdb/ -b sesta.com
-o csmig.out -e csmig.errors -m csmig.log -c orphan
-r calmaster migrate
```

6 **Check for any unresolved calendars in the error file (`csmig.errors`) and resolve them according to your plan from "3.5.4 csmig Utility Migration Steps" on page 99 under "3.5.4 csmig Utility Migration Steps" on page 99.**

7 **Run the `csdb check` command to check your migrated database. If any corruption is indicated, run `csdb rebuild` to rebuild the database.**

8 **Copy the new migrated database to the `/csdb` directory specified by the *caldb.berkeleydb.homedir.path* parameter. Or, edit this parameter to point to the new location of the migrated database.**

9 **Enable the LDAP CLD plug-in by making any necessary changes to the following configuration parameters in the `ics.conf` file:**

- *service.dwp.enable* = "yes"
- *service.dwp.port* = "59779"
- *csapi.plugin.calendarlookup* = "yes"
- *csapi.plugin.calendarlookup.name* = "*"
- *caldb.cld.type* = "directory"
- *caldb.dwp.server.default* = "default-server-name"
- *caldb.dwp.server.server-hostname*.ip = "*server-hostname*" (for each back-end server including the local server)
- *caldb.cld.cache.enable* = "yes" (if you want to use the CLD cache option)
- *caldb.cld.cache.homedir.path* specifies the location of the CLD cache directory. The default is /var/opt/SUNWics5/csdb/cld_cache.

  For information about setting configuration parameters for the LDAP CLD plug-in, see Chapter 5, "Configuring Calendar Database Distribution Across Multiple Machines in Calendar Server Version 6.3."

10 **Restart Calendar Server using the** start-cal **command.**

11 **Log in to Communications Express and verify that your configuration is working by checking several of the migrated calendars.**

To disable alarms while you are making your checks, set each of the following parameters in the ics.conf file to "no":

- *caldb.serveralarms* = "no"
- *caldb.serveralarms.dispatch* = "no"
- *service.ens.enable* = "no"
- *service.notify.enable* = "no"
- *ine.cancellation.enable* = "no"
- *ine.invitation.enable* = "no"
- *service.admin.alarm* = "no"

3.5.5 csmig Tips and Troubleshooting

The section describes the following tips and trouble shooting examples:

- "3.5.5.1 The csmig dry run calendar shows the wrong owner for a calendar." on page 104
- "3.5.5.2 The LDAP calendar search doesn't work correctly." on page 104
- "3.5.5.3 The csmig dry run indicates duplicate calendar names." on page 105
- "3.5.5.4 How do I assign orphan calendars to different owners?" on page 105
- "3.5.5.5 How do I move calendar users to another back-end server?" on page 105

3.5.5.1 The csmig dry run calendar shows the wrong owner for a calendar.

Example Problem

A calendar named `tchang:myCalendar` has the owner `jsmith` in the calendar database, and the `csmig` dry run shows the mapping as `jsmith:tchang_myCalendar`. However, you would like to name this calendar `tchang:myCalendar` and assign the owner as `tchang`.

Example Solution

Before the migration, use the `cscal` utility to change the owner of the calendar `tchang:myCalendar` to `tchang`. Once this is done, the migration will map this calendar to `tchang:myCalendar` and add `icsCalendarowned` to the LDAP entry for user ID `tchang`.

3.5.5.2 The LDAP calendar search doesn't work correctly.

Example Problem

After migration, the LDAP calendar search is enabled, but the calendar search dialog does not return any results or returns only partial results.

Example Solution

Enabling the LDAP calendar search allows Calendar Server to search `(&(objectclass=icscalendaruser)(icscalendarowned=*substr*))`.

Manually run two different searches on the LDAP data with the following filters and compare the output:

- LDAP search with filter `(&(objectclass=icscalendaruser)(icscalendarowned=*substr*))`
- LDAP search with filter `(icscalendarowned=*substr*)`

Since the server uses the filter that includes `icsCalendarUser` object class, the LDAP server might have been deployed with the schema check disabled, and some calendar entries may have been provisioned without the `icsCalendarUser` object class.

3.5.5.3 The csmig dry run indicates duplicate calendar names.

Example Problem

The `csmig` dry run mapping file and output file indicate that there is a duplicate calendar name.

For example, in the original database, `jsmith` owns the following calendars:

- `basketball` with 5 events
- `jsmith:basketball` with 10 events

The dry run indicates that during a migration, the two calendars will be merged, and the resulting calendar will be `jsmith:basketball` with owner `jsmith` and 15 total events

The output file will include the following warning message:

```
Error modifying calendar properties, error=2
```

Example Solution

If you don't want the two calendars to be merged, change the owner of `basketball` to a user other than `jsmith` before the migration. This will preserve the data integrity of the two separate calendars.

3.5.5.4 How do I assign orphan calendars to different owners?

Example Problem

By default `csmig` assigns all orphan calendars to a single owner, but I would like to assign different owners for some orphan calendars.

Example Solution

`csmig` does not accept the mapping file in the command line. However, you can assign owners to the orphan calendars in the original database before the migration. Check the dry run mapping file for all orphan calendars. Then use the `cscal` utility to assign owners to the orphan calendars before the migration. Run `csmig` in `-dryrun` mode again to verify the new owners.

3.5.5.5 How do I move calendar users to another back-end server?

Example Problem

How do I move users from one back-end server to another?

Example Solution

To move a calendar user, you `export` each of the user's calendars on the original server and then `import` the calendars on the second server. After the calendars are moved, you can delete the calendars on the original server. For instructions on how to move calendars, see "15.6 Managing User Calendars" on page 304.

3.6 csvdmig

The `csvdmig` utility prepares your calendar databases and LDAP user and group entries for use in a multiple-domain environment. Even if you plan to use only the default domain, you must run this utility.

Note – Be sure to run `csmig` before using this utility if you are migrating from a non-domain environment to the multiple domain environment in Calendar Server 6.3.

This sections contains the following topics:

- "3.6.1 csvdmig Functions" on page 106
- "3.6.2 csvdmig Syntax" on page 107
- "3.6.3 csvdmig Examples" on page 108

3.6.1 csvdmig Functions

The `csvdmig`utility performs the following changes to your databases and LDAP entries:

- The format of calendar IDs (`calids`) is changed:

  From: `userid[:calendar-name]`

  To: `userid@domain[:calendar-name]`
- Access Control List (ACL) access rules are changed:

  From: `userid`

  To: `userid@domain`
- The LDAP directory server user entries for the Calendar Server attributes are modified as:

  `userid[:calendar-name]` to `userid@domain[:calendar-name]`.
- Updates the owner and attendee fields in events and tasks in the calendar database. For example:

  If `jsmith` in the domain `sesta.com` is the owner of an event, the new owner field would contain `jsmith@sesta.com`.

Caution – The `csvdmig` utility updates the databases and LDAP directory in place. That is, it does not create a separate migrated database, but alters the database you are converting. Therefore, to be safe, run `csvdmig` against snapshots of your databases and LDAP directory.

3.6.2 csvdmig Syntax

The `csvdmig` utility has the following syntax:

```
csvdmig [-t DestinationDB]
         [-c ConfigFile]
         [-e ErrorFile]
         [-m MappingFile]
         migrate [DB|LDAP]
```

The following table lists the options used by `csvdmig`, and gives a description of each.

Option	Description and Default Value
`-m` *MappingFile*	Input parameter specifying a mapping file. For more information on the mapping file, see "3.6.2.1 Mapping File" on page 107. The default is `MigrateMapping`.
`-c` *ConfigFile*	Input parameter that specifies a Calendar Server configuration file. The default is the `ics.conf` file.
`-t` *DestinationDB*	Output parameter that specifies the location of the database to be migrated. The default is `MigratedDB`. **Tip –** Always use the `-t` option. For more information on this option, see "3.6.2.2 Destination DB" on page 108.
`-e` *ErrorFile*	output parameter that specifies the name of the error file for errors that cannot be resolved. The default is `MigrateError`.
`DB` | `LDAP`	Specifies which database to modify: `DB` – the calendar database `LDAP` – the LDAP directory The default is the calendar database (`DB`).

3.6.2.1 Mapping File

The mapping file is an input text file that maps existing users to their respective domains. You must create the mapping file before you run `csvdmig`. Specify one entry per line with a space between the old and new values. For example:

```
user1 user1@sesta.com
user2 user2@siroe.com
user3 user3@sesta.com
 ...
usern usern@siroe.com
```

3.6.2.2 Destination DB

The location of the database to be migrated. The utility updates the file in place. Be sure you have backed up this directory before using the csvdmig utility.

If you do not specify the -t option, the utility will attempt to migrate the contents of the current directory (the directory specified by performing pwd at the command line), with unpredictable results.

3.6.3 csvdmig Examples

The following are csvdmig examples

- Migrate the LDAP directory server data using default values:

  ```
  csvdmig migrate LDAP
  ```

- Migrate the Calendar Server database:

  ```
  csvdmig -t targetDB -e errorFile -m mappingFile migrate
  ```

3.7 commdirmig

The commdirmig utility migrates your LDAP data from Sun Java System LDAP Schema version 1 to Schema version 2 in preparation for using Access Manager for authentication services. If your previous installation already used Schema version 2, you do not have to run this utility again.

3.7.1 Who Should Run the commdirmig Utility

This migration utility migrates your Schema version 1 LDAP database to Schema version 2. If you are going to use Access Manager software for authentication, you must convert your LDAP entries to Schema version 2 format by running this utility.

If you are not using Access Manager, you should still consider migrating your LDAP data, since Schema version 2 is the preferred LDAP mode for all Communications Suite products that use LDAP.

Note – If you have a separate LDAP directory for preferences, you must run `commdirmig` on that LDAP as well as the one used for authentication.

3.7.2 When to Run the commdirmig Utility

Run `commdirmig` after you have run all the other migration utilities necessary to migrate your calendar and LDAP databases from your earlier version of Calendar Server software to the 6.3 version of the Calendar Server software.

3.7.3 Where to Find Documentation on the commdirmig Utility

The `commdirmig` migration utility requires special preparation and planning. It is documented in a separate guide, see *Sun Java Communications Suite 5 Schema Migration Guide*.

3.7.4 Where to Find the Utility

The `commdirmig` utility comes bundled with Delegated Administrator that you install from the Communications Suite installer.

A patch is also available from technical support for the utility.

PART III

Customizing Your Calendar Server Configuration

This part contains chapters on various features that can be configured by editing the configuration file, `ics.conf`.

This part contains the following chapters:

CHAPTER 4

Customizing Calendar Server

After installation and post installation configuration, Calendar Server can be run as is. However, you can customize features in your system (reconfigure your system) by editing the `ics.conf` file.

Note – Duplicate parameters are allowed in the `ics.conf` file. The system takes the value of the last instance of the parameter in the file.

Best Practices: To avoid confusion, add your customizations to the end of the file in a section you create for that purpose. For example, you can create a comment line with the following text: **`! My ics.conf Changes`**. Then add any new parameters or any parameters that you are modifying, and their values. Add comments to each parameter describing why the change was made and add the current date. This will give you a history of the changes made to the system for later reference.

If you make extensive customizations, to improve processing efficiency, you might consider commenting out the original parameter that you are replacing. Also, periodically review the file, commenting out obsolete duplicate parameters.

This chapter, and the chapters that follow in Part III, contain instructions and information you can use to reconfigure Calendar Server.

You can find the `ics.conf` file in the following directory:

For Solaris: `/etc/opt/SUNWics5/cal/config`

For Linux: `/etc/opt/sun/calendar/config`

Note –

Do not attempt to edit the configuration file until you have completed the following tasks:

- Install or upgrade to Calendar Server 6 2006Q3.
- Run the post installation configuration programs `comm_dssetup.pl` and `csconfigurator.sh`.
- Run `csmig`, `csvdmig`, `cs5migrate`, `csmigrate`, and `commdirmig` as needed against your existing calendar databases. See Chapter 3, "Database Migration Utilities for Calendar Server 6.3."

This chapter covers the following configuration topics:

4.1 Configuring for Communications Express

This section covers the two configuration file parameters to configure for Communications Express.

Communications Express requires the following:

- "To Configure Proxy Authentication" on page 114
- "To Enable Anonymous Access" on page 115

▼ To Configure Proxy Authentication

1. **Log in as an administrator with permission to change the configuration.**

2. **Stop Calendar Server services by issuing the** `stop-cal`.

3. **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4. **Save your old** `ics.conf` **file by copying and renaming it.**

5. **Edit the** `ics.conf` **parameters as shown in the following table:**

service.http.allowadminproxy Enables administrator proxy authentication when set to "`yes`". The default is "`yes`".

service.http.admins Lists the user ID's with administration rights to Calendar Server. The default is “calmaster”. This can be a space-separated list with multiple values. One of the values must be the value as specified in the `uwconfig.properties` file for *calendar.wcap.adminid*.

service.siteadmin.userid User ID of the calmaster. This should be the same as the user ID found in the *calendar.wcap.adminid* parameter of the `uwcconfig.properties` file.

service.siteadmin.cred Password for the calmaster. This should be the same as the user ID found in the *calendar.wcap.passwd* parameter of the `uwcconfig.properties` file.

Note – The `uwcconfig.properties` file is located in the *comms-express-svr-base*`/WEB-INF/config` directory, where *comm-express-svr-base* is the directory where Communications Express was installed.

---

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

See Also For instructions on configuring Communications Express, see the*Sun Java System Communications Express 6.3 Customization Guide* .

To Enable Anonymous Access

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal`.**

3 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4 **Save your old `ics.conf` file by copying and renaming it.**

5 **Edit the following parameters in the `ics.conf` to enable anonymous access:**

service.wcap.anonymous.allowpubliccalendarwrite
Enables or disables allowing anonymous access users to write to public calendars. Enable access by setting the value to "`yes`", which is the default.

service.wcap.allowpublicwritablecalendars
Enables users to have publicly writable calendars. This is enabled by default (set to "`yes`").

service.http.allowanonymouslogin
Enable anonymous access (login) by setting this parameter to "`yes`", if necessary. The default value is "`yes`".

service.calendarsearch.ldap
For security purposes with anonymous logins enabled, you might want to disable searching through the LDAP first when doing calendar searches, by setting this parameter to "`no`", which is the default.

Note – Communications Express expects the value of the *service.calendarsearch.ldap* parameter to be "`no`". This conflicts with instructions given for tuning your system for best performance in a DWP environment, (in which your database is distributed across multiple back-ends.) See "21.2 Improving Calendar Search Performance in a DWP Environment" on page 350.

---

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

For instructions on configuring Communications Express, see the*Sun Java System Communications Express 6.3 Administration Guide*.

4.2 Configuring Calendars

This section contains the following topics:

- "To Configure User Calendars" on page 116
- "To Configure Resource Calendars" on page 118
- "To Configure Group Calendars" on page 120
- "To Disable Autoprovisioning of Calendars" on page 121
- "To Configure Free-Busy Lookup" on page 122

▼ To Configure User Calendars

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the** `stop-cal`**.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit one or more of the parameters as shown in the following table:**

calstore.calendar.default.acl	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. The default is:
	`"@@o^a^r^g;@@o^c^wdeic^g;`
	`@^a^fs^g;@^c^^g;@^p^r^g"`
	For more information on the ACE format, see “15.4 Calendar Access Control” on page 296 Calendar Server utilities, see “D.5 `cscal`” on page 409.

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"`
calstore.freebusy.include.defaultcalendar	Specifies whether a user's default calendar is included in user's free/busy calendar list. The default is “`yes`”.
calstore.freebusy.remove.defaultcalendar	Specifies whether a user's default calendar can be removed from user's free/busy calendar list. The default is “`no`”.
service.wcap.freebusy.redirecturl	Specifies a URL to use to search for a calendar in a different database. This is only used while migrating calendar databases. During the time that calendars are split between two different databases, you can specify a URL other than the current Calendar Server database. The system searches the Calendar Server calendar database first and if it can’t find the user, checks to see if the redirect URL is available. This feature can be turned off by passing in the `noredirect` parameter set to `1` with the `get_freebusy` command.

calstore.subscribed.include.defaultcalendar

Specifies whether a user's default calendar is included in the user's subscribed calendar list. The default is “yes”.

service.wcap.login.calendar.publicread

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. The default is “no”.

user.allow.doublebook

Determines if a user calendar can have more than one event scheduled for the same time period:

- "no" prevents double booking.
- "yes" allows double booking, and is the default.

  This parameter is used only when a user calendar is created. Thereafter, Calendar Server checks the calendar properties file (`ics50calprops.db`) to determine if double booking is allowed.

  To change the value of the double booking calendar property, use `cscal` with the `-k` option.

user.invite.autoprovision

Determines if user calendar should be auto-created if the user receives an invitation but has no default calendar. The default is the enable this option (*”yes”*).

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure Resource Calendars

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the** `stop-cal`**.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit one or more of the parameters as shown in the following table:**

resource.allow.doublebook

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, and is the default.
- "`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

Specifies the default access control permissions used when a resource calendar is created. The default is:

```
"@@o^a^r^g;@@o^c^wdeic^g;
@^a^rsf^g"
```

resource.invite.autoaccept

When invitations are sent to resources should they be automatically marked as accepted? The default is "`yes`".

resource.invite.autoprovision

When a resource is invited to an event, if it has no existing calendar, should it be autoprovisioned?

The default is "`yes`".

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure Group Calendars

A group calendar can be scheduled with events similar to a user calendar. However, a user should not log into a group calendar. To view a group calendar, the user should subscribe to it. To configure a group calendar, edit the `ics.conf` file as shown in the steps that follow.

1. **Log in as an administrator with permission to change the configuration.**

2. **Stop Calendar Server services by issuing the** `stop-cal`**.**

3. **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4. **Save your old** `ics.conf` **file by copying and renaming it.**

5. **Edit one or more of the parameters as shown in the following table:**

group.allow.doublebook	Specifies whether the group calendar can be double booked. The default is `yes`.
group.default.acl	Specifies the default ACL for group calendars: `"@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g"`
group.invite.autoprovision	Specifies whether autoprovisioning is enabled or disabled. The default is `"yes"` (enabled).
group.invite.autoaccept	Specifies whether a group invitation will automatically have `PARTSTAT=ACCEPTED`.
group.invite.expand	Determines if a group should be expanded for invitations. If “`yes`”, the list will be expanded if it meets the constraints of the `calstore.group.attendee.maxsize` parameter. If the expansion fails, or this parameter is set to `"no"`, only the group name shows up on the attendee list and no RSVP is required.
calstore.group.attendee.maxsize	Specifies whether groups can be expanded. A value of `"0"` means no expansion limits. A group of any size can be expanded. If expansion is allowed, but not unlimited. The value of the parameter indicates the maximum number of attendees allowed in an expanded group. If the number in the group exceeds the maximum size, then the group is not expanded. A value of `"-1"` means no expansion allowed.

If expansion is not allowed because it exceeds the maximum size, only the group name appears in the attendee list and an error is returned to the organizer.

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

See Also For instructions on configuring groups, see "To Configure Calendar Server for Groups" on page 125.

▼ To Disable Autoprovisioning of Calendars

Autoprovisioning of user , resource and group calendars is enabled by default. That is, if a user attempting to log in does not yet have a default calendar, the system creates a user calendar with default settings.

If a user, resource or a group is invited to an event, but it does not yet have a default calendar, the system creates a resource or group calendar with default settings.

If you want to disable any of these calendars from being autoprovisioned, change the appropriate parameter in the `ics.conf` file as shown in the steps that follow.

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal`.**

3 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4 **Save your old `ics.conf` file by copying and renaming it.**

5 **Disable autoprovisioning of user, resource and group calendars by editing the following parameters:**

local.autoprovision Specifies whether autoprovisioning of user calendars is enabled (`"yes"`), or disabled (`"no"`). The default is `"yes"`.

resource.invite.autoprovision Specifies whether autoprovisioning of resource calendars is enabled (`"yes"`), or disabled (`"no"`). The default is `"yes"`.

group.invite.autoprovision Specifies whether autoprovisioning of group calendars is enabled (`"yes"`), or disabled (`"no"`). The default is `"yes"`.

autoprovisioning	Specifies whether auto-inviting of user calendars is enabled (“`yes`”), or disabled (“`no`”). The default is “`yes`”.

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure Free-Busy Lookup

The free-busy view is used for several purposes. There are a number of `ics.conf` parameters that can be set to customize how the free-busy view is generated.

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the** `stop-cal`**.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit one or more of the following** `ics.conf` **parameters shown in the following table:**

service.wcap.freebusybegin	Specifies the offset from the current time in days for `get_freebusy` for beginning of the range. The default is "`30`".
service.wcap.freebusyend	Specifies the offset from the current time in days for `get_freebusy` for end of the range. The default is "`30`".
calstore.freebusy.include.defaultcalendar	Specifies whether a user's default calendar is included in user's free/busy calendar list. The default is "`yes`".
calstore.freebusy.remove.defaultcalendar	Specifies whether a user's default calendar can be removed from user's free/busy calendar list. The default is "`no`".

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

4.3 Configuring Calendar for LDAP Users, Groups and Resources

This section contains instructions on configuring LDAP users, groups and resources.

This section includes the following topics:

- "To Configure Calendar Users" on page 123
- "To Set Calendar User Preferences" on page 124
- "To Configure Calendar Server for Groups" on page 125

To Configure Calendar Users

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the following `ics.conf` parameters shown in the following table:**

local.lookupldapsearchattr.aclgroup
The attribute used to specify which groups a user, group, or resource is a member of, for ACL evaluation. The default is "`aclgroupaddr`". (This is used to calculate dynamic groups.)

service.wcap.allowchangepassword
If "`yes`", allow users to change their passwords. The default is "`no`".

service.wcap.allowpublicwritablecalendars
If "`yes`", allow users to have publicly writable calendars. The default is "`yes`".

calstore.subscribed.remove.defaultcalendar
Specifies whether a user's default calendar can be removed from the user's subscribed calendar list. The default is "`no`".

service.wcap.allowcreatecalendars
If "`yes`", allow calendars to be created by users who do not have administrative privileges. The default is "`yes`".

service.wcap.allowdeletecalendars
If "`yes`", allow calendars to be deleted by users who do not have administrative privileges, but do have delete permission for that calendar. The default is "`yes`".

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Set Calendar User Preferences

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the following `ics.conf` parameters shown in the following table:**

service.wcap.allowsetprefs.cn	If "`yes`", allow `set_userprefs` to modify the user preference "cn" (LDAP user's common name). The default is "no".
service.wcap.allowsetprefs.givenname	If "`yes`", allow `set_userprefs` to modify the user preference "`givenname`" (LDAP user's given name). The default is "`no`".
service.wcap.allowsetprefs.icsCalendar	If "`yes`", allow `set_userprefs` to modify the user preference "`icsCalendar`" (a user's default calendar identifier). The default is "`no`".
service.wcap.allowsetprefs.mail	If "`yes`", allow `set_userprefs` to modify the user preference "mail" (user's email address). The default is "`no`".
service.wcap.allowsetprefs.preferredlanguage	If "`yes",` allow `set_userprefs` to modify the user preference "`preferredlanguage`" (LDAP user's preferred language). The default is "`no`".
service.wcap.allowsetprefs.sn	If "`yes`", allow `set_userprefs` to modify the user preference "sn" (LDAP user's surname). The default is "`no`".
service.wcap.userprefs.ldapproxyauth	If "`yes`", enables LDAP proxy authorization for `get_userprefs`. If "no", anonymous LDAP search is performed. The default is "`no`".

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

▼ To Configure Calendar Server for Groups

Calendar Server supports LDAP groups, which are a named collection of users. The group membership can be static, or dynamically created. Groups can be nested. Groups have a `groupid` that is analogous to a `uid` for a user. Groups also have a mail address.

In addition, groups can have a default calendar with a group `calid` that should correspond to the `groupid`, with the addition of the domain, for instance `groupid@sesta.com`. Group calendars do not have user interface preferences stored in the preferences database. Instead, the LDAP entry contains an `icsDefaultacl` attribute that is used in group creation.

A group is defined in the LDAP entry as an instance of `icsCalendarGroup`. For information on the other attributes available for group calendars, see the *Sun Java System Communications Services 6 2005Q4 Schema Reference*.

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Edit one or more of the following** `ics.conf` **parameters shown in the following table:**

local.lookupldapsearchattr.owner Owner attribute to use for groups and resources. The default is "`owner`".

local.lookupldapsearchattr.coowner Secondary owners attribute for groups and resources. The default is "`icsSecondaryowners`".

local.lookupldapsearchattr.groupid The attribute used to store the unique group identifier. The default is "`groupid`".

local.lookupldapsearchattr.defaultacl The attribute used to store the default ACL given to each group calendar at autoprovisioning. The default is "`icsDefaultacl`".

local.lookupldapsearchattr.doublebook The attribute used to specify whether doublebooking of group calendars is permitted. This is the attribute used when a default group calendar is auto-created. The default is "`icsDoublebooking`".

local.lookupldapsearchattr.autoaccept The attribute used to specify whether invitations to group calendars are automatically accepted. This is

	the attribute used when a default group calendar is auto-created. The default is `"icsAutoaccept"`.
local.lookupldapsearchattr.timezone	The attribute used to specify the time zone for an auto-created group calendar. The default is `"icsTimezone"`.
local.lookupldapsearchattr.aclgroup	The attribute used to specify which groups a user, group, or resource is a member of, for ACL evaluation. The default is `"aclgroupaddr"`. (For groups, this would be for nested groups.)

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

See Also If you plan to have calendars for groups, you need to configure group calendars. See "To Configure Group Calendars" on page 120.

If you are using groups, you should set the following domain level preferences in the group LDAP entry:

- *icsAllowRights* — Set bit 15 to indicate your domain-level preference for group calendar doublebooking.
- *icsExtendedDomainPrefs* — Set the `groupdefaultacl` property to determine the default ACL for group calendars in the domain.

For information on how to configure Calendar Server domains for groups, see "11.1 Configuring Domain Preferences for Groups in Calendar Server Version 6.3" on page 243.

4.4 Configuring Calendar Server

This section contains procedures for customizing server-side configuration by editing the `ics.conf` file.

This section contains the following topics:

- "To Configure Server Behavior" on page 127
- "To Configure Calendar Logging" on page 128
- "To Configure WCAP Commands" on page 130
- "To Enable Email Notifications" on page 131

▼ To Configure Server Behavior

The calendar store is configured by default as shown in The following table. If you wish to reconfigure the store, perform the following steps:

1. **Log in as an administrator with permission to change the configuration.**
2. **Stop Calendar Server services using `stop-cal`.**
3. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**
4. **Save your old `ics.conf` file by copying and renaming it.**
5. **Edit one or more of the parameters in the following table:**

Parameter	Description and Default Value
calstore.calendar.create.lowercase	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. The default is "`no`".
calstore.default.timezoneID	Time zone ID to be used when importing files, and no other time zone ID's can be found for any of the following: an event, a calendar, a user. The default is "`America/New_York`" An invalid value causes the server to use the GMT (Greenwich Mean Time) time zone.
calstore.filterprivateevents	Specifies whether Calendar Server filters (recognizes) Private and Confidential (Time-and-Date-Only) events and tasks.If "`no`", Calendar Server treats them the same as Public events and tasks. The default is "`yes`".
calstore.group.attendee.maxsize	Maximum number of members allowed when expanding a group. The default value, "`0`" means to expand the group without regard to size. A value of `-1` means no expansion of groups.
calstore.recurrence.bound	Maximum number of events that can be created by a recurrence expansion. The default is "`60`".
calstore.userlookup.maxsize	Maximum number of results returned from LDAP lookup from user search. Value of "`0`" means no limit. The default is "`200`".

Parameter	Description and Default Value
calstore.unqualifiedattendee.fmt1.type	Specifies how Calendar Server treats strings, such as `jdoe` or `jdoe:tv`, when performing a directory lookup for attendees of an event. Allowable values are: `uid`, `cn`, `gid`, `res`, `mailto`, `cap`. The default is "`uid`".
calstore.unqualifiedattendee.fmt2.type	Specifies how Calendar Server treats strings with an at sign (`@`), such as `jdoe@sesta.com`, when performing a directory lookup for attendees of an event. Allowable values are: `uid`, `cn`, `gid`, `res`, `mailto`, `cap`. The default is "`mailto`".
calstore.unqualifiedattendee.fmt3.type	Specifies how Calendar Server treats strings with a space, such as `john doe`, when performing a directory lookup for attendees of an event. Allowable values are: `uid`, `cn`, `gid`, `res`, `cap`. The default is "`cn`".
service.wcap.validateowners	If "`yes`", the server must validate that each owner of a calendar exists in the LDAP directory. The default is "`no`".
service.wcap.freebusy.redirecturl	If the requested calendar can't be found in the local calendar database, alternately, a URL found in this parameter can be used to redirect the search to another database. This is specifically used for scripts created when migrating between two databases and both are still being used. Then the `get_freebusy.wcap` command can be used to specify whether to look in the other database. See the `get_freebusy` command description in the *Sun Java System Calendar Server 6.3 WCAP Developer's Guide*.
store.partition.primary.path	Location of primary disk partition where calendar information is stored. The default is "`/var/opt/SUNWics5/csdb`".

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure Calendar Logging

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the parameters shown in the following table:**

Parameter	Description and Default Value
logfile.admin.logname	This log file contains history of the administrative tool commands issued. The default is "`admin.log`".
logfile.buffersize	Size in bytes for log buffers. The default is "`0`". Specify the size of each entry in the log files. If your buffers fill up too fast, consider making them larger.
logfile.dwp.logname	Name of the log file for logging Database Wire Protocol related administrative tools. The default is "`dwp.log`". Specify one per front-end server.
logfile.expirytime	Number of seconds before the log files expire. The default is "`604800`". After this time, a cleanup routine will purge the log. If you want to archive the log, you must write your own routine.
logfile.flushinterval	Number of seconds between the flushing of buffers to log files. the default is "`60`". If your system experiences a high volume of log information and your buffers fill up before 60 seconds, you will lose information. In that case consider decreasing this time interval. Note that decreasing the time interval increases system overhead.
logfile.http.logname	Name of the current log file for the `cshttpd` service. The default is "`http.log`".
logfile.http.access.logname	Name of the current HTTP access log file.
logfile.logdir	Directory location of the log files. The default is "`/var/opt/SUNWics5/logs`".
logfile.loglevel	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`. The default is `“NOTICE”`. 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`. Each succeeding log level also gives you all the more severe log levels before it. For example, if set to `WARNING`, only `CRITICAL`, `ERROR`, and `WARNING` level log entries are logged. If set to `DEBUG`, all levels are logged.
logfile.maxlogfiles	Maximum number of log files in the log directory. The default is "`10`". Before the system tries to create the 11th log, it runs the clean up routine to purge old log files.
logfile.maxlogfilesize	Maximum disk space in bytes for all log files. The default is "`2097152`". When creating the next log file will violate this limit, the system tries to free disk space by deleting the oldest logs.

Parameter	Description and Default Value
logfile.minfreediskspace	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. The default is `"5242880"`.
logfile.notify.logname	Name of the log file for the `csnotifyd` service. The default is `"notify.log"`.
logfile.rollovertime	Number of seconds before the log files are rotated. That is, the time interval between creation opening of new log files. The default is `"86400"`.
logfile.store.logname	Name of the log file for the calendar store. The default is `"store.log"`.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

See Also To configure transaction logging for the calendar database, see Chapter 9, "Configuring Automatic Backups (csstored)."

You do not have to configure the delete log (for deleted events and tasks). See Chapter 18, "Administering the Delete Log Database."

▼ To Configure WCAP Commands

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the following `ics.conf` parameters as shown in following table:**

Parameter	Description and Default Value
service.wcap.format	Specifies the default output format for commands. Two formats are supported: ■ "`text/calendar`" (default) ■ "`text/xml`" If you are using the Connector for Microsoft Outlook, you must use `text/calendar`.
service.wcap.version	WCAP version.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Enable Email Notifications

Three types of email notifications can be enabled:

- Send email notifications to attendees who are invited to an event.
- Send email notifications to attendees when an event is cancelled.
- Send email notifications to organizers when attendees reply.

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the following `ics.conf` parameters as shown in following table:**

Parameter	Description and Default Value
ine.invitation.enable	"`yes`"- (Default) To send notifications of invitations to attendees. "`no`"- Do not send email notifications of invitations to attendees.
ine.cancellation.enable	"`yes`"- (Default) To send notifications of event cancellations to attendees. "`no`"- Do not send notifications of cancellation to attendees.
ine.reply.enable	"`yes`"- (Default) To send organizers notifications of attendees' replies to invitations. "`no`"- Do not send organizers notifications of replies.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base/SUNWics5/cal/sbin/start-cal

See Also For more information about configuring notifications, see "E.4.1 Calendar Server Email Notifications Configuration Parameters and Format Files" on page 502.

4.5 Configuring Logins and Authentication

This section contains instructions for configuring logins and authentication.

This section contains the following topics:

- "To Configure Proxy Administrator Logins" on page 132
- "To Configure Authentication" on page 133
- "To Configure the Authentication Cache" on page 134
- "To Enable Checking the Client IP Address at Login" on page 135

▼ To Configure Proxy Administrator Logins

Proxy logins must be configured for Communications Express. For instructions on how to configure proxy logins for Communications Express, see"4.1 Configuring for Communications Express" on page 114.

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

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit the parameter that follows:**

service.http.allowadminproxy Specifies whether administrators are allowed to perform proxy logins to administer user calendars. If "`yes`", proxy logins are allowed. If "`no`" proxy logins are not allowed. The default value is "`yes`".

5 **Restart Calendar Server for the new value to take effect.**

6 **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&fmt-out=text/html
```

The following list contains an explanation of each variable in the previous example:

- *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.
- *fmt-out* is the specification for output format of the content. For example, text or HTML.

If the command is successful, Calendar Server displays the calendar for *calendar-user*. If problems occur, Calendar Server displays "Unauthorized".

Causes for error 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.

▼ To Configure Authentication

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the parameters shown in the following table:**

Parameter	Description/Default
local.authldapbasedn	Base DN for LDAP authentication. If not specified, *local.ugldapbasedn* is used.
local.authldaphost	Host for LDAP authentication. If not specified, uses the value of *local.ugldaphost*. The default is "`localhost`".
local.authldapbindcred	Bind credentials (password) for user specified in *local.authldapbinddn*.

Parameter	Description/Default
local.authldapbinddn	DN used to bind to LDAP authentication host to search for user's dn. If not specified or blank (" "), its assumed to be an anonymous bind.
local.authldapport	Port for LDAP authentication. If not specified, uses the value of *local.ugldapport*. The default is "`389`".
local.authldappoolsize	Minimum number of LDAP client connections that are maintained for LDAP authentication. If not specified, uses the value of *local.ugldappoolsize*. The default is "`1`".
local.authldapmaxpool	Maximum number of LDAP client connections that are maintained for LDAP authentication. If not specified, uses the value of *local.ugldapmaxpool*. The default is "`1024`".
local.user.authfilter	Specifies the authentication filter used for user lookup. The default is "`(uid=%U)`" This value is stored in the *inetDomainSearchFilter* attribute in the domain entry. It is possible to filter on a different attribute. For example, you could set this parameter to "`(mail=%U)`" The `uid` of the authenticated user is passed on to all other functions as the identity for that user, regardless of the attribute used for authentication.
service.plaintextloginpause	Number of seconds to delay after successfully authenticating a user with plain text passwords. The default is "`0`".

▼ To Configure the Authentication Cache

1. **Log in as an administrator with permission to change the configuration.**

2. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3. **Save your old `ics.conf` file by copying and renaming it.**

4. **Edit one or more of the parameters as shown in The following table:**

service.authcachesize	Maximum number of authenticated user ID's (`uids`) and passwords that Calendar Server will maintain in the cache. The default is `“10000”.`
service.authcachettl	Number of seconds since the last access before a `uid` and password are removed from the cache. The default is `“900”`.

5 **Save the file as** `ics.conf`**.**

6 **Restart Calendar Server.**

cal-svr-base/`SUNWics5/cal/sbin/start-cal`

▼ To Enable Checking the Client IP Address at Login

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Edit the following parameter as shown in the following table:**

service.dnsresolveclient	If "`yes`", when HTTP access is allowed, checks the client IP address against DNS. The default is "`no`".

5 **Save the file as** `ics.conf`**.**

6 **Restart Calendar Server.**

cal-svr-base/`SUNWics5/cal/sbin/start-cal`

4.6 Configuring Calendar Services (Daemons)

This section contains instructions on how to configure calendar services (daemons).

This section contains the following topics:

- "To Configure Start and Stop Services" on page 136
- "To Configure the Watcher Process for Calendar Server Version 6.3" on page 136
- "To Configure Administrative Services (csadmind)" on page 138
- "To Configure HTTP Services (cshttpd) for Calendar Server Version 6.3" on page 139
- "To Configure Alarm Notification for Calendar Server Version 6.3" on page 141

Tip – See also, Chapter 9, "Configuring Automatic Backups (csstored)."

▼ To Configure Start and Stop Services

The `start-cal` and `stop-cal` commands are wrapper scripts that allow ease of starting and stopping Calendar Server. The utility is defined in Appendix D, "Calendar Server Command-Line Utilities Reference."

1. **Log in as an administrator with permission to change the configuration.**

2. **Stop Calendar Server services by issuing the `stop-cal` command.**

3. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4. **Save your old `ics.conf` file by copying and renaming it.**

5. **Edit one or more of the parameters as shown in the following table:**

Parameter	Description and Default Value
local.serveruid	Runtime user identifier (`uid`). The default is "`icsuser`". This is the user identifier to use when super-user privileges are not needed.
local.servergid	Runtime group identifier (`gid`). The default is "`icsgroup`". This is the group identifier to use when super-user privileges are not needed.
local.autorestart	If this parameter is set to "`yes`", if a service that is connected to the `watcher` dies without properly disconnecting, it is automatically restarted.
local.autorestart.timeout	Defines the auto-restart timeout interval. To avoid infinite restart attempts on auto-start, if a service dies twice in a specific interval, it will not be restarted. The default setting is 10 minutes.

6. **Save the file as `ics.conf`.**

7. **Restart Calendar Server.**

   cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure the Watcher Process for Calendar Server Version 6.3

The watcher process, `watcher`, monitors failed socket connections. It is used with both Calendar Server and Messaging Server. To set the Calendar Server parameters to configure Watcher, perform the following steps:

1. **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4 **Save your old `ics.conf` file by copying and renaming it.**

5 **Edit one or more of the parameters as shown in the following table:**

Parameter	Description and Default Value
local.watcher.enable	If this parameter is set to `"yes"`, the start program attempts to start the `watcher` before any other services. And daemons will connect to it through a socket connection. The default is `"no"`, but the configuration program changes it to `"yes"`.
local.watcher.port	This is the port on which the `watcher` listens. Messaging Server uses port `49994`. A different port should be used for Calendar Server, for example `49995`.
local.watcher.config.file	The configuration file for `watcher`. If the path is relative, it is relative to the `config` directory. The default is `watcher.cnf`.
service.autorestart	If set to `"yes"`, the watcher automatically restarts any registered service that dies without properly disconnecting. If the service dies twice in 10 minutes, watcher will not restart it.

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

See Also For more information about the Watcher process, see *Sun Java System Messaging Server 6.3 Administration Guide*. Both Chapter 4 and Chapter 23 have information.

Note – If Watcher is enabled, each service the Watcher is monitoring must be registered with the Watcher process. This is done automatically and internally by Calendar Server daemons. Alternatively, the daemons create a `pid` files in the *cal-svr-base*`/data/proc` directory that contain the process ID of each service and its status, either `"init"`, or `"ready"`.

▼ To Configure Administrative Services (csadmind)

1. **Log in as an administrator with permission to change the configuration.**

2. **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3. **Save your old** `ics.conf` **file by copying and renaming it.**

4. **Edit one or more of the parameters as shown in the following table:**

Parameter	Description and Default Value
local.store.checkpoint.enable	If "`yes`", start the `csadmind` database checkpoint thread. If "`no`", no checkpoint log files created. The default is "`yes`".
service.admin.dbcachesize	Maximum cache size (in bytes) for Berkeley Database for administration sessions. The default is "`8388608`".
local.store.deadlock.enable	If "`yes`", start the `csadmind` database deadlock detection thread. The default is "`yes`".
service.admin.diskusage	If "`yes`", start the `csadmind` low disk space monitor thread. The default is "`no`". Disk usage is not monitored by default.
service.admin.enable	If "`yes`", start the `csadmind` service when starting all services and stop `csadmind` when stopping all services. The default is "`yes`".
service.admin.maxthreads	Maximum number of running threads per administration session. The default is "`10`".
service.admin.resourcetimeout	Number of seconds before timing out an administration connection. The default is "`900`".
service.admin.serverresponse	If "yes", start the csadmind service response thread. The default is "`no`".
service.admin.sessiondir.path	Temporary directory for administration session requests. No default.
service.admin.sessiontimeout	Number of seconds before timing out an HTTP session in `csadmind`. The default is "`1800`".
service.admin.sleeptime	Number of seconds to wait between checking for started, stopped, or ready calendar service. The default is "`2`".
service.admin.starttime	Number of seconds to wait for any calendar service to start. The default is "`300`".
service.admin.stoptime	Number of seconds to wait for any calendar service to stop. The default is "`300`".

Parameter	Description and Default Value
service.admin.stoptime.next	Number of seconds to wait between sending stop commands to any calendar service. The default is `“60”`.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure HTTP Services (cshttpd) for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the parameters as shown in the following table:**

Parameter	Description and Default Value
service.http.admins	Space separated list of user ID's with administration rights to this Calendar Server. The default is "`calmaster`".
service.http.allowadminproxy	If "`yes`", allow login via proxy, which is the default.
service.http.allowanonymouslogin	If "yes", allow anonymous (no authentication) access. This is a special type of login that is allowed only specified, restricted access (usually read only access to public calendars). The default is "`yes`".
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.commandlog	This parameter is for debugging only. If set to "`yes`", the system logs all incoming commands into the `http.commands` log file. Do not use this during production runtime. It will fill up the log file very quickly and could cause performance degradation.

Parameter	Description and Default Value
service.http.commandlog.all	This parameter is for debugging only. If set to "yes", the system logs all HTTP requests into the http.access log file. Do not use this during production runtime. It will fill up the log file very quickly and could cause performance degradation.
service.http.cookies	Tells the server to whether or to support cookies (yes/no). It must be set to "yes" to enable single sign-on. The default is "yes".
service.http.dbcachesize	Maximum cache size of Berkeley database for HTTP sessions. The default is "8388308".
service.http.domainallowed	If specified and not blank (" "), 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). The default is blank ("").
service.http.domainnotallowed	If specified and not blank (" "), 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). The default is blank (" ").
service.http.attachdir.path	Directory location relative to local.queuedir (or an absolute path if specified) where imported files are temporarily stored. The default is the current directory (".").
service.http.ipsecurity	If "yes", all requests that reference an existing session are verified as originating from the same IP address. The default is “yes”.
service.http.enable	If "yes", start the cshttpd service when starting all services and stop cshttpd when stopping all services. The default is “yes”. **Caution –** Disabling the HTTP service with this parameter will also disable HTTPS.
service.http.idletimeout	Number of seconds before timing out an HTTP connection. The default is “120”.
service.http.listenaddr	Specifies the TCP address that HTTP services will listen on for client requests. The default is "INADDR_ANY", which indicates any address.
service.http.logaccess	If "yes", HTTP connections to server are fully logged. The default is “no”.
service.http.maxsessions	Maximum number of HTTP sessions in cshttpd service. The default is “5000”.
service.http.maxthreads	Maximum number of threads to service HTTP requests in cshttpd service. The default is “20”.

Parameter	Description and Default Value
service.http.numprocesses	Maximum number of concurrently running HTTP service (`cshttpd`) processes that should run on a server. The default is “`1`”. For a server that has multiple CPU's, see “21.8 Using Load Balancing Across Multiple CPU's” on page 355.
service.http.port	Port for HTTP requests from Calendar Server users. The default is “`80`”.
service.http.proxydomainallowed	If specified and not "", filter for allowing proxy login based on TCP domains. Same syntax as *service.http.domainallowed*. The default is "".
service.http.resourcetimeout	Number of seconds before timing out an HTTP session. The default is “`900`”.
service.http.sessiondir.path	Directory for the HTTP session database. The default is “`http`”.
service.http.sessiontimeout	Number of seconds before timing out an HTTP session in `cshttpd` service. The default is “`1800`”.
service.http.sourceurl	Directory relative to executable where all URL references to files are stored. The default is "" (null).
service.http.tmpdir	Temporary directory for HTTP sessions. The default is “`/var/opt/SUNWics5/tmp`”.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure Alarm Notification for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the following `ics.conf` parameters as shown in the following table:**

Parameter	Description and Default Value
alarm.diskstat.msgalarmdescription	Description sent with insufficient disk space messages. The default description is: “percentage calendar partition diskspace available”.
alarm.diskstat.msgalarmstatinterval	Number of seconds between monitoring disk space. The default is “3600”.
alarm.diskstat.msgalarmthreshold	Percentage of available disk space that triggers sending a warning message. The default is “10”.
alarm.diskstat.msgalarmthresholddirection	Whether *alarm.diskstat.msgalarmthreshold* is above or below percentage. -1 is below and 1 is above. The default is “-1”.
alarm.diskstat.msgalarmwarninginterval	Number of hours between sending warning messages about insufficient disk space. The default is “24”.
alarm.msgalarmnoticehost	The host name of the SMTP server used to send server alarms. The default is “localhost”.
alarm.msgalarmnoticeport	The SMTP port used to send server alarms. The default is “25”.
alarm.msgalarmnoticercpt	The email address to whom server alarms sent. “Postmaster@localhost”
alarm.msgalarmnoticesender	The email address used as the sender when the server sends alarms. The default is “Postmaster@localhost”
alarm.msgalarmnoticetemplate	The default format used to send email alarms: "From: %s\nTo: %s\nSubject: ALARM: %s of \"%s\" is n\n%s\n"
alarm.responsestat.msgalarmdescription	Description sent with no service response messages. The default is “calendar service not responding”.
alarm.responsestat.msgalarmstatinterval	Number of seconds between monitoring services. The default is “3600”.
alarm.responsestat.msgalarmthreshold	The default is “100” (only trigger sending a warning message if no service response.)
alarm.responsestat. *msgalarmthresholddirection*	Specifies whether *alarm.responsestat.msgalarmthreshold* is above or below percentage. -1 is below and 1 is above. The default is “-1”
alarm.responsestat. *msgalarmwarninginterval*	Number of hours between sending warning messages about no service response sent out. The default is “24”.
local.rfc822header.allow8bit	Allow (“y”) or not allow (“n”) 8 bit headers in email messages sent by this server.

Parameter	Description and Default Value
service.admin.alarm	Enable (`"yes"`) or disable (`"no"`) alarm notifications for administration tools. The default is “`yes`”.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

4.7 Configuring Periodic Deadlock Checking for the Berkeley in Calendar Server Version 6.3

You can configure the Calendar Server to periodically check for deadlocks in the Berkeley databases.

It is possible for the Berkeley databases to get into a deadlocked state, thus preventing access to them. To detect this state as early as possible, enable periodic checking for deadlocks.

▼ To Enable Periodic Checking of Berkeley Databases for Deadlocks

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit the parameter shown in the following table:**

local.caldb.deadlock.autodetect	Periodically checks if the Berkeley database is in a deadlock state and, if so, instructs the database to reset. The default value is “`no`” (not enabled).

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

Troubleshooting For information about how to reset Berkeley databases once deadlocked, see "22.5.2 Detecting Database Corruption" on page 371"22.5.1.2 List of Available Tools" on page 370 in the Troubleshooting chapter.

4.8 Configuring LDAP for Calendar Server Version 6.3

This section contains instructions for configuring Calendar Server for LDAP.

This section contains the following topics:

▼ To Configure Anonymous Access to LDAP for Calendar Server Version 6.3

In general, anonymous access is allowed by default. If you want to restrict anonymous access, change the appropriate parameters.

1. **Log in as an administrator with permission to change the configuration.**

2. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3. **Save your old `ics.conf` file by copying and renaming it.**

4. **Edit one or more of the parameters in the following:**

Parameter	Description/Default
calstore.anonymous.calid	Specifies the anonymous login calendar identifier (`calid`). The default is "`anonymous`".
service.http.allowanonymouslogin	Specifies whether or not anonymous access is allowed without a login. The default is "`yes`". (Allows recipient of emailed calendar URL to access a free-busy version of the calendar without login in.)
service.wcap.anonymous. *allowpubliccalendarwrite*	Specifies whether or not to allow anonymous users to write to a publicly writable calendar. The default is "`yes`".
service.wcap.userprefs.ldapproxyauth	Enables anonymous search of the LDAP used for user preferences. The default is "`no`", which allows anonymous access. Specifying "`yes`" means using proxy authentication to do the search.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure LDAP Attendee Lookup for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the parameters in the following table:**

Parameter	Description/Default
local.lookupldap.search. *minwildcardsize*	Specifies the minimum string size for wildcard searches in an attendee lookup search. Zero (`0`) means always do a wildcard search.
sasl.default.ldap.searchfilter	Specifies the authentication filter for user lookup. The default is: `"(uid=%s)"`
local.lookupldapbasedn	Specifies the DN for LDAP attendee lookup. If not specified, uses *local.ugldapbsedn*. No default value.

Parameter	Description/Default
local.lookupldapbinddn	Specifies the DN to bind to the host used for LDAP attendee lookup. If not specified (default is ""), anonymous bind assumed.
local.lookupldapbindcred	Credentials (password) for user identified in *local.lookupldapbinddn*. No default value.
local.lookupldaphost	The host name for LDAP attendee lookup. If not specified, uses *local.ugldaphost*.
local.lookupldapmaxpool	Specifies the number of LDAP client connections maintained for LDAP attendee lookup. If not specified, uses *local.ugldapmaxpool*. The default is "1024".
local.lookupldappoolsize	Specifies the minimum number of LDAP client connections maintained for LDAP attendee lookup. If not specified, uses *local.ugldappoolsize*. The default is "1".
local.lookupldapport	Specifies the port to use for LDAP attendee lookup. If not specified, uses *local.ugldapport*.
local.lookupldapsearchattr.calid	Specifies the calid attribute for attendee lookup. The default is *icsCalendar*.
local.lookupldapsearchattr.mail	Specifies the mail attribute for attendee lookup. The default is *mail*.
local.lookupldapsearchattr. *mailalternateaddress*	Specifies the alternate mail address attribute for attendee lookup. The default is *mailalternateaddress*.
local.lookupldapsearchattr. *mailequivalentaddres*	Specifies the equivalent address mail attribute for attendee lookup. The default is *mailequivalentaddress*.
local.lookupldapsearchattr. *calendar*	Specifies the calendar attribute for attendee lookup. The default is *icsCalendar*.
local.lookupldapsearchattr.cn	Specifies the common name attribute for attendee lookup. The default is *cn*.
local.lookupldapsearchattr. *objectclass*	Specifies the object class attribute for attendee lookup. The default is *objectclass*.
local.lookupldapsearchattr. *objectclass.caluser*	Specifies the object class for calendar users. The default is *icsCalendarUser*.
local.lookupldapsearchattr. *objectclass.calresource*	Specifies the object class for calendar resources. The default is *icsCalendarResource*.
local.lookupldapsearchattr. *objectclass.group*	Specifies the object class for groups. The default is *icsCalendarGroup*.

Parameter	Description/Default
local.lookupldapsearchattr. *objectclass.person*	Specifies the object class for persons. The default is *person*.
local.lookupldapsearchattr. *memberurl*	Specifies the member URL attribute for attendee lookup. The default is *memberurl*.
local.lookupldapsearchattr. *uniquemember*	Specifies the unique member attribute for attendee lookup. The default is *uniquemember*.
local.lookupldapsearchattr. *givenname*	Specifies the given name attribute for attendee lookup. The default is *givenname*.
local.lookupldapsearchattr.sn	Specifies the screen name attribute for attendee lookup. The default is *sn*.
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 setting is "`sesta.com`".

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base/`SUNWics5/cal/sbin/start-cal`

▼ To Configure Search Filters for LDAP Attendee Lookup for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one or more of the parameters in the following table:**

Tip – In all the parameter descriptions that follow, `%s` allows only a single attendee.

Parameter	Description/Default
local.lookupldap.calid.direct	The search filter for calid-search-type using direct lookup. The default is: `"(icsCalendar=%s)` %s – The attendee string.

local.lookupldap.cn.direct	The search filter for cn-search-type in direct lookup. The default is:
	`"(&(cn=%s)`
	`(|(objectclass=groupofuniquenames)`
	`(objectclass=icsCalendarResource)`
	`(objectclass=person)))"`
	`%s` – The attendee string.

local.lookupldap.cn.search	The search filter for cn-search-type in search dialog lookup. The default is for a single attendee string (`%s`):
	`"(&(cn=%s)`
	`  (|(objectclass=groupofuniquenames)`
	`  (objectclass=icsCalendarResource)`
	`  (objectclass=person)))"`
	For a wild card search (multiple search strings):
	`"(&(cn=%w)`
	`  (|(objectclass=groupofuniquenames)`
	`  (objectclass=icsCalendarResource)`
	`  (objectclass=person)))"`
	`%w` – Causes expansion to a list of attendee strings. For example: `%w="Mary Ann Smith"` expands to:
	`(& (cn=*Mary*) (cn="*Ann")`
	` (cn=*Smith*)`

local.lookupldap.gid	The search filter for `gid` search type. The default is:
	`"(&(cn=%s)`
	`   (objectclass=groupofuniquenames))"`
	`%s` — A single attendee string.

<table>
<tr><th>Parameter</th><th>Description/Default</th></tr>
<tr><td><i>local.lookupldap.mailto.indomain</i></td><td>The search filter for mailto-search-type in the domain specified by <i>local.smtp.defaultdomain</i>. The default is:
"(|(mail=%s)(mail=%h)(mail=*<%s\>*)
(uid=%o))"
%s – The attendee string.
%o – The attendeeuid.
%h – The query string without the domain part.
For example: if %s=jdoe@sesta.com, %o=jdoe@sesta.com and %h=jdoe, then the value is:
(|(mail=jdoe@varrius.com)
(mail=jdoe)
(mail=*<jdoe@varrius.com\>*)
(uid=jdoe@varrius.com))</td></tr>
<tr><td><i>local.lookupldap.mailto.outdomain</i></td><td>The search filter for mailto-search-type where the domain is not the one specified by local.smtp.defaultdomain. The default is: "(|(mail=%s)(uid=%s))"
%s – The attendee string.</td></tr>
<tr><td><i>local.lookupldap.res</i></td><td>The search filter for res search type (resource search). The default is:
"(&(cn=%s)
(objectclass=icsCalendarResource))"
$s – The attendee string.</td></tr>
<tr><td><i>local.lookupldap.res.ugldap</i></td><td>The search filter for res search type (resource search) only on the User/Group LDAP server. This is only set when <i>local.lookupldap.resource.use.ugldap</i> is set to “yes”. The default is:
"(&(cn=%s)
(objectclass=icsCalendarResource))"
%s – The attendee string.</td></tr>
<tr><td><i>local.lookupldap.uid.direct</i></td><td>The search filter for uid search type using direct lookup. The default is:
"(|(uid=%s)(&(cn=%s)
(|(objectclass=groupofuniquenames)
(objectclass=icsCalendarResource)
(objectclass=person))))"
%s – The attendee string.</td></tr>
</table>

Parameter	Description/Default

local.lookupldap.uid.search	The search filter for `uid` search type lookup using a search dialog. The default is:
	`"(|(uid=%o)(&(cn=%w)`
	`(|(objectclass=groupofuniquenames)`
	`(objectclass=icsCalendarResource)`
	`(objectclass=person))))"`
	`%s` – The attendee string.
	`%w` – The attendee string with wildcards.
	`%o` – The attendee string without wildcards.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

To Configure LDAP Resource Lookup for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit the parameter shown in the following table:**

local.lookupldap.resource.use.ugldap	Whether to use the User/Group LDAP server for resource lookup, or the Lookup server. “`yes`” – Use the User/Group LDAP server. “`no`” – Use the Lookup server. The default is “`no`”.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure LDAP Mail-to-Calid Lookup for Calendar Server Version 6.3

1. **Log in as an administrator with permission to change the configuration.**

2. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3. **Save your old `ics.conf` file by copying and renaming it.**

4. **Edit one or more of the parameters in the following table:**

Parameter	Description/Default
local.lookupldap.mailtocalid.search	Specifies the mail attributes to use for mail-to-calid lookup. The default is `"(|(mail=%s)(mailalternateaddress=%s))"` You can substitute the attribute *mailequivalentaddress* in place of `mailalternateaddress.`
local.ugldapbasedn	Specifies the base DN for mail-to-calid lookup.
local.authldapbinddn	Specifies the DN to bind to the host used for mail-to-calid lookup. If not specified (default is `""`), anonymous bind assumed.
local.authldapbindcred	Specifies the password for the DN specified in *local.authldapbinddn*. No default.
local.ugldaphost	Specifies the LDAP host used for mail -to-calid lookup.
local.ugldapmaxpool	Specifies the maximum number of client connections maintained for mail-to-calid lookup. The default is `“1024”`.
local.ugldappoolsize	Specifies the minimum number of client connections to maintain for mail-to-calid lookup. The default is `“1”`.
local.ugldapport	Specifies the port for the LDAP mail-to-calid lookup. No default.

5. **Save the file as `ics.conf`.**

6. **Restart Calendar Server.**

   cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure the User Preferences LDAP Directory for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Edit one or more of the parameters in the following table:**

Parameter	Description/Default
local.enduseradmincred	Bind credentials (password) for LDAP user preferences authentication. No default.
local.enduseradmindn	DN used to bind to LDAP user preferences host. Must be specified. If blank (" ") or not specified, assumes an anonymous bind.
local.ugldappoolsize	Minimum number of LDAP client connections that are maintained for LDAP user preferences. The default is "`1`".
local.ugldapmaxpool	Maximum number of LDAP client connections that are maintained for LDAP user preferences. The default is "`1024`".
service.wcap.userprefs.ldapproxyauth	Enables anonymous search of the LDAP used for user preferences. The default is "`no`", which allows anonymous access. Specifying "`yes`" means using proxy authentication to do the search.

5 **Save the file as** `ics.conf`.

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure User Preferences for Calendar Server Version 6.3

You can restrict the preferences users are allowed to set by removing them from the default list.

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3. **Save your old `ics.conf` file by copying and renaming it.**

4. **Edit the list of user preferences in the parameter shown in the following table:**

Parameter	Default List of User Preferences	Description

local.
ugldapicsextendeduserprefs
	`"ceColorSet,`
	`ceFontFace,`
	`ceFontSizeDelta,`
	`ceDateOrder,`
	`ceDateSeparator,`
	`ceClock,`
	`ceDayHead,`
	`ceDayTail,`
	`ceInterval,`
	`ceToolText,`
	`ceToolImage,`
	`ceDefaultAlarmStart,`
	`ceSingleCalendarTZID,`
	`ceAllCalendarTZIDs,`
	`ceDefaultAlarmEmail,`
	`ceNotifyEmail,`
	`ceNotifyEnable,`
	`ceDefaultView,`
	`ceExcludeSatSun,`
	`ceGroupInviteAll"`
		User preference values are kept in LDAP. This parameter defines which user preferences are kept in LDAP in the *icsExtendedUserPrefs* attribute.

5. **Save the file as `ics.conf`.**

6. **Restart Calendar Server.**

   cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Enable and Configure the LDAP Data Cache for Calendar Server Version 6.3

Before You Begin For overview information about the LDAP Data Cache, see "1.7 LDAP Data Cache Option for Calendar Server Version 6.3" on page 48.

1. **Log in as an administrator with permission to change the configuration.**
2. **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**
3. **Save your old** `ics.conf` **file by copying and renaming it.**
4. **Enable the LDAP data cache by editing the parameter as shown in the following table:**

Parameter	Description and Default Value
local.ldap.cache.enable	Enable or disable the LDAP cache. If `"yes"`, the cache is enabled. If `"no"` the cache is disabled. 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	Not currently implemented. 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.

Parameter	Description and Default Value
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).

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

See Also For information about tuning the LDAP data cache, see "21.5 Improving Performance of the LDAP Data Cache" on page 352.

Caution – If Calendar Server or the server where Calendar Server is running is not properly shut down, manually delete all files in the `ldap_cache` directory to avoid any database corruption that might cause problems during a subsequent restart.

▼ To Enable and Configure the LDAP SDK Cache for Calendar Server Version 6.3

The LDAP SDK cache is disabled by default.

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Editing one or more of the parameters as shown in the following table:**

service.ldapmemcache	If "yes", enables LDAP SDK cache. The default is "no".
service.ldapmemcachettl	If *service.ldapmemcache* is "`yes`", this parameter is used to set 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. The default is "`30`".
service.ldapmemcachesize	If *service.ldapmemcache* is "`yes`", this parameter is used to set the maximum amount of memory in bytes that the cache will consume. If "`0`", the cache has no size limit. The default is "`131072`".

5 **Save the file as** `ics.conf`.

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Set the Date Range for Free Busy Searches for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Edit one or more of the following parameters as shown in the following table:**

service.wcap.freebusybegin	Specifies the offset from the current time in days for *get_freebusy* for beginning of the range. The default is “30”.
service.wcap.freebusyend	Specifies the offset from the current time in days for *get_freebusy* for end of the range. The default is “30”.

5 **Save the file as** `ics.conf`.

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Enable Wildcard LDAP Searches of Calendar Properties for Calendar Server Version 6.3

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Edit the parameter as shown in the following table:**

service.calendarsearch.ldap.primaryownersearchfilter

The default search filter used for *search_calprops* searches for exact matches to the search string. To allow wildcard searches such that matches are found when the search string is merely contained within the property value, uncomment this parameter. This enables the system to use the following search filter:

```
"(&(|(uid=*%s*)(cn=*%s*))
(objectclass=icsCalendarUser))"
```

Enabling this search filter can negatively impact performance.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Set the LDAP Root Suffix in Calendar Server Version 6.3

While it is possible to reset the root suffix for your LDAP organization tree (Schema version 2), or domain component tree (Schema version 1), this should be done with great care. It would be better to rerun the configuration program to do this.

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit one of the parameters as shown in the following table:**

service.dcroot	Root suffix of the DC tree in the directory. Required for multiple domain support using Schema version 1, and Schema version 2 compatibility mode (1.5). The default is "`o=internet`". See also "10.2 Setting up a Multiple Domain Environment for Calendar Server Version 6.3 for the First Time" on page 233.
service.schema2root	Root suffix of the DIT (Organization Tree) for Schema version 2. No default value.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server:**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

CHAPTER 5

Configuring Calendar Database Distribution Across Multiple Machines in Calendar Server Version 6.3

This chapter describes how to use the Calendar Lookup Database (CLD) plug-in to enable the calendar database to be distributed over multiple back-end servers. You must both enable and configure the CLD plug-in.

Caution – You must run the same version of Calendar Server on both the front-end and back-end servers.

This chapter contains the following topics:

- "5.1 CLD Plug-in Background Information for Calendar Server Version 6.3" on page 159
- "5.2 Configuring Calendar Servers for CLD and DWP" on page 165
- "5.3 Maintaining Security Between Front-End and Back-End Servers for Calendar Server Version 6.3" on page 171

Tip – For information on how to improve the performance of the CLD plug-in, see Chapter 21, "Tuning Calendar Server Performance."

5.1 CLD Plug-in Background Information for Calendar Server Version 6.3

This section covers valuable overview and background information that you might want to understand before actually enabling and configuring the CLD plug-in.

This section contains the following topics:

- "5.1.1 CLD Plug-in Overview for Calendar Server Version 6.3" on page 160
- "5.1.2 How the CLD Plug-in Works for Calendar Server Version 6.3" on page 160
- "5.1.3 Configurations Supported by the CLD Plug-in for Calendar Server Version 6.3" on page 161

5.1.1 CLD Plug-in Overview for Calendar Server Version 6.3

The Calendar Lookup Database (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. When the calendar database is distributed over several back-end servers, Calendar Server uses the CLD 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.

5.1.2 How the CLD Plug-in Works for Calendar Server Version 6.3

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

1. When an end user accesses a calendar through Communications Express, the CLD plug-in extracts the `userid` from the calendar's `calid` and then looks up the calendar owner in the LDAP directory database, or the CLD data cache (if enabled). For information and instructions on configuring a front-end machine, see "To Configure a Front-End Server for CLD" on page 165.
2. After finding the calendar owner, the plug-in uses the value in the *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).
4. Using DWP, Calendar Server sends the calendar data to the server where the user is logged in, so it can be rendered in one of the user interfaces.

Tip – If your site is using the CLD plug-in, all calendars created for the same user must reside on the same back-end server, as indicated by the LDAP user entry's *icsDWPHost* LDAP attribute. If you try to create a calendar on a different back-end server, Calendar Server returns an error.

5.1.3 Configurations Supported by the CLD Plug-in for Calendar Server Version 6.3

This section contains overview material about the CLD Plug-in.

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

- "5.1.3.1 Multiple Front-end Servers with Multiple Back-end Servers in Calendar Server Version 6.3" on page 161
- "5.1.3.2 Multiple Machines Functioning as Both Front-end and Back-end Servers in Calendar Server Version 6.3" on page 162

Tip –

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

- Be on the same hardware platform.
- Be running the same 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".

5.1.3.1 Multiple Front-end Servers with Multiple Back-end Servers in Calendar Server Version 6.3

Figure 5–1 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.

For configuration instructions, see "5.2 Configuring Calendar Servers for CLD and DWP" on page 165.

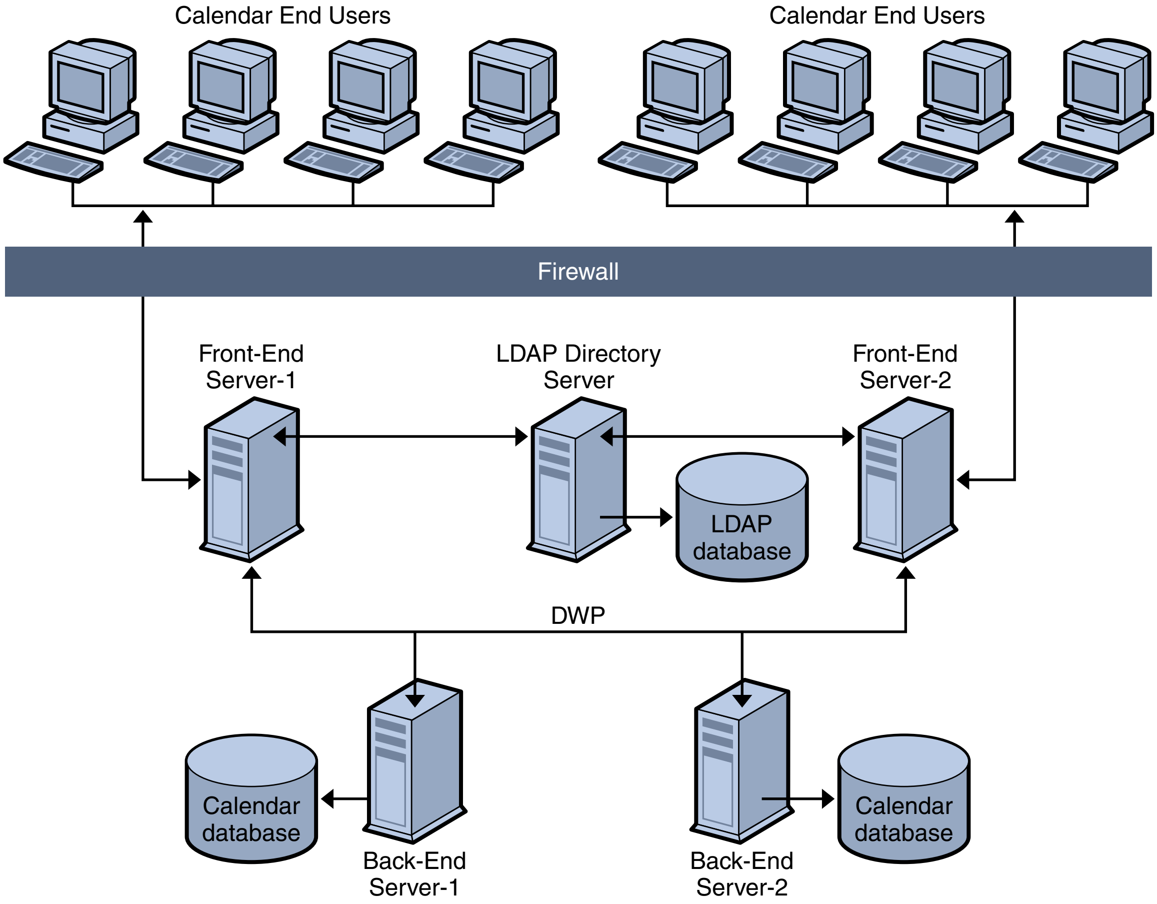


FIGURE 5–1 Multiple Front-End Servers with Multiple Back-End Servers

5.1.3.2 Multiple Machines Functioning as Both Front-end and Back-end Servers in Calendar Server Version 6.3

Figure 5–2 shows three machines functioning as both front-end and back-end servers. Each machine is connected to a calendar database. This configuration allows calendars to be geographically distributed. Calendar owners (end users) log into the machine where their calendars reside. For configuration instructions, see "To Configure a Server as Both a Front-end and a Back-end" on page 169.

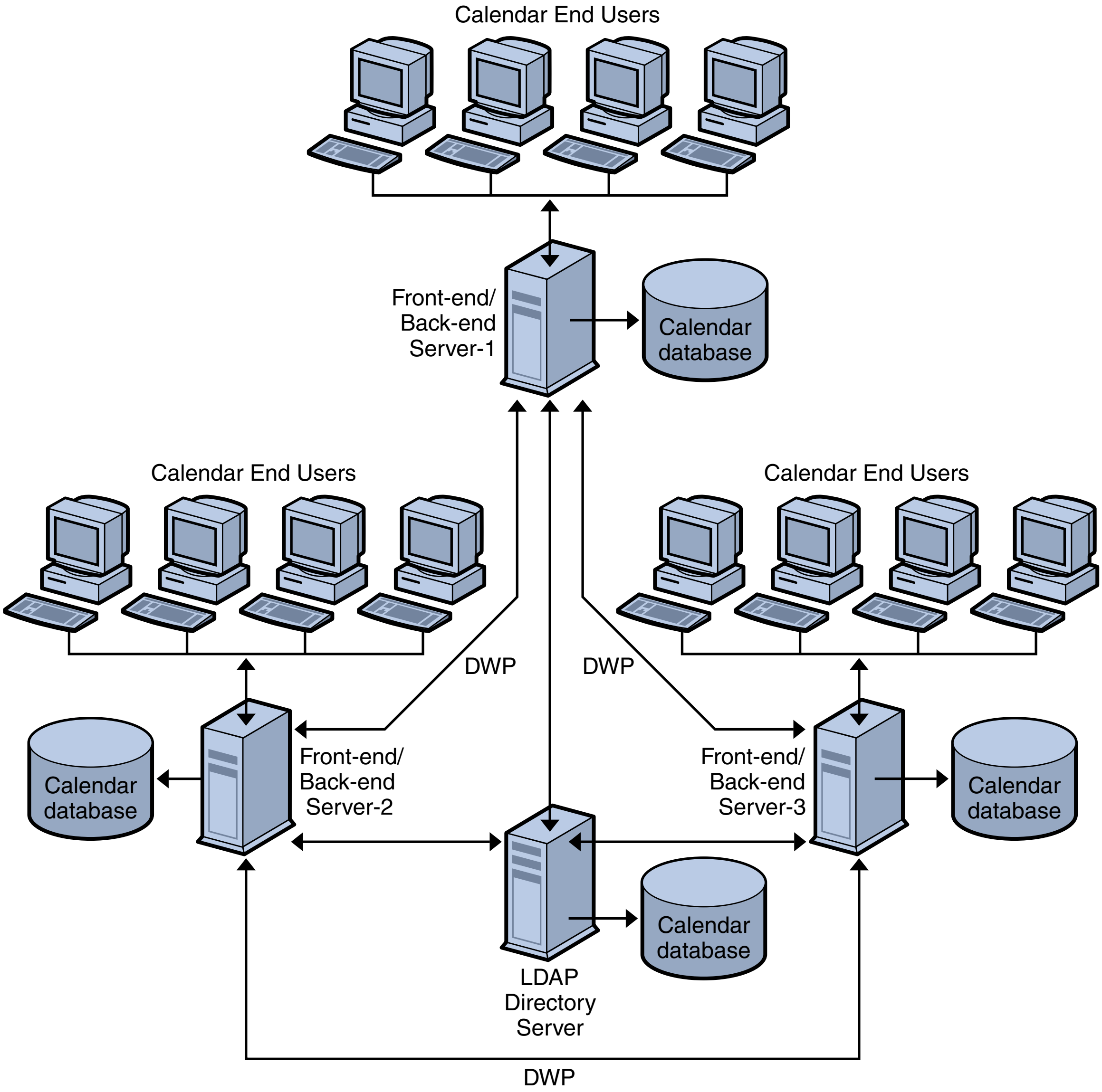


FIGURE 5–2 Multiple Servers as Functioning as Both Front-end and Back-end

5.1.4 Simple Sizing Exercise for Calendar Server 6.3 Storage Requirements

This section describes a simple sizing method using a few rough formulas based on a medium usage profile. They allow you to figure out how many front-end and back-end servers you need and how much storage.

This sections covers the following topics:

- "5.1.4.1 Definition of Medium Usage Profile for a Calendar Server 6.3 Deployment" on page 164
- "5.1.4.2 Number of Front-End CPU's" on page 164
- "5.1.4.3 Number of Back-End CPU's" on page 164
- "5.1.4.4 Amount of Storage Needed" on page 164

5.1.4.1 Definition of Medium Usage Profile for a Calendar Server 6.3 Deployment

For our rough estimates, we are assuming the following:

- All clients are Web clients.

  Therefore, the only inputs to be used are: total number of users, and percent concurrency.
- The average size calendar event size is 5K.
- Each person creates ten events or todos per week.
- 80% CPU utilization.
- 900 MHz CPU's.
- 1 GB RAM per CPU.
- Two years' worth of calendar data stored on system.
- Six hot backup copies held in storage.

5.1.4.2 Number of Front-End CPU's

The formula is:

Number of CPU's = Number of Concurrent Users divided by 4800

5.1.4.3 Number of Back-End CPU's

The formula is:

Number of CPU's = 4 CPU's per 500,000 configured users

5.1.4.4 Amount of Storage Needed

The formula is:

Amount of Storage Per User = 100 emails per week, multiplied by 52 weeks a year, multiplied by 5K per email, multiplied by the number of years worth of data to keep online, multiplied by the number of copies (5 backups + 1 working copy) kept online = 100*52*5K*2*(5+1) = 65 MB storage per user.

That is 2.6 MB per user per year per copy held online.

Note – The final number depends on how many hot backups or archival backups you keep online. For this example, 5 backup copies was the number used.

5.2 Configuring Calendar Servers for CLD and DWP

This sections contains instructions for configuring servers for CLD and DWP.

This section contains the following topics:

- "To Configure a Front-End Server for CLD" on page 165
- "To Configure a Back-end Server for CLD and DWP" on page 168
- "To Configure a Server as Both a Front-end and a Back-end" on page 169

▼ To Configure a Front-End Server for CLD

1 **On every front-end server, log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Edit the** `ics.conf` **parameters as shown in the following list:**

Parameters	Description
csapi.plugin.loadall	For every front-end server, set the value to "`y`" if you want all plug-ins starting with `cs_` to be loaded into the *cal-svr-base*`/SUNWics5/cal/bin/plugins` directory. Set to "`n`", to load only a specific plug-in, the name of which is found in *csapi.plugin.calendarlookup.name*.
csapi.plugin.calendarlookup	Set this parameter to "`yes`".

csapi.plugin.calendarlookup.name

Set this parameter to the name of the plug-in,*"calendarlookup"*. Or, to load all plug-ins, set the parameter to `"*"`.

caldb.cld.type

This parameter specifies whether calendars are to be distributed across multiple back-ends (set value to "`directory`"), or calendars are to be stored on the same server on which Calendar Server is installed (set value to,"`local`", which is the default value).

service.dwp.enable

Disable DWP service for the front-end, unless it is also serving as a back-end machine. For example: `service.dwp.enable="no"`

service.dwp.port

The default port is "`59979`". This port number must be the same for all front-end and back-end servers.

service.store.enable

This parameter is enabled by default (value = *"yes"*). It does not appear in the configuration file (`ics.conf`).

It must be added to the configuration file if you wish to disable it (value = *"no"*).

caldb.dwp.server.backend-server-n.ip

This is a multi-valued parameter. Create one `ics.conf`parameter for each back-end server in your Calendar Server deployment. The value of this parameter is the back-end server hostname. The server name must be fully qualified and be resolvable by your Domain Name Service (DNS) into a valid IP address. The server name must be identical and fully qualified in both the parameter name and the value.

For example:

```
caldb.dwp.server.calendar1.sesta.com="calendar1.sesta.com"

caldb.dwp.server.calendar2.sesta.com="calendar2.sesta.com"
```

caldb.dwp.server.default

Set the default DWP server name used by the system if the user or resource LDAP entry does not have an icsDWPHost attribute. The server name must be fully qualified and be resolvable by your DNS.

For example:

```
caldb.dwp.sever.default="calendar1.sesta.com"
```

local.authldaphost

The hostname where the Directory Server is installed. The default is "`localhost`".

local.ugldaphost

The hostname where the LDAP user preferences are stored. If you do not keep the user preferences in a separate LDAP host, then it should be set to the same value as *local.authldaphost*.

service.ens.enable

Disable ENS (`enpd`) for this front-end server, set this parameter to "`no`".

ENS must be enabled only on the back-end servers.

caldb.serveralarms

To disable server alarms for the front-end by setting this to "`0`".

Server alarms must be enabled ("`1`") only on the back-end servers.

caldb.serveralarms.dispatch

To disable the alarm dispatcher, set this parameter to "`no`".

The alarm dispatcher should be enabled ("`yes`") only on the back-end servers.

service.notify.enable

To disable the notify service, set this parameter to "`no`".

The notify service should be enabled ("`yes`") only on back-end servers.

caldb.berkeleydb.archive.enable

To disable the automatic archive backup service, set this parameter to "`no`". There is no need to have archiving configured on a front-end machine.

caldb.berkeleydb.hotbackup.enable

The automatic hot backup service should be disabled (value set to "`no`"). There is no need for hot backups on a front-end machine.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure a Back-end Server for CLD and DWP

1. **On every back-end server, log in as an administrator with permission to change the configuration.**

2. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3. **Save your old `ics.conf` file by copying and renaming it.**

4. **Edit the `ics.conf` parameters as shown in the following table:**

Parameters	Description
service.http.enable	Set this parameter to "`no`". There is no need for HTTP on a back-end server.
service.admin.enable	Enable the administration service (`csadmind`) by setting the parameter to "`yes`", which is the default.
caldb.cld.type	If this machine is a back-end only machine, set to "`local`". If this machine is both a front-end and a back-end, set to "`directory`".
csapi.plugin.calendarlookup	Set this parameter to "`no`". There is no need for a plug-in on a back-end server.
service.dwp.enable	Enable DWP by setting this parameter to "`yes`"
service.dwp.port	The default port is `“59979”`. This port number must be the same for all front-end and back-end servers.
caldb.dwp.server.backend-server-n.ip	This is a multi-valued parameter. Create one `ics.conf`parameter for each back-end server in your Calendar Server deployment. The value of this parameter is the back-end server hostname. The server name must be fully qualified and be resolvable by your Domain Name Service (DNS) into a valid IP address. The server name must be identical and fully qualified in both the parameter name and the value. For example:

```
caldb.dwp.server.calendar1.sesta.com="calendar1.sesta.com"

caldb.dwp.server.calendar2.sesta.com="calendar2.sesta.com"
```

caldb.dwp.server.default	Set the default DWP server name used by the system if the user or resource LDAP entry does not have an *icsDWPHost* attribute. The server name must be fully qualified and be resolvable by your DNS. For example: `caldb.dwp.sever.default="calendar1.sesta.com"`
local.authldaphost	The hostname where the Directory Server is installed. The default is "`localhost`".
local.ugldaphost	The hostname where the LDAP user preferences are stored. If you do not keep the user preferences in a separate LDAP host, then it should be set to the same value as *local.authldaphost*.
service.ens.enable	To enable ENS (`enpd`) for this back-end server, set this parameter to "`yes`".
caldb.serveralarms	Server alarms must be enabled ("`1`") on the back-end servers.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

To Configure a Server as Both a Front-end and a Back-end

1 **On every server, log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit the `ics.conf` parameters as shown in the following table:**

Parameters	Description
csapi.plugin.loadall	For every front-end server, set the value to "`y`" if you want all plug-ins starting with `cs_` to be loaded into the *cal-svr-base*`/SUNWics5/cal/bin/plugins` directory.

	Set to “`n`”, to load only the CLD plug-in, the name of which is found in *csapi.plugin.calendarlookup.name*.
csapi.plugin.calendarlookup	Set this parameter to "`yes`".
csapi.plugin.calendarlookup.name	To load all plug-ins, set the parameter to "`*`". If you want to load only the CLD plug-in, set this parameter to the name of the plug-in,"`calendarlookup`".
caldb.cld.type	This parameter specifies whether calendars are to be distributed across multiple back-ends (set value to “`directory`”), or calendars are to be stored on the same server on which Calendar Server is installed (set value to,“`local`”, which is the default value).
service.dwp.enable	Enable DWP by setting this parameter to "`yes`"
service.dwp.port	The default port is “`59979`”. This port number must be the same for all front-end and back-end servers.
caldb.dwp.server.backend-server-n.ip	This is a multi-valued parameter. Create one `ics.conf`parameter for each back-end server in your Calendar Server deployment. The value of this parameter is the back-end server hostname. The server name must be fully qualified and be resolvable by your Domain Name Service (DNS) into a valid IP address. The server name must be identical and fully qualified in both the parameter name and the value. For example:

```
caldb.dwp.server.calendar1.sesta.com="calendar1.sesta.com"

caldb.dwp.server.calendar2.sesta.com="calendar2.sesta.com"
```

caldb.dwp.server.default	Set the default DWP server name used by the system if the user or resource LDAP entry does not have an icsDWPHost attribute. The server name must be fully qualified and be resolvable by your DNS. For example: `aldb.dwp.sever.default="calendar1.sesta.com"`

local.authldaphost	The hostname where the Directory Server is installed. The default is “`localhost`”(on the same server as the front-end).
local.ugldaphost	The hostname where the LDAP user preferences are stored. If you do not keep the user preferences in a separate LDAP host, then it should be set to the same value as *local.authldaphost*.
service.ens.enable	Enable ENS by setting this parameter value to "`yes`".
caldb.serveralarms	Server alarms must be enabled ("`1`") on the back-end servers.
caldb.serveralarms.dispatch	The alarm dispatcher should be enabled ("`yes`") on the back-end servers.
service.notify.enable	The notify service should be enabled ("`yes`") on back-end servers.
caldb.berkeleydb.archive.enable	The automatic archive backup service should be enabled (value set to "`yes`"on the back-end systems.
caldb.berkeleydb.hotbackup.enable	The automatic hot backup service should be enabled (value set to "`yes`"on the back-end systems.

5 **Save the file as** `ics.conf`**.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

5.3 Maintaining Security Between Front-End and Back-End Servers for Calendar Server Version 6.3

You can configure password authentication between front-end and back-end servers. This section explains how secure communication between the two can be set up and how it works.

This section covers the following topics:

- “5.3.1 How Authentication is Accomplished in Calendar Server Version 6.3” on page 172
- “To Set Up Authentication for DWP Connections for a Front-end Server in Calendar Server Version 6.3” on page 172
- “To Set up Authentication for DWP Connections for a Back-end Server in Calendar Server Version 6.3” on page 173

5.3.1 How Authentication is Accomplished in Calendar Server Version 6.3

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 provides authentication for DWP connections between front-end and back-end servers using configuration parameters.

When the front-end server first connects to the back-end server, it sends the user ID and password specified in the `ics.conf` file. The back-end server checks the parameters in its `ics.conf` file, 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 servers, 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, no authentication is performed.

▼ To Set Up Authentication for DWP Connections for a Front-end Server in Calendar Server Version 6.3

Before You Begin

⚠ **Caution –** These parameters are not included in the installed version of 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 server.

1. **On every front-end server, log in as an administrator with permission to change the configuration.**

2. **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3. **Save your old** `ics.conf` **file by copying and renaming it.**

4. **Add the** `ics.conf` **parameters as shown in the following list:**

Parameter	Description

caldb.dwp.server.back-end-server.admin	On a front-end server, specifies the administrator's 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.

5 **Save the file as** `ics.conf`**.**

6 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Set up Authentication for DWP Connections for a Back-end Server in Calendar Server Version 6.3

Before You Begin

Caution – These parameters are not included in the installed version of the `ics.conf` file. To use authentication for DWP connections, you must add the required parameters to the `ics.conf` file on each back-end server.

1 **On every back-end server, log in as an administrator with permission to change the configuration.**

2 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

3 **Save your old** `ics.conf` **file by copying and renaming it.**

4 **Add the** `ics.conf` **parameters as shown in the following table:**

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.

5 **Save the file as** `ics.conf`**.**

6 **Restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

CHAPTER 6

Configuring Calendar Server 6.3 Software for High Availability (Failover Service)

This chapter explains how to install and configure high availability for Calendar Server 6.3 software using Sun Cluster 3.0 or 3.1.

Configuring Calendar Server for High availability (HA) provides for monitoring of and recovery from software and hardware failures. The Calendar Server HA feature is implemented as a failover service. This chapter describes two Calendar Server HA configurations using Sun Cluster software, one asymmetric and one symmetric.

This chapter includes the following topics to describe how to install and configure HA for Calendar Server:

You can find a set of worksheets to help you plan a Calendar Server HA configuration in the Appendix C, "Calendar Server Configuration Worksheet."

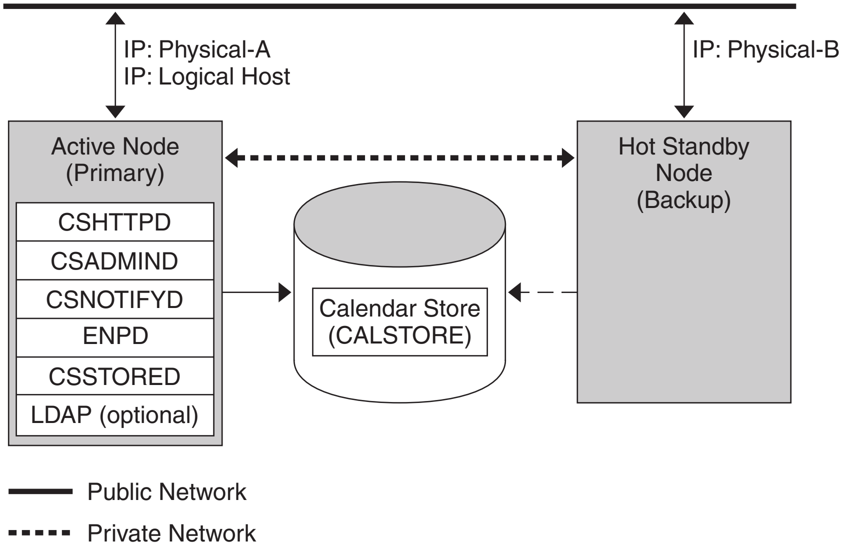
6.1 Overview of High Availability Choices for Calendar Server Version 6.3

High availability can be configured many ways. This section contains an overview of three high availability choices, and information to help you choose which is right for your needs.

This sections covers the following topics:

- “6.1.1 Understanding Asymmetric High Availability for Calendar Server Version 6.3” on page 176
- “6.1.2 Understanding Symmetric High Availability for Calendar Server Version 6.3” on page 177
- “6.1.3 Understanding N+1 (N Over 1): Multiple Asymmetric High Availability for Calendar Server Version 6.3” on page 179
- “6.1.4 Choosing a High Availability Model for Your Calendar Server Version 6.3 Deployment” on page 180
- “6.1.5 System Down Time Calculations for High Availability in Your Calendar Server 6.3 Deployment” on page 181

6.1.1 Understanding Asymmetric High Availability for Calendar Server Version 6.3



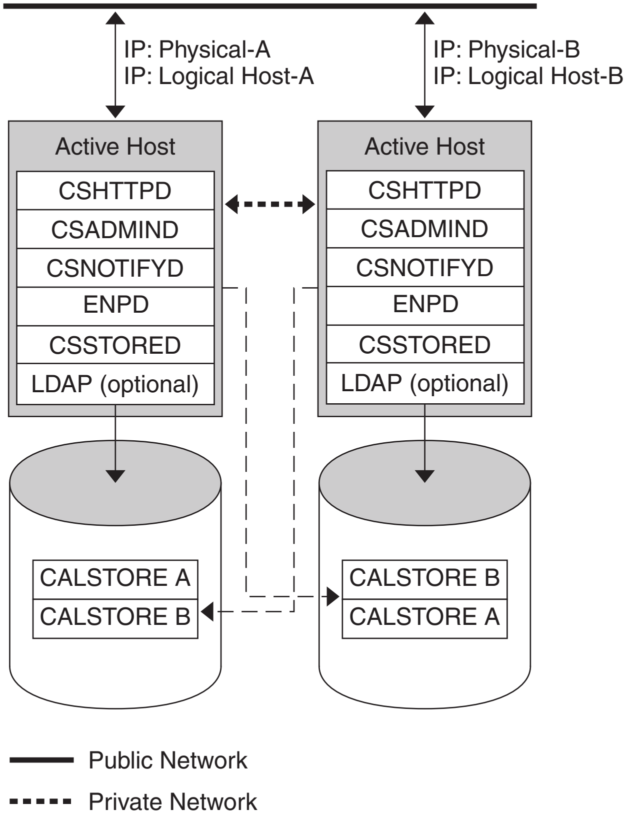
A simple asymmetric high availability system has two physical nodes. The primary node is usually active, with the other node acting as a backup node, ready to take over if the primary node fails. To accomplish a fail over, the shared disk array is switched so that it is mastered by the backup node. The Calendar Server processes are stopped on the failing primary node and started on the backup node.

There are several advantages of this type of high availability system. One advantage is that the backup node is dedicated and completely reserved for the primary node. This means there is no resource contention on the backup node when a failover occurs. Another advantage is the ability to perform a *rolling upgrade*; that is, you can upgrade one node while continuing to run Calendar Server software on the other node. Changes you make to the `ics.conf` file while upgrading the first node will not interfere with the other instance of Calendar Server software running on the secondary node because the configuration file is read only once, at startup. You must stop and restart the calendar processes before the new configuration takes effect. When you want to upgrade the other node, you perform a failover to the upgraded primary node and proceed with the upgrade on the secondary node.

Note – You can, of course, choose to upgrade the secondary node first, and then the primary node.

The asymmetric high availability model also has some disadvantages. One disadvantage is that the backup node stays idle most of the time, making this resource underutilized. Another possible disadvantage is the single storage array. In the event of a disk array failure with a simple asymmetric high availability system, no backup is available

6.1.2 Understanding Symmetric High Availability for Calendar Server Version 6.3



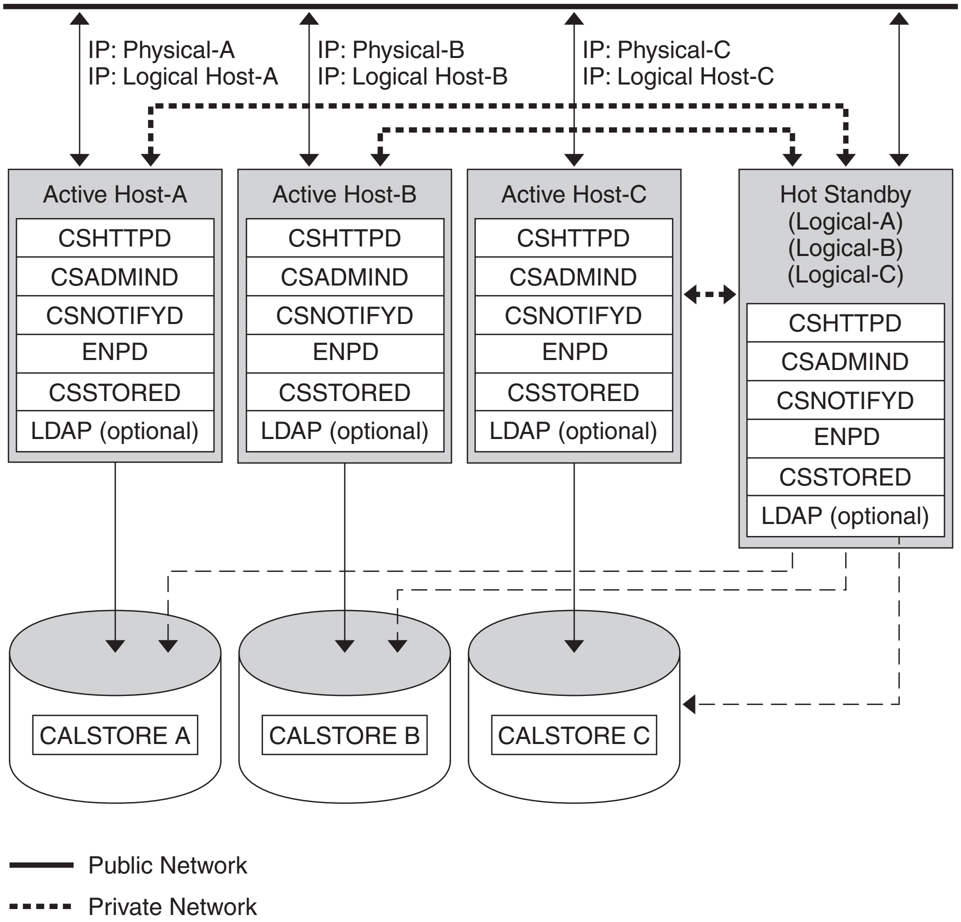
A simple symmetric high availability system has two active physical nodes, each with its own disk array with two storage volumes, one volume for the local calendar store, and the other a mirror image of the other node's calendar store. Each node acts as the backup node for the other. When one node fails over to its backup, two instances of Calendar Server run concurrently on the backup node, each running from its own installation directory and accessing its own calendar store. The only thing shared is the computing power of the back up node.

The advantage of this type of high availability system is that both nodes are active simultaneously, thus fully utilizing machine resources. However, during a failure, the backup node will have more resource contention as it runs services for Calendar Server from both nodes.

Symmetric high availability also provides a backup storage array. In the event of a disk array failure, its redundant image can be picked up by the service on its backup node.

Note – To configure a symmetric high availability system, you install the Calendar Server binaries on your shared disk. Doing so might prevent you from performing rolling upgrades, a feature planned for future releases of Calendar Server that enables you to update your system with a Calendar Server patch release with minimal or no down time.

6.1.3 Understanding N+1 (N Over 1): Multiple Asymmetric High Availability for Calendar Server Version 6.3



In addition to the two types of highly available systems described in this chapter, a third type which is a hybrid of the two is also possible. This is a multi-node asymmetric high availability system. In this type, “N” disk arrays and “N” nodes all use the same backup node which is held in reserve and is not active normally. This backup node is capable of running Calendar Server for any of the “N” nodes. It shares each of the “N” node's disk array, as shown in the preceding graphic. If multiple nodes fail at the same time, the backup node must be capable of running up to “N” instances of Calendar Server concurrently. Each of the “N” nodes has its own disk array.

The advantages of the N+1 model are that Calendar Server load can be distributed to multiple nodes, and that only one backup node is necessary to sustain all the possible node failures.

The disadvantage of this type of high availability is the same as any asymmetric system; the backup node is idle most of the time. In addition, the N+1 high availability system backup node must have excess capacity in the event it must host multiple instances of Calendar Server. This means a higher cost machine is sitting idle. However, the machine idle ratio is 1:N as opposed to 1:1, as is the case in a single asymmetric system.

To configure this type of system, use the instructions for the asymmetric high availability system for each of the “N” nodes and the backup. Use the same backup node each time, but with a different primary node.

6.1.4 Choosing a High Availability Model for Your Calendar Server Version 6.3 Deployment

The following table summarizes the advantages and disadvantages of each high availability model. Use this information to help you determine which model is right for your deployment.

TABLE 6–1 Advantages and Disadvantages of Both High Availability Model

Model	Advantages	Disadvantages	Recommended Users
Asymmetric	■ Simple Configuration ■ Backup node is 100 percent reserved ■ Rolling Upgrade, with zero downtime	Machine resources are not fully utilized.	A small service provider with plans to expand in the future
Symmetric	■ Better use of system resources ■ Higher availability	Resource contention on the backup node. HA requires fully redundant disks.	A small corporate deployment that can accept performance penalties in the event of a single server failure

TABLE 6–1 Advantages and Disadvantages of Both High Availability Model *(Continued)*

Model	Advantages	Disadvantages	Recommended Users
N+1	■ Load distribution ■ Easy expansion	Management and configuration complexity.	A large service provider who requires distribution with no resource constraints

6.1.5 System Down Time Calculations for High Availability in Your Calendar Server 6.3 Deployment

The following table illustrates the probability that on any given day the calendar service will be unavailable due to system failure. These calculations assume that on average, each server goes down for one day every three months due to either a system crash or server hang, and that each storage device goes down one day every 12 months. These calculations also ignore the small probability of both nodes being down simultaneously.

TABLE 6–2 System Down Time Calculations

Model	Server Down Time Probability
Single server (no high availability)	Pr(down) = (4 days of system down + 1 day of storage down)/365 = 1.37%
Asymmetric	Pr(down) = (0 days of system down + 1 day of storage down)/365 = 0.27%
Symmetric	Pr(down) = (0 days of system down + 0 days of storage down)/365 = (near 0)
N + 1 Asymmetric	Pr(down) = (5 hours of system down + 1 day of storage down)/(365xN) = 0.27%/N

6.2 Prerequisites for an HA Environment for Your Calendar Server Version 6.3 Deployment

This sections lists the prerequisites for installing Calendar Server in an HA environment.

The following prerequisites apply:

- Either the Solaris 9 or the Solaris 10 operating system must be installed on all nodes of the cluster, with required patches
- Sun Cluster 3.0 or 3.1 must be installed on all nodes of the cluster
- Calendar Server HA Agents package (`SUNWscics`) must be installed on all nodes of the cluster using the Java Enterprise System installer

- Specify local file systems as HAStoragePlus Failover File System (FFS) systems, or HAStorage Cluster File System (CFS)

Note – If you have a version of Sun Cluster 3.0 dated December 2001 or earlier, you must use the global file system, specified as a HAStorage Cluster File System (CFS).

- If logical volumes are being created, which is true for the symmetric high availability system, use either Solstice DiskSuite or Veritas Volume Manager.

6.2.1 About HAStoragePlus for a Calendar Server 6.3 HA Deployment

Use the HAStoragePlus resource type to make locally mounted file systems highly available within a Sun Cluster environment. Any file system resident on a Sun Cluster global device group can be used with HAStoragePlus. An HAStoragePlus file system is available on only one cluster node at any given point of time. These locally mounted file systems can only be used in failover mode and in failover resource groups. HAStoragePlus offers Failover File System (FFS), in addition to supporting the older Global File System (GFS), or Cluster File System (CFS).

HAStoragePlus has a number of benefits over its predecessor, HAStorage:

- HAStoragePlus bypasses the global file service layer completely. For data services requiring an intensive number of disk accesses, this leads to a significant performance increase.
- HAStoragePlus can work with any file system (like UFS, VxFS, and so forth), even those that might not work with the global file service layer. If a file system is supported by the Solaris operating system, it will work with HAStoragePlus.

Note – Use HAStoragePlus resources in a data service resource group with Sun Cluster 3.0 Release May 2002 and later.

For more information on HAStoragePlus, see *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*.

6.3 High-Level Task List for an Asymmetric High Availability Deployment with Calendar Server 6.3 Software

The following is a list of the tasks necessary to install and configure Calendar Server for Asymmetric High Availability:

1. Prepare the nodes.

   a. Install the Solaris Operating System software on all nodes of the cluster.
   b. Install Sun Cluster software on all nodes of the cluster.
   c. Install the Calendar Server HA Agents package, `SUNWscics`, on all nodes of the cluster using the Java Enterprise System installer
   d. Create a file system on the shared disk.
   e. Install Calendar Server on the Primary and Secondary nodes of the cluster, using the Communications Suite 5 installer.
2. Run the Directory Preparation Script, `comm_dssetup.pl` on the machine where the Directory Server LDAP directory resides.
3. Installing and configuring the first (primary) node.
   a. Using the Sun Cluster command-line interface, set up HA on the primary node.
   b. Run the Calendar Server configuration program, `csconfigurator.sh`, on the primary node.
   c. Using the Sun Cluster command-line interface, switch to the secondary node.
4. Create a symbolic link from the Calendar Server `config` directory on the primary node to the shared disk `config` directory.
5. Install and configure the second (secondary) node.
   a. Run the Calendar Server configuration program on the secondary node by reusing the state file created when you configured the primary node.
   b. Edit the Configuration File, `ics.conf`.
   c. Using the Sun Cluster command-line interface, configure and enable a resource group for Calendar Server.
   d. Using the Sun Cluster command-line interface to test the successful creation of the resource group, perform a fail over to the primary node.

For step-by-step instructions, see “6.6 Installing and Configuring Calendar Server 6.3 Software in an Asymmetric High Availability Environment” on page 187.

6.4 High-Level Task List for a Symmetric High Availability Deployment with Calendar Server 6.3 Software

The following is a list of the tasks necessary to install and configure Calendar Server for Symmetric High Availability:

1. Prepare the nodes.
   a. Install the Solaris Operating System software on all nodes of the cluster.
   b. Install Sun Cluster software on all nodes of the cluster.

   c. Create six file systems, either Cluster File Systems (Global File systems) or Fail Over File Systems (Local File systems).

   d. Create the necessary directories.

   e. Install the Calendar Server HA Agents package, `SUNWscics`, on all nodes of the cluster using the Java Enterprise System installer

2. Install and Configure the first node.

   a. Using the Communications Suite 5 installer, install Calendar Server on the first node of the cluster.

   b. Run the Directory Preparation Script, `comm_dssetup.pl`, on the machine where the Directory Server LDAP database resides.

   Note – If the instances of Calendar Server on the two nodes share the same LDAP server, it is not necessary to repeat this step after installing Calendar Server software on the second node.

   ---

   c. Using the Sun Cluster command-line interface, configure HA on the first node.

   d. Run the Calendar Server configuration program, `csconfigurator.sh`, on the first node.

   e. Using the Sun Cluster command-line interface, fail over to the second node.

   f. Edit the Configuration File, `ics.conf`, on the first node.

   g. Using the Sun Cluster command-line interface, configure and enable a resource group for Calendar Server on the first node.

   h. Using the Sun Cluster command-line interface, create and enable a resource group for the first node.

   i. Using the Sun Cluster command-line interface to test the successful creation of the resource group, perform a fail over to the first node.

3. Install and configure the second node.

   a. Using the Communications Suite 5 installer, install Calendar Server on the second node of the cluster.

   b. Using the Sun Cluster command-line interface, configure HA on the second node.

   c. Run the Calendar Server configuration program, `csconfigurator.sh`, on the second node by reusing the state file created when you configured the first node.

   d. Using the Sun Cluster command-line interface, fail over to the first node.

   e. Edit the Configuration File, `ics.conf`, on the second node.

   f. Using the Sun Cluster command-line interface, create and enable a resource group for Calendar Server on the second node.

   g. Using the Sun Cluster command-line interface to test the successful creation of the resource group, perform a fail over to the second node.

For step-by-step instructions, see "6.7 Configuring a Symmetric High Availability Calendar Server System" on page 193.

6.5 Naming Conventions for All Examples in this Deployment Example for Configuring High Availability in Calendar Server Version 6.3

Tip – Print out this section and record the values you use as you go through the HA installation and configuration process.

This section contains four tables showing the variable names used in all examples:

- Table 6–3 Directory Name Variables Used in Asymmetric Examples
- Table 6–4 Directory Name Variables Used in Symmetric Examples
- Table 6–5 Resource Name Variables for Asymmetric Examples
- Table 6–6 Resource Name Variables for Symmetric Examples
- Table 6–7 Variable Name for IP Address in Asymmetric Examples
- Table 6–8 Variable Name for IP Address in Symmetric Examples

TABLE 6–3 Directory Name Variables Used in Asymmetric Examples

Example Name	Directory	Description
`install-root`	`/opt`	The directory in which Calendar Server is installed.
`cal-svr-base`	`/opt/SUNWics5/cal`	The directory in which all Calendar Server files are located.
`var-cal-dir`	`/var/opt/SUNWics5`	The */var* directory.
`share-disk-dir`	`/cal`	A global directory; that is, a directory shared between nodes in an asymmetric high availability system.

TABLE 6–4 Directory Name Variables Used in Symmetric Examples

Example Name	Directory	Description
`install-rootCS1` `install-rootCS2`	`/opt/Node1` `/opt/Node2`	The directory in which an instance of Calendar Server is installed.
`cal-svr-baseCS1` `cal-svr-baseCS2`	`/opt/Node1/SUNWics5/cal` `/opt/Node2/SUNWics5/cal`	The directory in which all Calendar Server files are located for the node.
`var-cal-dirCS1` `var-cal-dirCS2`	`/var/opt/Node1/SUNWics5` `/var/opt/Node2/SUNWics5`	The */var* directories for each node.
`share-disk-dirCS1` `share-disk-dirCS2`	`/cal/Node1` `/cal/Node2`	The global (shared) directories each instance of Calendar Server shares with its fail over node. This is used in a symmetric high availability system.

TABLE 6–5 Resource Name Variables for Asymmetric Examples

Variable Name	Description
`CAL-RG`	A calendar resource group.
`LOG-HOST-RS`	A logical hostname resource.
`LOG-HOST-RS-Domain.com`	The fully qualified logical hostname resource.
`CAL-HASP-RS`	An HAStoragePlus resource.
`CAL-SVR-RS`	A Calendar Server resource group.

TABLE 6–6 Resource Name Variables for Symmetric Examples

Variable Name	Description
`CAL-CS1-RG`	A calendar resource group for the first instance of Calendar Server.
`CAL-CS2-RG`	A calendar resource group for the second instance of Calendar Server.
`LOG-HOST-CS1-RS`	A logical hostname resource for the first instance of Calendar Server.
`LOG-HOST-CS1-RS-Domain.com`	The fully qualified logical hostname resource for the first instance of Calendar Server.
`LOG-HOST-CS2-RS`	A logical hostname resource for the second instance of Calendar Server.
`LOG-HOST-CS2-RS-Domain.com`	The fully qualified logical hostname resource for the second instance of Calendar Server.

TABLE 6–6 Resource Name Variables for Symmetric Examples *(Continued)*

Variable Name	Description
`CAL-HASP-CS1-RS`	An HAStoragePlus resource for the first instance of Calendar Server.
`CAL-HASP-CS2-RS`	An HAStoragePlus resource for the second instance of Calendar Server.
`CAL-SVR-CS1-RS`	A Calendar Server resource group for the first instance of Calendar Server.
`CAL-SVR-CS2-RS`	A Calendar Server resource group for the second instance of Calendar Server.

TABLE 6–7 Variable Name for IP Address in Asymmetric Examples

Logical IP Address	Description
`IPAddress`	The IP Address of the port on which the `chsttpd` daemon will listen. It should be in the standard IP format, for example: "123.45.67.890"

TABLE 6–8 Variable Name for IP Address in Symmetric Examples

Logical IP Address	Description
`IPAddressCS1`	The IP Address of the port on which the `chsttpd` daemon for the first instance of Calendar Server will listen. It should be in the standard IP format, for example: "123.45.67.890"
`IPAddressCS2`	The IP Address of the port on which the `chsttpd` daemon for the second instance of Calendar Server will listen. It should be in the standard IP format, for example: "123.45.67.890"

6.6 Installing and Configuring Calendar Server 6.3 Software in an Asymmetric High Availability Environment

This section contains instructions for configuring an asymmetric high availability Calendar Server cluster.

This sections contains the following topics:

- "6.6.1 Creating the File Systems for Your Calendar Server 6.3 HA Deployment" on page 188
- "6.6.2 Creating the Calendar Directory on All Shared Disks of the Cluster in Your Calendar Server 6.3 HA Deployment" on page 188
- "6.6.3 Installing and Configuring High Availability for Calendar Server 6.3 Software" on page 188

6.6.1 Creating the File Systems for Your Calendar Server 6.3 HA Deployment

Create a file system on the shared disk. The /etc/vfstab should be identical on all the nodes of the cluster.

For CFS, it should look similar to the following example.

```
## Cluster File System/Global File System ##
/dev/md/penguin/dsk/d400 /dev/md/penguin/rdsk/d400 /cal ufs 2 yes global,logging
```

For example, for FFS:

```
## Fail Over File System/Local File System ##
/dev/md/penguin/dsk/d400 /dev/md/penguin/rdsk/d400 /cal ufs 2 no logging
```

Note – The fields in these commands are separated by tabs, not just spaces.

6.6.2 Creating the Calendar Directory on All Shared Disks of the Cluster in Your Calendar Server 6.3 HA Deployment

For all nodes of the cluster, create a directory, /Cal, on the shared disk where configuration and data is held. For example, do the following command for every shared disk:

```
mkdir -P /Cal
```

6.6.3 Installing and Configuring High Availability for Calendar Server 6.3 Software

This section contains instructions for the tasks involved in installing and configuring high availability for Calendar Server.

Perform each of the following tasks in turn to complete the configuration:

- “To Prepare Each Node of the Cluster” on page 189
- “To Set Up the Primary Node” on page 189
- “To Run the Configuration Utility (csconfigurator.sh) on the Primary Node” on page 190
- “To Configure the Secondary Node” on page 191

▼ To Prepare Each Node of the Cluster

1 Install Calendar Server on the Primary and Secondary nodes of the cluster, using the Communications Suite 5 installer.

Note – Be sure to specify the same installation root on all nodes.

a. At the Specify Installation Directories panel, answer with the installation root for both nodes.

This will install the Calendar Server binaries in the following directory:`/install-root/SUNWics5/cal`. This directory is called the Calendar Server base (*cal-svr-base*).

b. Choose the Configure Later option.

c. After the installation is complete, verify that the files are installed.

```
# pwd
/cal-svr-base

# ls -rlt

total 16
drwxr-xr-x   4 root     bin          512 Dec 14 12:52 share
drwxr-xr-x   3 root     bin          512 Dec 14 12:52 tools
drwxr-xr-x   4 root     bin         2048 Dec 14 12:52 lib
drwxr-xr-x   2 root     bin         1024 Dec 14 12:52 sbin
drwxr-xr-x   8 root     bin          512 Dec 14 12:52 csapi
drwxr-xr-x  11 root     bin         2048 Dec 14 12:52 html
```

2 Run the Directory Preparation Script (`comm_dssetup.pl`) against your existing Directory Server LDAP.

This prepares your Directory Server by setting up new LDAP schema, index, and configuration data.

For instructions and further information about running `comm_dssetup.pl`, see Chapter 8, "Directory Preparation Tool (comm_dssetup.pl)," in *Sun Java Communications Suite 5 Installation Guide*.

▼ To Set Up the Primary Node

Use the Sun Cluster command line interface as indicated to set up HA on the first node.

Note – Refer to "6.5 Naming Conventions for All Examples in this Deployment Example for Configuring High Availability in Calendar Server Version 6.3 " on page 185 as a key for directory names and Sun Cluster resource names in the examples.

1 **Register the Calendar Server and HAStoragePlus resource**

```
./scrgadm -a -t SUNW.HAStoragePlus
./scrgadm -a -t SUNW.scics
```

2 **Create a failover Calendar Server resource group.**

For example, the following instruction creates the calendar resource group `CAL-RG` with the primary node as `Node1` and the secondary, or failover, node as `Node2`.

```
./scrgadm -a -g CAL-RG -h node1,node2
```

3 **Create a logical hostname resource in the Calendar Server resource group and bring the resource group online.**

For example, the following instructions create the logical hostname resource *LOG-HOST-RS*, and then brings the resource group *CAL-RG* online.

```
./scrgadm -a -L -g CAL-RG -l LOG-HOST-RS
./scrgadm -c -j LOG-HOST-RS -y    \
      R_description="LogicalHostname resource for LOG-HOST-RS"
./scswitch -Z -g CAL-RG
```

4 **Create and enable the HAStoragePlus resource.**

For example, the following instructions create and enable the HAStoragePlus resource *CAL-HASP-RS*.

```
scrgadm -a -j CAL-HASP-RS -g CAL-RG -t
     SUNW.HAStoragePlus:4 -x FilesystemMountPoints=/cal
scrgadm -c -j CAL-HASP-RS -y
     R_description="Failover data service resource for SUNW.HAStoragePlus:4"
scswitch -e -j CAL-HASP-RS
```

▼ To Run the Configuration Utility (csconfigurator.sh) on the Primary Node

1 **Run the configuration program.**

For example, from the `/cal-svr-base/sbin` directory:

```
# pwd
     /cal-svr-base/sbin


# ./csconfigurator.sh
```

For further information about running the configuration script, see Chapter 2, "Initial Runtime Configuration Program for Calendar Server 6.3 software (csconfigurator.sh)"also in this guide.

2 **At the Run Time Configuration panel, deselect both Calendar Server startup options.**

3 **At the Directories panel, configure all directories on a shared disk. Use the following locations:**

Config Directory	`/share-disk-dir/config`
Database Directory	`/share-disk-dir/csdb`
Attachment Store Directory	`/share-disk-dir/store`
Logs Directory	`/share-disk-dir/logs`
Temporary Files Directory	`/share-disk-dir/tmp`

Once you have finished specifying the directories, choose Create Directory.

4 **At the Archive and Hot Backup panel, specify the following choices:**

Archive Directory	`/share-disk-dir/csdb/archive`
Hot Backup Directory	`/share-disk-dir/csdb/hotbackup`

When you have finished specifying the directories, choose the Create Directory option.

5 **Verify that the configuration is successful.**

Look at the end of the configuration output to make sure it says: "All Tasks Passed." The following example shows the last part of the configuration output.

```
...
All Tasks Passed. Please check install log
/var/sadm/install/logs/Sun_Java_System_Calendar_Server_install.B12141351
 for further details.
```

For a larger sample of the output, see "6.11 Example Output from the Calendar Configuration Program (Condensed)" on page 205

6 **Click Next to finish configuration.**

▼ To Configure the Secondary Node

1 **Switch to the secondary node.**

Using the Sun Cluster command line interface, switch to the secondary node. For example, the following command switches the resource group to the secondary (failover) node, *Node2*:

```
scswitch -z -g CAL-RG -h Node2
```

2 **Create a symbolic link from the Calendar Server `config` directory to the `config` directory of the Shared File System.**

For example, perform the following commands:

```
# pwd
/cal-svr-base

# ln -s /share-disk-dir/config .
```

Note – Do not forget the dot (.) at the end of the `ln` command.

---

3 **Configure Calendar Server on the secondary node using the state file from the primary node configuration.**

Share the configuration of the primary node by running the state file created when you ran the configuration program.

For example, run the following command:

```
# /cal-svr-base/sbin/csconfigurator.sh -nodisplay -noconsole -novalidate
```

Check that all the tasks passed as with the first time you ran the configuration program.

4 **Edit the Configuration File (ics.conf)**

Edit the `ics.conf` file by adding the following parameters to the end of the file. The logical hostname of the calendar resource is *LOG-HOST-RS*.

Note – Back up your `ics.conf` file before performing this step.

```
! The following are the changes for making Calendar Server
! Highly Available
!
local.server.ha.enabled="yes"
local.server.ha.agent="SUNWscics"
service.http.listenaddr="IPAddress"
local.hostname="LOG-HOST-RS"
local.servername="LOG-HOST-RS"
service.ens.host="LOG-HOST-RS"
service.http.calendarhostname="LOG-HOST-RS-Domain.com"
local.autorestart="yes"
service.listenaddr="IPAddress"
```

5 **Create the Calendar Server resource group and enable it.**

For this example, the resource group name is *CAL-SVR-RS*. You will also be required to supply the logical host resource name and the HAStoragePlus resource name.

```
./scrgadm -a -j CAL-SVR-RS -g CAL-RG
     -t SUNW.scics -x ICS_serverroot=/cal-svr-base
     -y Resource_dependencies=CAL-HASP-RS,LOG-HOST-RS

./scrgadm -e -j CAL-SVR-RS
```

6 **Test the successful creation of the calendar resource group by performing a fail over.**

```
./ scswitch -z -g CAL-RG -h Node1
```

When you have finished this step, you have completed the creation and configuration of the asymmetric high availability system for Calendar Server. The section that follows explains how to set up logging on Sun Cluster for debug purposes.

You have now finished the installation and configuration of an asymmetric Calendar Server HA system.

6.7 Configuring a Symmetric High Availability Calendar Server System

This section contains instructions for configuring a symmetric high availability Calendar Server system

To configure a symmetric high availability Calendar Server system, follow the instructions in the following sections:

- "6.7.1 Initial Tasks" on page 193
- "6.7.2 Installing and Configuring the First Instance of Calendar Server" on page 195
- "6.7.3 Installing and Configuring the Second Instance of Calendar Server" on page 199

6.7.1 Initial Tasks

There are two preparatory tasks that must be completed before installing Calendar Server on the nodes.

The preparatory tasks are as follows:

- "Creating the File Systems" on page 194
- "6.7.1.1 Installing the Calendar Server HA Package" on page 195

Note – In various places in the examples, you need to provide the installation directory (*cal-svr-base*) for each node. For a symmetric HA system, the *cal-svr-base* is different than the asymmetric HA system. For symmetric HA systems, *cal-svr-base* has the following format: `/opt/node/SUNWics5/cal`, where */opt/node* is the name of the root directory in which Calendar Server is installed (*install-root*).

For the purposes of the examples, and to differentiate the installation directories of the two Calendar Server instances, they are designated as *cal-svr-baseCS1* and *cal-svr-baseCS1*.

To differentiate the installation roots for the two Calendar Server instances in this example, they are designated as *install-rootCS1* and *install-rootCS2:*

▼ Creating the File Systems

1 **Create six file systems, using either Cluster File Systems (Global File systems) or Fail Over File Systems (Local File systems).**

This example is for Global File Systems. The contents of the `/etc/vfstab` file should look like the following: (Note that the fields are all tab separated.)

```
# Cluster File System/Global File System ##
/dev/md/penguin/dsk/d500  /dev/md/penguin/rdsk/d500
    /cal-svr-baseCS1  ufs  2  yes  logging,global
/dev/md/penguin/dsk/d400  /dev/md/penguin/rdsk/d400
    /share-disk-dirCS1  ufs  2  yes  logging,global
/dev/md/polarbear/dsk/d200  /dev/md/polarbear/rdsk/d200
    /cal-svr-baseCS2  ufs  2  yes  logging,global
/dev/md/polarbear/dsk/d300  /dev/md/polarbear/rdsk/d300
    /share-disk-dirCS2  ufs  2  yes logging,global
/dev/md/polarbear/dsk/d600  /dev/md/polarbear/rdsk/d300
    /var-cal-dirCS1  ufs  2   yes  logging,global
/dev/md/polarbear/dsk/d700  /dev/md/polarbear/rdsk/d300
    /var-cal-dirCS2  ufs   2   yes  logging,global
```

This example is for the Failover File Systems. The contents of the `/etc/vfstab` file should look like the following: (Note that the fields are all tab separated.)

```
# Failover File System/Local File System ##
/dev/md/penguin/dsk/d500  /dev/md/penguin/rdsk/d500
    /cal-svr-baseCS1  ufs  2  yes  logging
/dev/md/penguin/dsk/d400  /dev/md/penguin/rdsk/d400
    /share-disk-dirCS1  ufs  2  yes  logging
/dev/md/polarbear/dsk/d200  /dev/md/polarbear/rdsk/d200
   /cal-svr-baseCS2  ufs  2  yes  logging
/dev/md/polarbear/dsk/d300  /dev/md/polarbear/rdsk/d300
    /share-disk-dirCS2  ufs  2  yes  logging
```

```
/dev/md/polarbear/dsk/d600  /dev/md/polarbear/rdsk/d300
    /var-cal-dirCS1  ufs  2   yes  logging
/dev/md/polarbear/dsk/d700  /dev/md/polarbear/rdsk/d300
   /var-cal-dirCS2  ufs  2   yes  logging
```

2 **Create the following required directories on all nodes of the cluster.**

```
# mkdir -p /install-rootCS1 share-disk-dirCS1
     install-rootCS2 share-disk-dirCS2 var-cal-dirCS1
     var-cal-dirCS2
```

6.7.1.1 Installing the Calendar Server HA Package

Install the Calendar Server HA package, `SUNWscics`, on all nodes of the cluster.

This must be done from the Java Enterprise System installer.

For more information about the Java Enterprise System installer, refer to the *Sun Java Enterprise System 5 Installation and Configuration Guide*.

6.7.2 Installing and Configuring the First Instance of Calendar Server

Follow the instructions in this section to install and configure the first instance of Calendar Server. This section covers the following topics:

▼ To Install Calendar Server

1 **Verify the files are mounted.**

On the primary node (`Node1`), enter the following command:

```
df -k
```

The following is an example of the output you should see:

```
/dev/md/penguin/dsk/d500     35020572
     34738 34635629   1%   /install-rootCS1
/dev/md/penguin/dsk/d400     35020572
     34738 34635629   1%   /share-disk-dirCS1
/dev/md/polarbear/dsk/d300   35020572
     34738 34635629   1%   /share-disk-dirCS2
/dev/md/polarbear/dsk/d200   35020572
     34738 34635629   1%   /install-rootCS2
/dev/md/polarbear/dsk/d600   35020572
     34738 34635629   1%   /var-cal-dirCS1
```

```
/dev/md/polarbear/dsk/d700   35020572
     34738 34635629   1%   /var-cal-dirCS2
```

2 **Using the Sun Java Systems Communications Suite installer, install Calendar Server on the Primary Node.**

a. **At the Specify Installation Directories panel, specify the installation root (install-rootCS1):**

For example, if your Primary node is named *red* and the root directory is *dawn*, the installation root would be `/dawn/red`. This is the directory where you are installing Calendar Server on the first node.

b. **Choose Configure Later.**

3 **Run the Directory Preparation Tool script on the machine with the Directory Server.**

▼ To Configure Sun Cluster on the First Node

Using the Sun Cluster command-line interface, configure Sun Cluster on the first node by performing the following steps:

1 **Register the following resource types:**

```
./scrgadm -a -t SUNW.HAStoragePlus
./scrgadm -a -t SUNW.scics
```

2 **Create a fail over resource group.**

In the following example, the resource group is *CAL-CS1-RG*, and the two nodes are named *Node1* as the primary node and *Node2* as the fail over node.

```
./scrgadm -a -g CAL-CS1-RG -h Node1,Node2
```

3 **Create a logical hostname resource for this node.**

The calendar client listens on this logical hostname. The example that follows uses *LOG-HOST-CS1-RS* in the place where you will substitute in the actual hostname.

```
./scrgadm -a -L -g CAL-RG -l LOG-HOST-CS1-RS
./scrgadm -c -j LOG-HOST-CS1-RS -y R_description=
     "LogicalHostname resource for LOG-HOST-CS1-RS"
```

4 **Bring the resource group online.**

```
scswitch -Z -g CAL-CS1-RG
```

5 **Create an HAStoragePlus resource and add it to the fail over resource group.**

In this example, the resource is called *CAL-HASP-CS1-RS*. You will substitute you own resource name. Note that the lines are cut and show as two lines in the example for display purposes in this document.

```
./scrgadm -a -j CAL-HASP-CS1-RS -g CAL-CS1-RG -t
     SUNW.HAStoragePlus:4 -x FilesystemMountPoints=/install-rootCS1,
     /share-disk-dirCS1,/cal-svr-baseCS1
./scrgadm -c -j CAL-HASP-CS1-RS -y R_description="Failover data
     service resource for SUNW.HAStoragePlus:4"
```

6 **Enable the HAStoragePlus resource.**

```
./scswitch -e -j CAL-HASP-CS1-RS
```

▼ To Configure the First Instance of Calendar Server

1 **Run the configuration program on the primary node.**

```
# cd /cal-svr-baseCS1/sbin/

# ./csconfigurator.sh
```

For further information about running the configuration script, see the *Sun Java System Calendar Server 6.3 Administration Guide*.

2 **On the Runtime Configuration panel, deselect both of the Calendar Server start up options.**

3 **On the Directories to Store Configuration and Data Files panel, provide the shared disk directories as shown in the following list:**

Config Directory	`/share-disk-dirCS1/config`
Database Directory	`/share-disk-dirCS1/csdb`
Attachment Store Directory	`/share-disk-dirCS1/store`
Logs Directory	`/share-disk-dirCS1/logs`
Temporary Files Directory	`/share-disk-dirCS1/tmp`

When you have finished specifying the directories, choose Create Directory.

4 **On the Archive and Hot Backup panel, provide the shared disk directory names as shown in the following list:**

Archive Directory	`/share-disk-dirCS1/csdb/archive`
Hot Backup Directory	`/share-disk-dirCS1/csdb/hotbackup`

After specifying these directories, choose Create Directory.

5 **Verify that the configuration was successful.**

The configuration program will display a series of messages. If they all start with *PASSED*, which means it was successful. For an example of the output you might see, check the example at: "6.11 Example Output from the Calendar Configuration Program (Condensed)" on page 205.

▼ To Perform the Final Configuration Steps for the First Instance

1 **Using the Sun Cluster command-line interface, perform a fail over to the second node.**

For example:

```
# /usr/cluster/bin/scswitch -z -g CAL-CS1-RG -h Node2
```

2 **Edit the configuration file, `ics.conf`, by adding the parameters shown in the example that follows.**

Note – Back up the `ics.conf` file before starting this step.

```
! The following changes were made to configure Calendar Server
! Highly Available
!
local.server.ha.enabled="yes"
local.server.ha.agent="SUNWscics"
service.http.listenaddr="IPAddressCS1"
local.hostname="LOG-HOST-CS1-RS"
local.servername="LOG-HOST-CS1-RS"
service.ens.host="LOG-HOST-CS1-RS"
service.http.calendarhostname="LOG-HOST-CS1-RS-Domain.com"
local.autorestart="yes"
service.listenaddr = "IPAddressCS1"
```

Note – The expected value for `service.http.calendarhostname` is a fully qualified hostname.

---

3 **Using the Sun Cluster command-line interface, create the Calendar Server resource group.**

Create a calendar resource group and enable it.

For example:

```
./scrgadm -a -j CAL-SVR-CS1-RS -g CAL-CS1-RG
      -t SUNW.scics  -x ICS_serverroot=/cal-svr-baseCS1
      -y Resource_dependencies=CAL-HASP-CS1-RS,LOG-HOST-CS1-RS

./scrgadm -e -j CAL-SVR-CS1-RS
```

4 **Using the Sun Cluster command-line interface to test the successful creation of the Calendar Server resource group, perform a fail over to the first node, which is the Primary node.**

For example:

```
./scswitch -z -g CAL-CS1-RG -h Node1
```

6.7.3 Installing and Configuring the Second Instance of Calendar Server

The primary node for the second Calendar Server instance is the second node (*Node2*).

▼ To Install Calendar Server on the Second Node

1 **Verify the files are mounted.**

On the primary node (`Node2`), enter the following command:

```
df -k
```

The following is an example of the output you should see:

```
/dev/md/penguin/dsk/d500     35020572
     34738 34635629   1%   /install-rootCS1
/dev/md/penguin/dsk/d400     35020572
     34738 34635629   1%   /share-disk-dirCS1
/dev/md/polarbear/dsk/d300   35020572
     34738 34635629   1%   /share-disk-dirCS2
/dev/md/polarbear/dsk/d200   35020572
     34738 34635629   1%   /install-rootCS2
/dev/md/polarbear/dsk/d600   35020572
     34738 34635629   1%   /var-cal-dirCS1
/dev/md/polarbear/dsk/d700   35020572
     34738 34635629   1%   /var-cal-dirCS2
```

2 **Using the Sun Java Systems Communications Suite installer, install Calendar Server on the new Primary Node (second node).**

a. **At the Specify Installation Directories panel, specify the installation root for the second node (/install-rootNode2):**

For example, if your Node 2 machine is named `blue` and your root directory is `ocean`, your installation directory would be `/ocean/blue`.

b. **Select the Configure Later option.**

▼ To Configure Sun Cluster for the Second Instance

Using the Sun Cluster command-line interface, configure the second instance of Calendar Server as described in the following steps:

1. **Create a fail over resource group.**

   In the following example, the resource group is *CAL-CS2-RG*, and the two nodes are named *Node2* as the primary node and *Node1* as the fail over node.

   ```
   ./scrgadm -a -g CAL-CS2-RG -h Node2,Node1
   ```

2. **Create a logical hostname resource.**

   The calendar client listens on this logical hostname. The example that follows uses *LOG-HOST-CS2-RS* in the place where you will substitute in the actual hostname.

   ```
   ./scrgadm -a -L -g CAL-CS2-RG -l LOG-HOST-CS2-RS
   ./scrgadm -c -j LOG-HOST-CS2-RS -y R_description="LogicalHostname
        resource for LOG-HOST-CS2-RS"
   ```

3. **Bring the resource group online.**

   ```
   scswitch -Z -g CAL-CS2-RG
   ```

4. **Create an HAStoragePlus resource and add it to the fail over resource group.**

   In this example, the resource is called *CAL-SVR-CS2-RS*. You will substitute you own resource name.

   ```
   ./scrgadm -a -j CAL-SVR-CS2-RS -g CAL-CS2-RG -t
        SUNW.HAStoragePlus:4 -x FilesystemMountPoints=/install-rootCS2,
        /share-disk-dirCS2,/var-cal-dirCS2
   ./scrgadm -c -j CAL-HASP-CS2-RS -y R_description="Failover data
        service resource for SUNW.HAStoragePlus:4"
   ```

5. **Enable the HAStoragePlus resource.**

   ```
   ./scswitch -e -j CAL-HASP-CS2-RS
   ```

▼ To Configure the Second Instance of Calendar Server

1. **Run the configuration program again on the secondary node.**

   ```
   # cd /cal-svr-baseCS2/sbin/

   # ./csconfigurator.sh
   ```

   For further information about running the configuration script, see the *Sun Java System Calendar Server 6.3 Administration Guide*.

2. **On the Runtime Configuration panel, deselect both of the Calendar Server start up options.**

3 **On the Directories to Store Configuration and Data Files panel, provide the proper directories as shown in the following list:**

Config Directory	share-disk-dirCS2/config
Database Directory	/share-disk-dirCS2/csdb
Attachment Store Directory	/share-disk-dirCS2/store
Logs Directory	/share-disk-dirCS2/logs
Temporary Files Directory	/share-disk-dirCS2/tmp

When you have finished specifying the directories, choose Create Directory.

4 **On the Archive and Hot Backup panel, provide the appropriate directory names as shown in the following list:**

Archive Directory	/share-disk-dirCS2/csdb/archive
Hot Backup Directory	/share-disk-dirCS2/csdb/hotbackup

After specifying these directories, choose Create Directory.

5 **Verify that the configuration was successful.**

The configuration program will display a series of messages. If they all start with *PASSED*, which means it was successful. For an example of the output you might see, check the example at: "6.11 Example Output from the Calendar Configuration Program (Condensed)" on page 205.

▼ To Perform the Final Configuration Steps for the Second Instance

1 **Using the Sun Cluster command-line interface, perform a fail over to the first node.**

For example:

```
# /usr/cluster/bin/scswitch -z -g CAL-CS2-RG -h Node1
```

2 **Edit the configuration file,** ics.conf**by adding the parameters shown in the example that follows.**

Note – The values shown are examples only. You must substitute your own information for the values in the example.

Back up the ics.conf file before starting this step.

```
! The following changes were made to configure Calendar Server
! Highly Available
!
local.server.ha.enabled="yes"
local.server.ha.agent="SUNWscics"
service.http.listenaddr="IPAddressCS2"
local.hostname="LOG-HOST-CS2-RS"
local.servername="LOG-HOST-CS2-RS"
service.ens.host="LOG-HOST-CS2-RS"
service.http.calendarhostname="LOG-HOST-CS2-RS-Domain.com"
local.autorestart="yes"
service.listenaddr = "IPAddressCS2"
```

Note – The value for `service.http.calendarhostname` must be a fully qualified hostname.

3 **Using the Sun Cluster command-line interface, create a Calendar Server resource group.**

Create a Calendar Server resource group and enable it.

For example:

```
./scrgadm -a -j CAL-SVR-CS2-RS -g CAL-CS2-RG
      -t SUNW.scics -x ICS_serverroot=/cal-svr-baseCS2
      -y Resource_dependencies=CAL-HASP-CS2-RS,LOG-HOST-CS2-RS

./scrgadm -e -j CAL-SVR-CS2-RS
```

4 **Using the Sun Cluster command-line interface to test the successful creation of the calendar resource group, perform a fail over to the second node, which is primary node for this Calendar Server instance.**

For example:

```
./scswitch -z -g CAL-CS2-RG -h Node2
```

Your have now finished installing and configuring a symmetric HA Calendar Server.

6.8 Starting and Stopping Calendar Server HA Service

Use the following commands to start, fail over, disable, remove, and restart the Calendar Server HA service:

To enable and start the Calendar Server HA service:

```
# scswitch -e -j CAL-SVR-RS
```

To Fail Over the Calendar Server HA service:

```
# scswitch -z -g CAL-RG -h Node2
```

To disable the Calendar Server HA Service:

```
# scswitch -n -j CAL-SVR-RS
```

To remove the Calendar Server resource:

```
# scrgadm -r -j CAL-SVR-RS
```

To restart the Calendar Server HA service:

```
# scrgadm -R -j CAL-SVR-RS
```

6.9 Removing HA from Your Calendar Server Configuration

This section describes how to undo the HA configuration for Sun Cluster. This section assumes the simple asymmetric example configuration described in this chapter. You must adapt this scenario to fit your own installation.

▼ To Remove HA Components

1 **Become a superuser.**

Note – All of the following Sun Cluster commands require that you be running as a superuser.

2 **Bring the resource group offline. Use the following command to shut down all of the resources in the resource group (For example, the Calendar Server and the HA logical host name).**

```
# scswitch -F -g CAL-RG
```

3 **Disable the individual resources.**

4 **Remove the resources one-by-one from the resource group using the commands:**

```
# scswitch -n -j CAL-SVR-RS
# scswitch -n -j CAL-HASP-RS
# scswitch -n -j LOG-HOST-RS
```

5 **Remove the resource group itself using the command:**

```
# scrgadm -r -g CAL-RG
```

6 **Remove the resource types (optional). If you want to remove the resource types from the cluster, use the command:**

```
# scrgadm -r -t SUNW.scics
# scrgadm -r -t SUNW.HAStorage
```

6.10 Debugging on Sun Cluster

The Calendar Server Sun Cluster agents use two different API's to log messages:

- scds_syslog_debug() — Used by Calendar Server agents. Messages are logged at `daemon.debug` level.
- scds_syslog() — Used by Calendar Server agents and Sun Cluster data services. Messages are logged at `daemon.notice`, `daemon.info`, and `daemon.error`levels.

To Enable Logging

The following task must be done on each HA node since the `/var/adm` file can't be shared. This file is on the root partition of the individual nodes.

1. **Create a logging directory for Calendar Server agents.**

```
mkdir -p /var/cluster/rgm/rt/SUNW.scics
```

2. **Set the debug level to 9.**

```
echo 9 >/var/cluster/rgm/rt/SUNW.scics/loglevel
```

The following example shows log messages you might see in the directory. Note that, in the last line, *ICS-serverroot* is asking for the `cal-svr-base`, or installation directory.

```
Dec 11 18:24:46 mars SC[SUNW.scics,CAL-RG,cal-rs,ics_svc_start]:
     [ID 831728 daemon.debug] Groupname icsgroup exists.
Dec 11 18:24:46 mars SC[SUNW.scics,CAL-RG,LOG-HOST-RS,ics_svc_start]:
     [ID 383726 daemon.debug] Username icsuser icsgroup
Dec 11 18:24:46 mars SC[SUNW.scics,CAL-RG,LOG-HOST-RS,ics_svc_start]:
     [ID 244341 daemon.debug] ICS_serverroot = /cal-svr-base
```

3. **Enable Sun Cluster Data Services Logging.**

Edit the `syslog.conf` file by adding the following line .

```
daemon.debug /var/adm/clusterlog
```

This will cause all the debug messages to be logged into the `daemon.debug` `/var/adm/clusterlog` file.

4. **Restart the syslogd daemon.**

```
pkill -HUP syslogd
```

All syslog debug messages are prefixed with the following:

```
SC[resourceTypeName, resourceGroupName, resourceName, methodName]
```

The following example messages have been split and carried over to multiple lines for display purposes.

```
Dec 11 15:55:52 Node1 SC
      [SUNW.scics,CAL-RG,CalendarResource,ics_svc_validate]:
      [ID 855581 daemon.error] Failed to get the configuration info
Dec 11 18:24:46 Node1 SC
      [SUNW.scics,CAL-RG,LOG-HOST-RS,ics_svc_start]:
      [ID 833212 daemon.info] Attempting to start the data service under
      process monitor facility.
```

6.11 Example Output from the Calendar Configuration Program (Condensed)

This section contains a partial listing of the output from the configuration program. Your output will be much longer. At the end, it should say: "All Tasks Passed." Inspect the log files. The location of the files is given at the end of the printout.

```
/usr/jdk/entsys-j2se/bin/java -cp /opt/Node2/SUNWics5/cal/share/lib:
     /opt/Node2/SUNWics5/cal/share -Djava.library.path=
     /opt/Node2/SUNWics5/cal/lib configure -nodisplay -noconsole -novalidate
# ./csconfigurator.sh -nodisplay -noconsole -novalidate
/usr/jdk/entsys-j2se/bin/java -cp /opt/Node2/SUNWics5/cal/share/lib:
     /opt/Node2/SUNWics5/cal/share -Djava.library.path=
     /opt/Node2/SUNWics5/cal/lib configure -nodisplay -noconsole -novalidate
Java Accessibility Bridge for GNOME loaded.

Loading Default Properties...

Checking disk space...

Starting Task Sequence
===== Mon Dec 18 15:33:29 PST 2006 =====
Running /bin/rm -f /opt/Node2/SUNWics5/cal/config
/opt/Node2/SUNWics5/cal/data

===== Mon Dec 18 15:33:29 PST 2006 =====
Running /usr/sbin/groupadd icsgroup

===== Mon Dec 18 15:33:29 PST 2006 =====
Running /usr/sbin/useradd -g icsgroup -d / icsuser

===== Mon Dec 18 15:33:30 PST 2006 =====
Running /usr/sbin/usermod -G icsgroup icsuser
```

```
===== Mon Dec 18 15:33:30 PST 2006 =====
Running /bin/sh -c /usr/bin/crle


===== Mon Dec 18 15:33:32 PST 2006 =====
Running /bin/chown icsuser:icsgroup /etc/opt/Node2/SUNWics5/config/watcher.
cnf


...

Sequence Completed

PASSED: /bin/rm -f /opt/Node2/SUNWics5/cal/config
/opt/Node2/SUNWics5/cal/data : status = 0

PASSED: /usr/sbin/groupadd icsgroup : status = 9

PASSED: /usr/sbin/useradd -g icsgroup -d / icsuser : status = 9


...

All Tasks Passed. Please check install log
/var/sadm/install/logs/Sun_Java_System_Calendar_Server_install.B12181533 for
further details.
```

6.12 Related Documentation

For more instruction about Sun Cluster, there are many documents that can be found at `docs.sun.com`.

The following is a partial list of documentation titles:

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

CHAPTER 7

Configuring SSL

Secure Socket Layer (SSL) is a protocol for encrypting and decrypting data across a secure connection from a client to a server with SSL capabilities. The server is responsible for sending the client, a digital certificate and a public key for encryption. If the client trusts the server's certificate, an SSL connection can be established. All data passing from one side to the other will be encrypted. Only the client and the server will be able to decrypt the data.

Sun Java System servers support the authentication of users through examination of their digital certificates. Instead of presenting a password, the client presents the user's certificate when it establishes an SSL session with the server. If the certificate is validated, the user is authenticated. Calendar Server supports the 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) `certutil` tool, which are also used by Sun Java System Messaging Server. The NSS `certutil` tool is bundled in the `sbin` directory of the Calendar Server product.

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 covers the three tasks necessary to configure SSL and troubleshooting:

Note – Calendar Server does not support client-based SSL authentication.

7.1 Configuring SSL for Calendar Server

This section contains instructions for configuring SSL for Calendar Server.

This section contains the following topics:

- "To Create a Certificate Database" on page 208
- "7.1.1 Self-signed Certificate" on page 211
- "To Request and Import a Certificate from a Root Certificate Authority" on page 212
- "To Configure SSL Parameters in the `ics.conf` File" on page 214

▼ To Create a Certificate Database

A certificate is required by the gateway to send its public keys to the clients. The certificate contains the gateway's public key, the Distinguished Name associated with the gateway's certificate, the serial number or issue date of the certificate, and the expiration date of the certificate. A certificate is issued by a certification authority (CA), which verifies the identity of the gateway. CA is an authority trusted by one or more users to issue and manage X.509 Public Key Certificates and CARLs or Certification Revocation List (CRL)s. CA is the basic building block of the Public Key Infrastructure (PKI). On the other hand, PKI is a set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.

The CA inserts its name in every certificate and CRL it generates and digitally signs the certificate with its private key. Once you establish that they trust a CA (directly, or through a certification path), you can trust certificates issued by that CA. You can easily identify certificates issued by that CA by comparing its name. However, its public key can be used to ensure that the certificate is valid.

The CA performs four basic PKI functions:

- Issues (creates and signs) certificates.
- Maintains certificate status information and issues CRL.
- Publishes its current (unexpired) certificates and CRLs
- Maintains archives of status information about the expired certificates.

Your server's certificate and key pair represent your server's identity. They are stored in a certificate database that can be either internal to the server or on an external, removable hardware card (smartcard). An SSL implementation for Calendar Server requires a certificate database. The certificate database must define a Certificate Authority (CA) and certificates for Calendar Server. This section contains conceptual and task information:

Before You Begin Before you create the certificate database, familiarize yourself with the following:

- **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 (`http://mozilla.org/projects/security/pki/nss/tools/certutil.html`)

  Tip – Familiarize yourself with the tool syntax before attempting to generate your certificate database.

- 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/lib`

  You can also download the most recent version from the Web site.

- **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:

  - `/etc/opt/SUNWics5/config` is the directory that contains the certificate database.

    Backup the certificate database regularly. You can choose to create the certificate database in another directory. If you do, you must also place the certificate password file in the same directory.

  - `sslpassword.conf` is a text file that contains the certificate database password.

    After Calendar Server is set up for SSL, Calendar Server requires the `sslpassword.conf` file to start up in SSL mode. The `certutil` utility uses a different password file. Create `sslpassword.conf` in the following directory:

    `/etc/opt/SUNWics5/config`

  - The file at `/etc/passwd` introduces entropy for random number generation, that is, this directory is used to generate varied and unique seeds that help ensure truly random results from the random number generator.

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

2 **Specify the certificate database password in** `/etc/opt/SUNWics5/config/sslpassword.conf`**.**

For example:

```
# echo "password file entry"
      /etc/opt/SUNWics5/config/sslpassword.conf
```

The format of *password file entry* in the `sslpassword.conf` file is as follows:

`Internal (Software) Token:`*password*

where *password* is your password.

Note that the password entry in the `sslpassword.conf` file must use the format shown above, including the string, `Internal (Software) Token:`. However, the password entry used with the password file associated with the `certutil -f`*passwordfile* command must use the following simple format:

password.

3 **Create the certificate database directory. For example:**

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

4 **Change to the** `bin` **directory and generate the certificate database (**`cert8.db`**) and key database (**`key3.db`**). For example:**

```
# cd /opt/SUNWics5/cal/bin
 # ./certutil -N -d /etc/opt/SUNWics5/config
                -f /mypath/mypassworfile
```

Note – For this and other times when you must run the `certutil` utility, follow the examples exactly, or consult the `certutil` help page to understand the syntax.

For example, in this case, do not run the utility with the `-N` option without also specifying the `-d /`file information.

---

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 /etc/opt/SUNWics5/config/SampleRootCA.crt
 -d /etc/opt/SUNWics5/config
 -f /mypath/mypassworfile -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 /etc/opt/SUNWics5/config/SampleSSLServer.crt
-d /etc/opt/SUNWics5/config
-f /mypath/mypassworfile
-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 /etc/opt/SUNWics5/config
 # ./certutil -V -u V -n SampleSSLServerCert
   -d /etc/opt/SUNWics5/config
```

8 **List the certificates. For example:**

```
# ./certutil -L -d /etc/opt/SUNWics5/config
 # ./certutil -L -n SampleSSLServerCert
   -d /etc/opt/SUNWics5/config
```

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

```
# ./modutil -list -dbdir /etc/opt/SUNWics5/config
```

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 /etc/opt/SUNWics5/config -exec chown icsuser {};
 # find /etc/opt/SUNWics5/config -exec chgrp icsgroup {};
```

7.1.1 Self-signed Certificate

A self-signed certificate is signed with the gateway's own private key. Self-signed certificates are not secure, but they can be used to test applications that require certificates before a signed certificate is available for use. A self-signed certificate uses its own certificate request as a signature rather than the signature of a CA.

There are ten common fields in which six are mandatory and four are optional in creating a self-signed certificate through PKI. The serial number, certificate signature algorithm identifier, certificate issuer name, certificate validity period, public key, and the subject name are the mandatory fields. The optional fields are the version number, two unique identifiers, and the extension. These optional fields appear only in version 2 and 3 certificates.

The mandatory Validity field indicates the dates on which the certificate becomes valid and the date on which the certificate expires. The default value for expiration date provided in the NSS certutils is three months. However, the validity data in a certificate become unreliable before the expiration date arrives. The X.509 CRL mechanism provides a status update for the certificates they have issued and to take care about the certificate expiration dates. Also, CA enforces certificate expiration to one or two years.

When a certificate is expired or its validity date is over, it needs to be renewed. Renewal is an act or process of extending the validity of the data binding asserted by a public key certificate by issuing a new certificate. You can validate a certificate using the command:

```
-V -n certname -b validity-time -u certusage [-e] [-l] [-d certdir]
```

The following example shows how to use the command to validate a certificate:

```
certutil -V -n jsmith@netscape.com -b 9803201212Z -u SR -e -l -d certdir.
```

The Certificate Database Tool shows results similar to the following:

```
Certificate:'jsmith@netscape.com' is valid.
```

or

```
UID=jsmith, E=jsmith@netscape.com, CN=John Smith, O=Netscape Communications
Corp., C=US : Expired certificate
```

or

```
UID=jsmith, E=jsmith@netscape.com, CN=John Smith, O=Netscape Communications
Corp., C=US : Certificate not approved for this operation
```

▼ To Request and Import a Certificate from a Root Certificate Authority

The following steps tell you how to generate a certificate request, submit it to the Public Key Infrastructure (PKI) Web site, and then import the certificate. These instruction assume you are placing the certificate database under the `config` directory.

Before You Begin Both the certificate database and the password file must reside in the same directory. The default shown here is the `config` directory, but you can choose another directory, in which case, you must configure a different path parameter, as shown in the procedure that follows.

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 /etc/opt/SUNWics5/config
-f /mypath/mypassworfile  -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 /etc/opt/SUNWics5/config
    -a
    -i /export/wspace/Certificates/CA_Certificate_1.txt
    -f /mypath/mypassworfile
# ./certutil -A -n "Sesta TEST Root CA"
    -t "TCu,TCu,TCuw"
    -d /etc/opt/SUNWics5/config
    -a
    -i /export/wspace/Certificates/CA_Certificate_2.txt
    -f /mypath/mypassworfile
```

7 **Import the signed SSL server certificate:**

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

8 **List the certificates in the certificate database:**

```
# ./certutil -L -d /etc/opt/SUNWics5/config
```

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*

▼ To Configure SSL Parameters in the `ics.conf` File

To implement SSL with Calendar Server, you must set specific parameters in the `ics.conf` file. If any of the parameters listed in the following table are not in the `ics.conf` file, add them to the file with the value specified. Since the `ics.conf` is read only at system startup (when `start-cal` is issued), the new values will not take effect until the Calendar Server is restarted. For a description of these SSL parameters, see "E.2.10 Calendar Server SSL Configuration Parameters" on page 471.

1. **Log in as an administrator with permission to change the configuration.**

2. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3. **Save your old `ics.conf` file by copying and renaming it.**

4. **Edit one or more of the parameters as shown in the following table:**

Parameter	Value
encryption.rsa.nssslactivation	`"on"`
encryption.rsa.nssslpersonalityssl	`"SampleSSLServerCert"`
encryption.rsa.nsssltoken	`"internal"`
service.http.tmpdir	`"/var/opt/SUNWics5/tmp"`
service.http.uidir.path	`"html"`
service.http.ssl.cachedir	`"."`
service.http.ssl.cachesize	`"10000"`
local.ssldbpath	`"/etc/opt/SUNWics5/config"`
service.http.ssl.port.enable	`"yes"`
service.http.ssl.port	`"443"` (Default SSL port) **Note –** Not on port `"80"`, which is the HTTP default port.
service.http.securesession	`"yes"` (Entire session encrypted)
service.http.ssl.sourceurl	`"https://localhost:port"` (Supply the name of your localhost, and the *service.http.ssl.port* value.)

Parameter	Value

service.http.ssl.ssl3.ciphers	`"rsa_red_40_md5,`
	`rsa_rc2_40_md5,`
	`rsa_des_sha,`
	`rsa_rc4_128_md5,`
	`rsa_3des_sha"`

service.http.ssl.ssl3.sessiontimeout	`"0"`
service.http.sslusessl	`"yes"`

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server for the changes to take effect.**

cal-svr-base/SUNWics5/cal/sbin/start-cal

7.2 Troubleshooting SSL for Calendar Server 6.3 Software

First, always backup your certificate database on a regular basis in case unrecoverable problems occur. This section contains things to consider after you back up your database.

- If you have problems with SSL, here are some things to consider:“7.2.1 Checking for the `cshttpd` Process” on page 215
- “7.2.2 Verifying Certificates” on page 215
- “7.2.3 Reviewing Calendar Server Log Files” on page 216
- “7.2.4 Connecting to the SSL Port” on page 216
- “7.2.5 Making cshttpd Stop Listening on the Regular HTTP Port” on page 216

7.2.1 Checking for the `cshttpd` Process

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

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

7.2.2 Verifying Certificates

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

```
# ./certutil -L -d /etc/opt/SUNWics5/config
```

7.2.3 Reviewing Calendar Server Log Files

Check the Calendar Server log files for any SSL errors.

7.2.4 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.

7.2.5 Making cshttpd Stop Listening on the Regular HTTP Port

HTTP and HTTPS listen on different ports (443 for SSL, and 80 for HTTP), so you will never have both listening to the same port. Currently, there is no way to tell `cshttpd` to stop listening to the regular HTTP port. However, an administrator can change the *service.http.port* to an undisclosed number.

Caution – Do not set *service.http.enable ="no"* in an attempt to prevent `cshttpd` from listening to HTTP. Doing so would cause HTTPS to fail also. Both *service.http.enable* and *service.http.ssl.port.enable* must be set to `"yes"` for SSL to be configured properly.

CHAPTER 8

Configuring Single Sign-on for a Calendar Server 6.3 System

This chapter describes how to configure 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 Java System communications servers, including Calendar Server and Messaging Server, can implement SSO as follows:

- "8.1 Configuring SSO Through Access Manager" on page 217
- "8.1.2 Configuring SSO Through Communications Servers Trusted Circle Technology" on page 219

8.1 Configuring SSO Through Access Manager

Sun Java Enterprise System servers, including Calendar Server and Messaging Server, can implement SSO using Sun Java System Access Manager (release 6 2003Q4 or later)

Access Manager serves as the SSO gateway for Sun Java Enterprise System servers. That is, users log in to Access Manager and then can access other Sun Java Enterprise System servers, as long as the servers are configured properly for SSO.

▼ To use SSO with Calendar Server

1 **Make sure that Access Manager and Directory Server are installed and configured. For information about installing and configuring these products, refer to the** *Sun Java Enterprise System 5 Installation Guide for UNIX***.**

2 **After stopping Calendar Server services, configure SSO for Calendar Server by setting the parameters shown in "8.1 Configuring SSO Through Access Manager" on page 217. For the values to take effect, you must restart Calendar Server services.**

Note – When you set the *local.calendar.sso.amnamingurl* parameter, you must use a fully qualified host name for where Access Manager software is installed.

3 **To configure SSO for Messaging Server, refer to the** *Sun Java System Messaging Server 6.3 Administration Guide***.**

4 **Users log into Access Manager 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 Java Enterprise System servers.)**

5 **After logging in, users can access Calendar Server through Communications Express using the appropriate URL. Users can also access other Communications Suite servers such as Messaging Server, if the servers are configured properly for SSO.**

Parameter	Description
local.calendar.sso.amnamingurl	Specifies the URL of the Access Manager SSO naming service. Default is `http://`*AccessManager*`:`*port*`/amserver/namingservice` where *AccessManager* is the *fully qualified name* of Access Manager, and *port* is the Access Manager port number.
local.calendar.sso.amcookiename	Specifies the name of the Access Manager SSO cookie. Default is *"iPlanetDirectoryPro"*.
local.calendar.sso.amloglevel	Specifies the log level for Access Manager SSO. Range is from 1 (quiet) to 5 (verbose). Default is “3“.
local.calendar.sso.logname	Specifies the name of the Access Manager 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 Access Manager. If enabled, a user who logs out of Calendar Server is also logged out of Access Manager, and any other sessions the user had initiated through Access Manager (such as a Messaging Server Webmail session) are terminated. Because Access Manager is the authentication gateway, single sign-off is always enabled from Access Manager to Calendar Server. Default is “yes“.

Tip – A best practice for changing the `ics.conf` file is to add the parameter and its new value to the end of the file. The system reads the entire file and uses the last value found for the parameter.

8.1.1 Considerations for Using Single Sign-on With Access Manager

This section lists some considerations for using Single Sign-on (SSO) with Access Manager.

The following are some of the considerations:

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

8.1.2 Configuring SSO Through Communications Servers Trusted Circle Technology

When configuring SSO through Communications Servers trusted circle technology (that is, not through Access Manager), 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 URL's. For example, SSO works for:`http://servername`.

The following table describes the Calendar Server configuration parameters for SSO through Communications Servers trusted circle technology.

TABLE 8–1 Calendar Server SSO Parameters Through Communications Servers Trusted Circle Technology

Parameter	Description
sso.enable	This parameter must be set to "1" (the default) to enable SSO. "0" disables SSO.
sso.appid	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	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	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	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	This parameter sets the domain used as part of the user's SSO authentication.

sso.*appid*.url = "*verifyurl*"	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. This parameter contains the following parts:
	■ 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 question mark (`?`).
	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`.
	For example:
	`sso.ics50.url= "http://sesta.com:8883 /VerifySSO?"`
	`sso.msg50.url= "http://sesta.com:8882 /VerifySSO?"`

The following table describes the Messaging Server configuration parameters for SSO through Communications Servers trusted circle technology.

TABLE 8–2 Messaging Server SSO Parameters Through Communications Servers Trusted Circle Technology

Parameter	Description
local.webmail.sso.enable	This parameter must be set to a non-zero value to enable SSO.
local.webmail.sso.prefix	This parameter specifies a prefix used when formatting SSO cookies set by the HTTP server. For example: `ssogrp1`
local.webmail.sso.id	This parameter specifies the unique application ID (for example: `msg50`) for the Messaging Server. Each trusted application must also have a unique application ID.
local.webmail.sso.cookiedomain	This parameter specifies the cookie domain value of all SSO cookies set by the HTTP server.
local.webmail.sso.singlesignoff	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.

`local.sso.`*appid*`.url=`*verifyurl*	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 parts:
	■ 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 `?`).
	For example:
	`local.sso.ics50.verifyurl=`
	`http://sesta.com:8883/VerifySSO?`
	In this example, the Calendar Server application ID is `ics50,` the host URL is `sesta.com`, and the port is `8883`.
	`local.sso.msg50.verifyurl=`
	`http://sesta.com:8882/VerifySSO?`
	In this example, the Messaging Server application ID is `msg50,` the host URL is `sesta.com`, and the port is `8882`.

For more information about configuring Messaging Server for SSO, see the *Sun Java System Messaging Server 6.3 Administration Guide*.

◆ ◆ ◆ CHAPTER 9

Configuring Automatic Backups (csstored)

At configuration time you are given the opportunity to enable automatic backups. However, you can also enable or disable automatic backups at any time thereafter. A good backup system is essential to safeguard your data and minimize operational downtime.

Information in this chapter explains how to configure the Calendar Server service `csstored` to perform automatic backups.

Note – If you choose not to use the automatic backup process discussed here, you must implement your own backup strategy to safeguard your data. For information on how to use other Calendar Server tools for protecting your data, see Chapter 17, "Backing Up and Restoring Calendar Server Data."

For an overview of `csstored`, see the *Sun Java Communications Suite 5 Deployment Planning Guide*.

9.1 Enabling the Calendar Server Store Service (csstored)

The store service must be enabled in the `ics.conf` file. Verify that the store service is enabled by setting the following `ics. conf` file parameter is set to `"yes"`:

local.store.enable	If this parameter is set to "`yes`", each service that accesses the store depends on the successful start up of the store service.

9.2 Overview of Automatic Backups in a Calendar Server 6.3 System

This section is an overview of how automatic backups are implemented in a Calendar Server system.

This section covers the following topics:

9.2.1 How Automatic Backups Work in a Calendar Server 6.3 System

The Calendar Server system records each transaction for the calendar database (additions, modifications or deletions to calendars and their properties) in a transaction log file. At some predetermined interval, the log file is closed for writing and another is created. The system then applies the transactions from the oldest closed transaction log to the live calendar databases as time permits. When all the transactions in the log have been applied to the database, the log is marked as "already applied".

When hot backups are configured, a snapshot of the live databases is taken every 24 hours. The already applied logs are then applied to the hot backup copy of the databases. The hot backup databases are as current as the number of transactions still waiting to be applied.

9.2.2 How csstored Works for Backups in a Calendar Server 6.3 System

One of the Calendar Server services launched at startup is `csstored`. When configured, this service performs automatic backups (either hot backups or archival backups, or both) of your calendar databases.

You can configure `csstored` for automatic backups when you run the configuration program, `csconfigurator.sh`. If you choose one or both of the automatic backups at that time, no further configuration steps are necessary.

If `csstored` is disabled, none of the other daemons that access the database will work. The `csstored` daemon performs other necessary tasks for the databases. Therefore, the daemon should not be disabled.

Note – When automatic backups are disabled, the circular logging `ics.conf` parameter, *caldb.berkeley.circularlogging*, should be set to `"yes"`. This enables purging of old database transaction logs, which conserves disk space.

9.2.3 How Circular Backups Work in a Calendar Server 6.3 System

With automatic backups enabled, `csstored` automatically manages the number of backup copies retained in your backup database files using a circular backup system.

The `csstored` process stores backups in your backup database directory until either the maximum number of backup copies have accumulated, or the maximum disk space allowed has been reached. At that point, it purges backup copies (oldest first) until it reaches the minimum number of copies to retain and it is under the disk space threshold.

There are a cluster of `ics.conf` parameters that control circular backups. These parameters have default values, and do not require further customization. If you wish to tune how backups work in your system, see “21.7 Tuning Automatic Backups” on page 354.

9.2.4 High Level Steps for Enabling Automatic Backups

If you did not configure automatic backups when you ran the configuration file, you can set them up later. This section contains a list of high-level steps necessary for enabling automatic backups for the Calendar Server 6.3 system after the configuration program has already run.

The following is the list of high-level tasks:

- “9.3 Setting up Transaction Log Files for Calendar Server 6.3 Backups” on page 226
- “9.4 Specifying the Calendar Server Administrator’s Email Address” on page 227
- “9.5 Enabling Hot Backups for Calendar Server 6.3 Databases” on page 228
- “9.6 Enabling Archive Backups for Calendar Server 6.3 Databases” on page 229

9.3 Setting up Transaction Log Files for Calendar Server 6.3 Backups

This section contains both overview and instructions for setting up transaction log files.

This section covers the following topics:

- "9.3.1 Understanding Transaction Log Files for Calendar Server 6.3 Backups" on page 226
- "To Set up Transaction Log Files" on page 226

9.3.1 Understanding Transaction Log Files for Calendar Server 6.3 Backups

Transaction log files are used by Calendar Server to capture all of the additions, modifications and deletions made to the calendar database since the latest snapshot. The transactions are not actually applied to the live database until the log file is closed for writing. The interval parameter specifies how often the old log files are closed and new log files are created.

The log file names consist of a configurable name with a unique number appended to the end.

As the log files are closed, they are ready to be applied to the live database. This happens asynchronously, meaning the creation of log files and the writing of transactions into them is done in "real time", whereas the program applying the transactions to the database is running independently, without regard to the writing of the transactions into the log files. If the system is very busy, the number of log files awaiting application to the database can rise. When the system has slow periods, the program applying the transactions has time to "catch up" and may actually sit idle, waiting for the next transaction log.

After the transactions have been applied to the live database, they are applied to the hot backup snapshot (if enabled). The log files are also written to the same archive directory where the snapshot resides.

▼ To Set up Transaction Log Files

1. **At a command line, change to the directory where the `ics.conf` is located:**

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

2. **Specify the transaction log name:**

   `logfile.store.logname=storename.`*log*

3. **Specify the directory path for the transaction log directory:**

   The default value is: `logfile.logdir="logs"`

4 **When you have completed editing the `ics.conf` file, restart Calendar Server:**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

9.4 Specifying the Calendar Server Administrator’s Email Address

This section contains overview and instructions for setting the Calendar Server administrator's email address.

This section covers the following topics:

- “9.4.1 Email Messages Sent to the Administrator” on page 227
- “To Set the Calendar Server 6.3 System Administrator’s Email Address” on page 227

9.4.1 Email Messages Sent to the Administrator

When certain events or errors occur, the administrator is notified by email.

The events causing an email message to be generated are:

- Automatic backups not enabled or not configured properly.

  Every 24 hours, when its time to take a snapshot, if automatic backups are not enabled, the `csstored` process reports that automatic backups are not properly configured.
- The disk space threshold has been exceeded.

  This message is sent out periodically until the condition has cleared.
- A service has stopped and can’t be restarted.

  The notification email will explain what required action is needed before the service can be started.

▼ To Set the Calendar Server 6.3 System Administrator’s Email Address

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4 **Save your old `ics.conf` file by copying and renaming it.**

5 **Edit the `ics.conf` parameter that follows to specify the administrator's email address:**

```
alarm.msgalarmnoticercpt="admin@email_address"
```

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

9.5 Enabling Hot Backups for Calendar Server 6.3 Databases

This section contains an overview and instructions for enabling hot backups for Calendar Server 6.3 databases, if you did not configure them when you ran the configuration program.

This section covers the following topics:

- "9.5.1 What are Hot Backups in Calendar Server Version 6.3?" on page 228
- "To Enable Hot Backups for a Calendar Server 6.3 System" on page 228

9.5.1 What are Hot Backups in Calendar Server Version 6.3?

Ideally, hot backups consist of the latest snapshot with all of the transaction logs applied to the it, except the transaction log currently being written. The system can get behind in applying the transaction logs, depending on how busy the system is. It is possible that there might be several log files that have not yet been applied to either the database or the hot backup.

This "almost duplicate" of the live database is designed to minimize down time and data loss if something catastrophic happens, or if a corruption of the database is detected.

A new hot backup is started every 24 hours when a new snapshot is taken. The old hot backup is verified and kept until purged. For more information, see "9.2.3 How Circular Backups Work in a Calendar Server 6.3 System" on page 225.

▼ To Enable Hot Backups for a Calendar Server 6.3 System

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **At a command line, change to the directory where the `ics.conf` is located:**

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

4 **Enable hot backups by setting the following `ics.conf` parameter to `"yes"`:**

```
caldb.berkeleydb.hotbackup.enable="yes"
```

5 **Specify the directory path for the hot backup directory:**

```
caldb.berkeleydb.hotbackup.path=
   /var/opt/SUNWics5/hotbackup_directory
```

The default hot backup directory for Calendar Server is `/var/opt/SUNWics5/csdb` for Solaris and `/var/opt/sun/calendar/csdb` for Linux. The Communications Suite installer puts the archive and hot backup directories in the `csdb` directory by default because it is the convenient subdirectory that is known to the installer.

Note – It is highly recommended that the Calendar Server administrator needs to put the archive and hot backup in a different disk or volume or partition than the `csdb` directory because of the size issues.

The number of archive and hot backup directories is configurable. So, if you choose to have six of each archive and hot backup directories, it means that they might have 6 + 6 + 1 copies of your live database in the `csdb` directory. The `csstored` utility calculates the size of the required archive and hot backup based off of the size of the contents of the `csdb` directory and the physical disk that `csdb` is located on.

The archive and hot backup directories are installed inside the `csdb` directory by default for convenience. But, it should be located in a directory outside of `csdb` for a real life deployment.

You might choose to place hot backups on an alternate disk or disk subsystem in case of a hardware failure on the primary disk drive. Doing this might also reduce input-output contention on the primary drive or subsystem.

If you have a high availability (HA) configuration, specify the path as a subdirectory in shared storage (`/`*global*`/cal/`). See also, Chapter 6, “Configuring Calendar Server 6.3 Software for High Availability (Failover Service).”

6 **When you have completed editing the `ics.conf` file, restart Calendar Server:**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

9.6 Enabling Archive Backups for Calendar Server 6.3 Databases

This section contains overview material and instructions for enabling archival backups for Calendar Server databases if you did not configure them when the configuration program ran.

This section covers the following topics:

- "9.6.1 What are Archive Backups in Calendar Server Version 6.3?" on page 230
- "To Enable Archive Backups for a Calendar Server 6.3 System" on page 230

9.6.1 What are Archive Backups in Calendar Server Version 6.3?

Archive backups consist of a snapshot and the log files that were created for it. The log files are not applied to the snapshot. The archive databases remain on disk until purged. See "9.2.3 How Circular Backups Work in a Calendar Server 6.3 System" on page 225.

▼ To Enable Archive Backups for a Calendar Server 6.3 System

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **At a command line, change to the directory where the `ics.conf` is located:**

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

4 **Enable archive backups by setting the following `ics.conf` parameter to "`yes`":**

```
caldb.berkeleydb.archive.enable="yes"
```

5 **Specify the directory path for the archive directory:**

```
caldb.berkeleydb.archive.path=
   /var/opt/SUNWics5/archive_backup_directory
```

You might choose to place the archive backups on an alternate disk or disk subsystem in case of a hardware failure on the primary disk drive. Doing this might also reduce I/O contention on the primary drive or subsystem.

If you have a high availability (HA) configuration, specify the path as a subdirectory in shared storage (/*global*/`cal/`). See also, Chapter 6, "Configuring Calendar Server 6.3 Software for High Availability (Failover Service)."

6 **When you have completed editing the `ics.conf` file, restart Calendar Server:**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

The calendar services do not need to be stopped to edit the `ics.conf` file, but you must restart the services in order for the changes to take effect.

CHAPTER 10

Setting Up a Multiple Domain Calendar Server 6.3 Environment

This chapter contains overview material and instructions on how to set up a multiple domain environment for the first time.

Tip – In the past, domains in a multiple domain environment have been referred to as both *hosted domains*, and *virtual domains*. These terms are interchangeable in this document.

This chapter covers the following topics:

10.1 Overview of Multiple Domains in Calendar Server Version 6.3

Calendar Server 6.3 features multiple domains as the default, and only, way to organize your user and group LDAP entries. That is, you must have at least one domain under the root node and all of your user and group entries must reside under a domain node. In earlier versions of Calendar Server, the use of domains to contain user and group entries was optional. You could run Calendar Server without using domains at all. That is no longer true for Calendar Server 6.3; you must have at least one (default) domain.

In a multiple domain environment, each domain shares the same instance of the Calendar Server system, 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. All user and calendar IDs for domains must be fully qualified.

The configuration program asks you for the necessary information to set up the default domain. When the configuration program is done and you have created all of the domains you need, before copying your user and group LDAP entries to the desired domains, you must prepare the user and group entries by running the migration utilities to convert your non-domain user and group LDAP entries to domain ready user and group entries. The utilities to run are `csmig` and `csvdmig`.

If you are upgrading to Calendar Server 6.3 from a non-domain version of Calendar Server, you have several choices to make:

- You can choose to move to a single default domain mode.

  In which case, you must move your user and group records under the domain node in LDAP.
- You can choose to move to a multiple domain mode and distribute the users and groups amongst multiple domains.

  Use Delegated Administrator to create more domains.

  If you wish to distribute your existing users into multiple domains, you need to run the migration utilities to add fully qualified domain names to your database user IDs and calendar IDs. This way you can distribute your users among the domains you created with Delegated Administrator. Create the domains before you run the configuration program.

If you had hosted (multiple) domains set up prior to upgrading to the current version, your user IDs and calendar IDs need not be altered. However, there are some new `ics.conf` parameters that need to be configured, as shown in the following section:"10.4 Additional Parameters Required for Multiple Domain Mode in Calendar Server Version 6.3" on page 239.

Caution – If your site is currently configured for multiple instances of Calendar Server on a single machine, or if you have implemented a limited virtual domain mode, contact your Sun Microsystems sales account representative for an evaluation of your migration requirements.

10.2 Setting up a Multiple Domain Environment for Calendar Server Version 6.3 for the First Time

To move from a non- or single domain environment to a multiple domain environment, you might need to perform the following tasks before creating any LDAP entries:

1. Run the database migration utilities.

   If you are migrating from Calendar Server version 5, be sure that you have already run `csmig`, `csvdmig`, `cs5migrate`, and `cs5migrate` before attempting to set up multiple domains. For more information on these migrations utilities, see Chapter 3, “Database Migration Utilities for Calendar Server 6.3.”
2. Run the `comm_dsseetup.pl` if you have not already done so.
3. Log in as an administrator with permission to change the configuration.
4. Stop Calendar Server services by issuing the `stop-cal` command.
5. Edit the `ics.conf` file to enable multiple domains.

   The following table lists and describes the configuration parameters in the `ics.conf` file used for multiple domain support. If any of the parameters listed in this table 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.

Parameter	Description
service.virtualdomain.support	Enables ("`yes`") support for multiple domains. Default is "`yes`". **Note** – Do not change this parameter to “`no`” even if you are going to operate in a single domain. The current version of Calendar Server requires this to be “`yes`”.
local.schemaversion	Specifies the version of the LDAP schema: ■ "`1`" for “10.3.3 Sun LDAP Schema Version 1 for Calendar Server Version 6.3” on page 238. See also `service.dcroot`. ■ "`1.5`" or "`2`" for “10.3.2 Sun LDAP Schema Version 2 for Calendar Server Version 6.3” on page 237. See also *service.schema2root*. Default is "`2`".

Parameter	Description
service.dcroot	For multiple domain support, this replaces *local.ugldapbasedn* and `local.authldapbasedn`. If `local.schemaversion="1"` or `local.schemaversion="1.5"`, specifies the root suffix of the DC tree, underneath which all domains are found. For example: "`o=internet`".
service.schema2root	if `local.schemaversion="2"`, specifies the root suffix of the Organization Tree, underneath which all domains are found. 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: "`red.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`.
service.virtualdomain.scope	Controls cross domain searching: ■ "`primary`" Search only within the domain where the user 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).

6. Create a default domain entry.

   For Schema version 2, the default domain is created by the Delegated Administrator configuration program (`config-commda`).

   For Schema version 1, create a default domain (one of your multiple domains), placing it one or more levels under your DC tree root suffix, depending on your DC tree structure. For example, if your root suffix is `o=internet`, then the next level down node could be `com`, as shown in "10.3.3 Sun LDAP Schema Version 1 for Calendar Server Version 6.3" on page 238. However, your default domain would be one node lower, such as `sesta.com`. Use *csdomain* to create DC tree nodes, as illustrated in the example that follows:

```
csdomain -n o=com,dc=com,o=internet create comcsdomain
    -n o=sesta.com,dc=sesta,dc=com,o=internet create sesta.com
```

7. Enable calendaring services for the default domain entry.

   For Schema version 1: Add the `icsCalendarDomain` object class to the `o=sesta.com` domain entry in LDAP using *csattribute*.

   For Schema version 2: After configuring Delegated Administrator, modify the default domain (created by the Delegated Administrator configuration program) to add Calendar (and Mail) services. In the following example, calendar and mail services are added to a domain:

   ```
   commadmin domain modify -D admin -w passwd -d defaultdomain -S cal,mail
   ```

8. Create as many domains as you want on your system.

   For instructions on how to add a domain in Schema version 2 mode, see "13.2 Creating New Calendar Server Domains" on page 264.

   To create a Schema version 1 domain, use `csdomain create`, as illustrated in the example that follows:

   ```
   csdomain -n o=red.sesta.com,dc=red,dc=sesta,dc=com
      create red.sesta.com
   ```

9. Add calendar (and mail, if wanted) services for the new domains.

10. Create the `calmaster` site administrator user if it does not already exist.

    For Schema version 2, create the `calmaster` user using the `commadmin user create` command as illustrated in the following example:

    ```
    commadmin user create -D admin -w passwd -F Calendar
        -L Administrator -l calmaster -W calmasterpasswd -d sesta.com -S cal
    ```

    Note – To create the `calmaster` using the Delegated Administrator Console's `Create New User` wizard, see the Delegated Administrator online help.

    For Schema version 1, create the `calmaster` user on the organization tree with `csuser` as illustrated in the following example:

    ```
    csuser o=sesta.com,o=rootsuffix -d sesta.com
        -g Calendar -s Administrator -ycalmasterpasswordcreate calmaster
    ```

11. If the `calmaster` site administrator user already exists from an earlier non-domain environment (Schema version 1), move it to the default domain by performing the following steps:
    a. Perform an LDAP dump of the existing `calmaster` LDAP entry and save it in a temporary file, such as `/tmp/calmaster.ldif`.
    b. Delete the existing `calmaster` LDAP entry on the organization tree root suffix using `ldapdelete`, as follows:

    ```
    #ldapdelete -D "cn=Directory Manager" -w password
       uid=calmaster,ou=People,o=rootsuffix
    ```

    c. Modify the calendar administrator's group entry (update the `uniqueMember` attribute) to reflect the changes as shown in the LDIF example that follows:

    ```
    dn:cn=Calendar Administrators,ou=Groups,o=rootsuffix
    changetype:modifyreplace:uniqueMember
    uniqueMember:uid=calmaster,ou=People,o=sesta.com,o=rootsuffix
    ```

    It is not necessary to move the administrator's group entry to the domain.
12. Update any administration scripts you have so that all calendar IDs (`calid`) in the WCAP commands are fully qualified. That is, each `calid` must now include the domain name. For example: `jsmith@sesta.com`.

10.3 How the Multiple Domains Feature in Calendar Server Version 6.3 Influences Your Schema Choices

This section contains conceptual information you can use to better understand the process for implementing multiple domains and what it has to do with choosing a schema version.

This section contains the following topics:

10.3.1 Overview of Multiple Domains and the Implications for Schema Choice for Calendar Server Version 6.3

With a multiple 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). User, group and resource unique IDs are unique within each domain. For example, there can be only one user in each domain with the `uid` of `jdoe`. A distinguished name (DN) is a fully qualified domain name.

Calendar Server supports both of these LDAP directory schema versions: Schema version 1 and Schema version 2. When you run the Directory Server Setup script (`comm_dssetup.pl`), you can choose either LDAP Schema version 1 or LDAP Schema version 2. Use Schema version 2, unless you have specific reasons for using Schema version 1

The following are two reasons to use Schema version 1:

- You already have Schema version 1 and you are not planning on using Delegated Administrator for populating LDAP.
- You already have Schema version 1 and you are not planning to use Access Manager features.

10.3.2 Sun LDAP Schema Version 2 for Calendar Server Version 6.3

The following graphic shows an LDAP directory organization for a multiple domain installation that uses Sun LDAP Schema version 2.

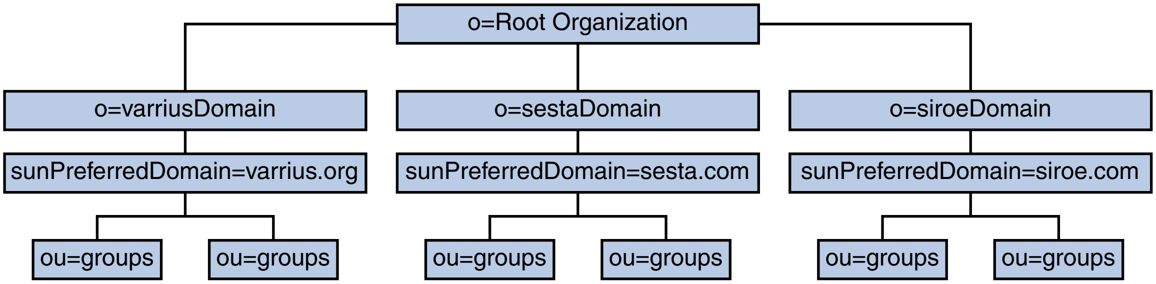


FIGURE 10–1 LDAP Directory Organization Using LDAP Schema Version 2

LDAP Schema version 2 uses a flat LDAP directory organization, that is, the domains are all at the same level; they are not nested. For a multiple domain installation, the first level entries (as shown by `varriusDomain`, `sestaDomain`, and `siroeDomain` in the graphic) must be parallel in the directory organization. These entries cannot be nested.

If you want to use Access Manager features such as single sign-on (SSO), or use Delegated Administrator to provision users, Schema version 2 is required. However, there is a hybrid variation, a two tree scheme that uses both the DC tree and the Organization tree, much like

Schema version 1, but it uses the Schema version 2 object classes and attributes. This is Schema version 2 compatibility mode, which is called Schema version 1.5 in the configuration program (`csconfigurator.sh`).

10.3.3 Sun LDAP Schema Version 1 for Calendar Server Version 6.3

The graphic that follows shows an example of an LDAP directory organization for a multiple domain installation that uses Sun LDAP Schema version 1.

This organization includes two trees for domain management:

- DC tree
- Organization (OSI) tree

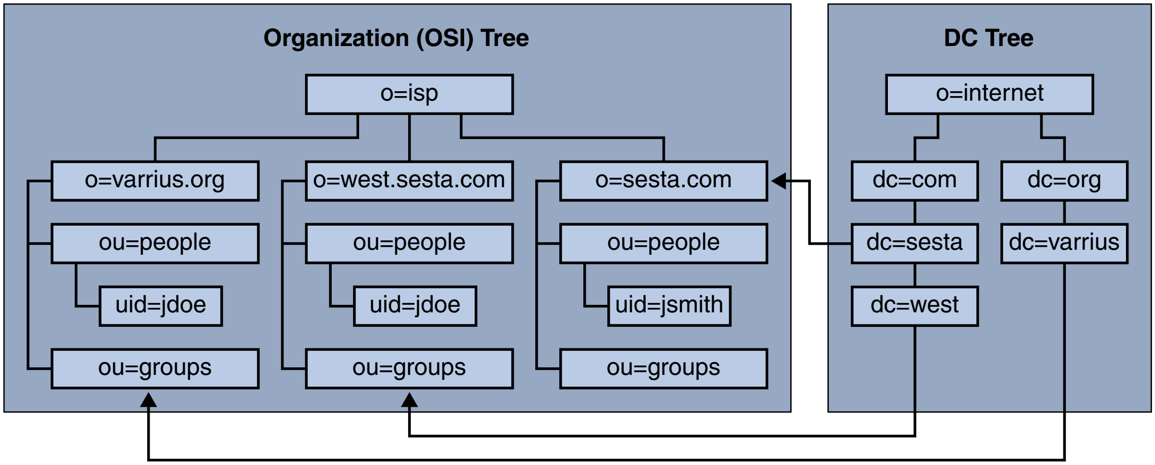


FIGURE 10–2 LDAP Directory Organization Using LDAP Schema Version 1

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 organization tree (node). Within each domain, the identifiers for Calendar Server users, resources, and groups must be unique.

Note – If your earlier LDAP configuration did not contain a DC tree, in order to use Schema version 1 mode or Schema version 2 compatibility mode, you must create the DC tree nodes yourself as explained in "10.2 Setting up a Multiple Domain Environment for Calendar Server Version 6.3 for the First Time" on page 233. However, Schema version 2 is the preferred mode.

In a multiple domain installation using LDAP Schema version 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 organization tree.
2. In the organization 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.

10.4 Additional Parameters Required for Multiple Domain Mode in Calendar Server Version 6.3

Starting with Calendar Server 6 , every deployment is configured for multiple domains. If you are upgrading from an older version of Calendar Server, and did not have hosted (multiple) domains configured, you must add the parameters, as shown, for your schema mode:

- "10.4.1 Schema Version 1 Parameter Additions for Calendar Server Version 6.3" on page 239
- "10.4.2 Schema Version 2 Parameter Additions for Calendar Server Version 6.3" on page 240

10.4.1 Schema Version 1 Parameter Additions for Calendar Server Version 6.3

The following list of parameters should be added to the configuration file (`ics.conf`), if they do not already exist.

service.dcroot
service.defaultdomain
service.loginseparator
service.virtualdomain.support set to `"yes"`
service.virtualdomain.scope
service.siteadmin.userid
service.siteadmin.cred
local.schemaversion set to `"1"`

10.4.2 Schema Version 2 Parameter Additions for Calendar Server Version 6.3

The following list of parameters should be added to the configuration file (`ics.conf`), if they do not already exist:

service.dcroot
service.defaultdomain
service.loginseparator
service.virtualdomain.support set to "`yes`"
service.virtualdomain.scope
service.siteadmin.userid
service.siteadmin.cred
local.schemaversion set to "`2`"
service.schema2root

10.5 Calendar Server 6.3 Logins

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

userid`[@`*domain-name*`]`

If *domain-name* is omitted, Calendar Server uses the default 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.

If autoprovisioning is enabled, the first time a user logs in, Calendar Server creates a default calendar for the user. For information about calendar creation, see Chapter 15, "Administering Calendars."

Login permission is based on the `icsStatus` or `icsAllowedServiceAccess` attribute. For more information, see "D.9.3 LDAP Attributes and Property Names" on page 422.

10.6 Migrating from a Non-Domain Environment in Calendar Server Version 6.3

In Calendar Server version 5.0 and earlier, there were no domains. Therefore, the user and calendar ID's were not required to be fully qualified. That is, they did not need a domain name to be part of the ID such as `jdoe@domain.com`. If your `uid`'s and `calid`'s were not fully qualified before installing the current version of Calendar Server, your data does not have to be altered.

The system assumes any unqualified uid's and calid's it encounters to belong to the default domain. If you want to implement multiple domains, however, you must migrate your LDAP and components databases to indicate which domain each user belongs to.

In addition, your data might need to be migrated in other ways. There are several migration programs. Check the migration information found in Chapter 3, “Database Migration Utilities for Calendar Server 6.3.”

CHAPTER 11

Customizing Existing Domains for a Calendar Server 6.3 System

This chapter contains conceptual information and instructions on how to customize your existing domains.

This chapter describes these topics:

11.1 Configuring Domain Preferences for Groups in Calendar Server Version 6.3

If you have user groups set up in LDAP, you can specify domain level preferences for doublebooking and set default ACL's.

11.1.1 Setting the Doublebooking Domain Preference in Calendar Server Version 6.3

Set bit 15 of the `icsAllowRights` attribute in the domain LDAP entry. Use `"0"` if doublebooking is not allowed. Use `"1"` if doublebooking is allowed.

11.1.2 Specifying the Default ACL for Groups in Calendar Server Version 6.3

You can change the default access control permissions for groups at various levels.

This section covers the following group ACL topics:

- "11.1.2.1 For All Groups" on page 244
- "11.1.2.2 For All Groups in a Specific Domain" on page 244
- "11.1.2.3 For a Specific Group in a Domain" on page 244

11.1.2.1 For All Groups

The default ACL for groups in any domain is specified in the `ics.conf` file parameter *group.default.acl*. The default ACL is:

```
"@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g"
```

You can change the ACL by editing it.

11.1.2.2 For All Groups in a Specific Domain

To change the default ACL for groups in a specific domain, you must edit the domain LDAP entry. Change the value of the `groupdefaultacl` property in the *icsExtendedDomainPrefs* attribute.

11.1.2.3 For a Specific Group in a Domain

To change the default ACL for a specific group, edit the group LDAP entry. Change the value of the *icsDefaultacl* attribute.

11.2 Cross Domain Searching in Calendar Server 6.3 Systems

This section contains conceptual information and high-level tasks for setting up cross domain searches.

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

The following is a list of requirements you must meet to successfully implement cross domain searches:

- 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.

  Tip – To specify more than one domain, supply a semicolon separated list of domain names for the value of the `domainAccess` property.

  Caution – There can be only one instance of the `domainAccess` property in an LDAP domain entry. If you use LDAP tools to add ACLs to a domain entry, you must ensure that you are not inadvertently creating a duplicate of the `domainAccess` property.

  For a description of `domainAccess`, see "D.9.3 LDAP Attributes and Property Names" on page 422. For general information about ACLs, see "1.8.3 Access Control Lists (ACLs) in Calendar Server Version 6.3" on page 52.
- 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 "D.9.3 LDAP Attributes and Property Names" on page 422.

For instructions on how to enable cross domain searches, see "13.3 Enabling Cross Domain Searches" on page 265.

11.3 Using Domains Created by Messaging Server in Calendar Server Version 6.3

If Messaging Server has already created domains, you can add calendar services in either Schema version 1 or Schema version 2 mode.

This section covers the following topics:

- "11.3.1 Adding Calendar Services to Messaging Domains in Schema Version 1 Mode for Calendar Server Version 6.3" on page 246
- "11.3.2 Adding Calendar Services to Messaging Domains in Schema 2 Mode in Calendar Server Version 6.3" on page 246

11.3.1 Adding Calendar Services to Messaging Domains in Schema Version 1 Mode for Calendar Server Version 6.3

To add calendar services to a domain, add the following object class and two attributes to the domain's LDAP entry:

- Object class: `icsCalendarDomain`.
- Attribute:*icsStatus*. Set the value to "`active`".
- Attribute: *icsExtendedDomainPrefs*. Set the value of the *domainAccess* attribute option to the ACL you want to use for access control.

You can do this in one of two ways: use `csattribute add` command or use `ldapmodify` as shown in the example that follows:

```
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
```

11.3.2 Adding Calendar Services to Messaging Domains in Schema 2 Mode in Calendar Server Version 6.3

If Messaging Server is in Schema version 2 mode, perform the following two steps to add calendar services to the existing domains:

1. Use the Delegated Administrator Utility command `commadmin domain modify` with the `-S` option to add calendar service to each domain.

   Alternately, you can use the Delegated Administrator Console to allocate service packages containing calendar service to the affected domains. To do this, use the Allocate Service Packages button on the Organization List page.

2. Use the Delegated Administrator Utility command `commadmin user modify` with the `-S` option to add calendar service to each user in each domain you enabled for calendar.

   Alternately, you can use the Delegated Administrator Console to assign a service package containing calendar service to each user in the affected domains. To do this, use the Assign Service Package button on the User List page in each affected organization.

For the `commadmin` commands, see the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

For more information about Delegated Administrator Console, see its online help.

For `commdirmig` information, see the *Sun Java Communications Suite 5 Schema Migration Guide*.

PART IV

Calendar Server 6.3 Administration

This section contains chapters dealing with administration of your Calendar Server deployment.

This part contains the following chapters:

CHAPTER 12

Server Administration for a Calendar Server 6.3 Deployment

This chapter describes server administration for a Calendar Server deployment.

This chapter contains the following sections:

You manage Calendar Server by running either the Delegated Administrator utility (formerly the User Management Utility) or the Calendar Server 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 Appendix D, "Calendar Server Command-Line Utilities Reference."

Note – Additional administration topics are covered in separate chapters.

The chapters cover the following topics:

- Chapter 13, "Administering Calendar Server Domains"
- Chapter 14, "Administering Users, Groups, and Resources."
- Chapter 15, "Administering Calendars"
- Chapter 16, "Administering Calendar Server Databases with the csdb Utility"
- Chapter 17, "Backing Up and Restoring Calendar Server Data"
- Chapter 18, "Administering the Delete Log Database"

- Chapter 19, "Administering Calendar Server Time Zones"
- Chapter 20, "Using Instant Messaging Pop-up Reminders"
- Chapter 21, "Tuning Calendar Server Performance"
- Chapter 22, "Troubleshooting Calendar Server 6.3 Software"

12.1 Starting and Stopping Calendar Server 6.3 Processes

This section contains conceptual information and instructions on how to use the `start-cal` and `stop-cal` commands.

This section contains the following topics:

- "12.1.1 About Calendar Server 6.3 Commands: start-cal and stop-cal" on page 252
- "To Start Calendar Server 6.3 Services with start-cal" on page 253
- "To Stop Calendar Server with stop-cal" on page 253

12.1.1 About Calendar Server 6.3 Commands: start-cal and stop-cal

You can start and stop Calendar Server using the `start-cal` and `stop-cal` commands.The `start-cal` and `stop-cal` utilities are located in the *cal-svr-base*`/SUNWics5/cal/sbin` directory. You must run these utilities on the local machine where Calendar Server is installed.

Note – Check your scripts to make sure you do not use the old `csstart` and `csstop` utilities. Use the `start-cal` and `stop-cal` utilities to start and stop Calendar Server.

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

1. `watcher` — Watcher, the process that monitors the system
2. `enpd`— Event Notification Service (ENS)
3. `csstored`— Automatic Backup Service
4. `csnotifyd`— Notification Service
5. `csadmind`— Administration Service
6. `csdwpd`— Database Wire Protocol (DWP) service, the distributed database service starts only if you have a remote Calendar Server database configuration
7. `cshttpd`— HTTP Service

For a description of these services, see"1.10 Services Running As Daemons in Calendar Server Version 6.3" on page 58

▼ To Start Calendar Server 6.3 Services with start-cal

1. **Log in as a user who has administrative rights to the system.**
2. **Verify that all Calendar Server services are stopped by issuing the `stop-cal` command.**
3. **Change to the directory.**

   cal-svr-base`/SUNWics5/cal/sbin`
4. **Start Calendar Server.**

   ```
   ./start-cal
   ```

▼ To Stop Calendar Server with stop-cal

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

   cal-svr-base`/SUNWics5/cal/sbin`
3. **Stop Calendar Server.**

   ```
   ./stop-cal
   ```

12.2 Enabling or Disabling Automatic Backups in Calendar Server Version 6.3

Automatic backups are managed by the `csstored` process that starts up automatically when `start-cal` is issued. However, you can either enable or disable automatic backups at will. The default is for automatic backups to be disabled. The `csstored` process runs even if automatic backups are not enabled.

There are two kinds of automatic backups: hot backups and archival backups. You enable or disable them separately.

For information on automatic backups and instructions on configuring `csstored`, see Chapter 9, "Configuring Automatic Backups (csstored)."

The following is a list of tasks for enabling and disabling automatic backups:

- "To Enable Hot Backups in Calendar Server Version 6.3" on page 254
- "To Enable Archive Backups in Calendar Server Version 6.3" on page 254
- "To Disable Hot Backups in Calendar Server Version 6.3" on page 255

- "To Disable Archive Backups in Calendar Server Version 6.3" on page 255

To Enable Hot Backups in Calendar Server Version 6.3

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3 **Change to the directory where the** `ics.conf` **file is located.**

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

4 **Enable hot backups by setting the following** `ics.conf` **parameter to** "`yes`"**:**

```
caldb.berkeleydb.hotbackup.enable="yes"
```

5 **Specify the directory path for the hot backup directory:**

```
caldb.berkeleydb.hotbackup.path=
   /var/opt/SUNWics5/hotbackup_directory
```

The default is the current directory.

6 **When you have completed editing the** `ics.conf` **file, restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

The calendar services do not need to be stopped to edit the `ics.conf` file, but you must restart the services in order for the changes to take effect.

To Enable Archive Backups in Calendar Server Version 6.3

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3 **Change to the directory where the** `ics.conf` **file is located.**

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

4 **Enable archive backups by setting the following** `ics.conf` **parameter to** "`yes`"**:**

```
caldb.berkeleydb.archive.enable="yes"
```

5 **Specify the directory path for the archive directory.**

```
caldb.berkeleydb.archive.path=
   /var/opt/SUNWics5/hotbackup_directory
```

The default is the current directory.

6 **When you have completed editing the `ics.conf` file, restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

The calendar services do not need to be stopped to edit the `ics.conf` file, but you must restart the services in order for the changes to take effect.

▼ To Disable Hot Backups in Calendar Server Version 6.3

Backups are disabled by default. If you have previously enabled them and want to disable them, perform the following steps:

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **Change to the directory where the `ics.conf` file is located.**

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

4 **Disable hot backups by setting the following `ics.conf` parameter to "`no`":**

```
caldb.berkeleydb.hotbackup.enable="no"
```

5 **When you have completed editing the `ics.conf` file, restart Calendar Server.**

```
cal-svr-base/SUNWics5/cal/sbin/start-cal
```

The calendar services do not need to be stopped to edit the `ics.conf` file, but you must restart the services in order for the changes to take effect.

▼ To Disable Archive Backups in Calendar Server Version 6.3

Backups are disabled by default. If you have previously enabled them and want to disable them, perform the following steps:

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **Change to the directory where the `ics.conf` file is located.**

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

4 **Disable archive backups by setting the following `ics.conf` parameter to "`no`":**

```
caldb.berkeleydb.archive.enable="no"
```

5 **When you have completed editing the `ics.conf` file, restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

The calendar services do not need to be stopped to edit the `ics.conf` file, but you must restart the services in order for the changes to take effect.

12.3 Managing the Group Scheduling Engine Queue for Calendar Server Version 6.3

This section contains conceptual information and instructions for managing the Group Scheduling Engine (GSE)

The GSE keeps a queue of events that will be used to update the component database. An administrator can change the timeout value to adjust the time between Calendar Server scans of the queue. Events in the queue can also be listed and specific events deleted if necessary.

This section contains the following topics:

- "12.3.1 About GSE for Calendar Server Version 6.3" on page 256
- "12.3.2 About the Calendar Server 6.3 GSE Queue" on page 256
- "12.3.3 Listing Entries in the Calendar Server 6.3 GSE Queue" on page 257
- "12.3.4 Deleting Entries in the GSE Queue in Calendar Server Version 6.3" on page 257

12.3.1 About GSE for Calendar Server Version 6.3

The GSE allows a Calendar Server user to create events and invite other attendees. 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 and the GSE will update the event with the reply.

12.3.2 About the Calendar Server 6.3 GSE Queue

The GSE queue is in reality a separate database managed by the `csadmind` process Calendar Server scans the queue for updates that need to be made to the components database.

You can tune Calendar Server by adjusting the frequency of this scan. This is accomplished by changing the timeout value of *gse.belowthresholdtimeout* in the `ics.conf` file. See Chapter 21, "Tuning Calendar Server Performance."

The GSE queue entries can be managed (listed and deleted) using `csschedule`. You must run `csschedule` on the local machine where Calendar Server is installed.

12.3.3 Listing Entries in the Calendar Server 6.3 GSE Queue

To list entries in the GSE queue, use the `csschedule` utility `list` command.

For example, to list all entries in the GSE queue:

```
csschedule list
```

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

```
csschedule -c 10 list
```

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

```
csschedule -v list Holiday_Schedule
```

12.3.4 Deleting Entries in the GSE Queue in Calendar Server Version 6.3

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
```

12.4 Monitoring Calendar Server 6.3 Processes

Calendar Server and Messaging Server now use the same stop and start mechanism, as part of the Sun Java™ Enterprise System Monitoring Framework (JESMF). The `start-cal` command starts the `watcher` process first, and then starts all other processes. The `watcher`process is aware of any dependencies the other services have, and in which sequence the services should be started.

Each registered service (process) opens a connection to the Watcher. If a process dies without properly disconnecting, the Watcher automatically restarts it. If the process dies twice in a defined interval, Watcher does not restart it. This timeout interval is configurable.

Watcher writes to a single log, *cal-svr-base*`/data/log/watcher.log` , which contains the following information:

- Failure notices and non-response error messages that were sent to the administrative console.
- Records of all server stops and starts.

For information on how to configure Watcher, see“To Configure the Watcher Process for Calendar Server Version 6.3” on page 136

12.5 Clearing the CLD Cache in Calendar Server Version 6.3

This section covers conceptual information and instructions on how to clear the CLD cache.

This section covers the following topics:

- “12.5.1 Why Clear the Calendar Server 6.3 CLD Cache?” on page 258
- “To Clear the CLD Cache” on page 259

12.5.1 Why Clear the Calendar Server 6.3 CLD Cache?

If you have enabled the CLD Cache, you might need to clear the cache from time to time. The CLD cache can get out of sync (stale) with your system configuration for various reasons.

The following are a few reasons the CLD cache might become stale.

- You add, delete or rename a server.
- You move a server from one function to another in your configuration.
- You move one or more calendars to different back-end servers.

If any of these things happen, in order to refresh the CLD cache, you must clear it.

▼ To Clear the CLD Cache

1. **Stop Calendar Server.**

2. **Remove all files in the `/var/opt/SUNWics5/csdb/cld_cache` directory, but do not remove the `cld_cache` directory itself.**

3. **Restart Calendar Server.**

12.6 Changing a Server Name

If you add, delete or change a server name in your configuration, there are several "housekeeping" steps you should perform to avoid errors.

The following steps are useful to keep your CLD uptodate:

- Clear the CLD Cache
- If an old server is taken out, delete it from the `ics.conf` parameters where it appears.

12.7 Configuring Anonymous Access for Calendar Server Users

This sections contains instructions on how to enable and disable anonymous access (login).

Anonymous access is a special login that does not require authentication. With anonymous login enabled, read and write access to public calendars is enabled by default. It is possible to deny write access to the public calendars.

This section covers the following topics:

- "To Enable Anonymous Access" on page 259
- "To Disable Anonymous Users from Writing to Public Calendars" on page 260

Note – Communications Express expects anonymous logins to be allowed for writing as well as reading. See "4.1 Configuring for Communications Express" on page 114.

▼ To Enable Anonymous Access

1. **Log in as an administrator with configuration privileges.**

2. **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit the following parameters in the** `ics.conf` **to enable anonymous access:**

Parameters	Description and Default Value
service.http.allowanonymouslogin	Enable anonymous access (login) by setting this parameter to "`yes`", if necessary. The default value is "`yes`".
service.calendarsearch.ldap	For security purposes with anonymous logins enabled, you might want to disable searching through the LDAP first when doing calendar searches, by setting this parameter to "`no`", which is the default.

Note – Communications Express expects the value of the `service.calendarsearch.ldap` parameter to be "`no`". This conflicts with instructions given for tuning your system for best performance in a DWP environment. (Your database is distributed across multiple back-ends.) See "21.2 Improving Calendar Search Performance in a DWP Environment" on page 350.

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Disable Anonymous Users from Writing to Public Calendars

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit the following** `ics.conf` **parameter as shown in the following table:**

Parameter	Description and Default Value
service.wcap.anonymous. *allowpubliccalendarwrite*	Enables or disables allowing anonymous access users to write to public calendars. Enable access by setting the value to `"yes"`, which is the default.

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

cal-svr-base/SUNWics5/cal/sbin/start-cal

12.8 Enabling Proxy Administrator Logins

Proxy administrator logins (proxy authentication) must be enabled for Communications Express. For instructions on configuring proxy authentication for Communications Express, see "4.1 Configuring for Communications Express" on page 114.

However, proxy authentication can be enabled even if you are not using Communications Express. This section contains the procedure for enabling proxy authentication without Communications Express:

- "To Enable Proxy Authentication without Communications Express" on page 261
- "To Verify Proxy Authentication is Working" on page 262

▼ To Enable Proxy Authentication without Communications Express

1 **Log in as an administrator with permission to change the configuration.**

2 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

3 **Save your old `ics.conf` file by copying and renaming it.**

4 **Edit the `ics.conf` file, confirming that the following parameter is set as shown:**

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

If not, change it to `"yes"`.

5 **Save the file as `ics.conf`.**

6 **Restart Calendar Server for the new value to take effect.**

▼ To Verify Proxy Authentication is Working

- **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&fmt-out=text/html
```

This list defines the variables in the previous example:

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

If the command is successful, the system displays the calendar for *calendar-user*. If problems occur, the system displays `Unauthorized`.

The following is a list of some reasons the command might fail:

- 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.

12.9 Refreshing the Calendar Server Configuration

In the Calendar Server 6.3 release, use the `stop-cal` and `start-cal` commands to refresh a configuration. For more information, see "12.1 Starting and Stopping Calendar Server 6.3 Processes" on page 252.

◆ ◆ ◆ CHAPTER 13

Administering Calendar Server Domains

This chapter contains conceptual information and instructions on how to administer domains in your Calendar Server deployment.

This chapter contains the following sections about administering multiple domains:

- "13.1 Choosing the Correct User Management Tool" on page 263
- "13.2 Creating New Calendar Server Domains" on page 264
- "13.3 Enabling Cross Domain Searches" on page 265

13.1 Choosing the Correct User Management Tool

There are two ways to administer Calendar Server domains.

Use one of the two following set of tools:

- Delegated Administrator Console or Utility (for Schema version 2 environments).

  Delegated Administrator is a separately installable component in the Java Enterprise System installer. For more information on the Utility, see the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*. For more information on the Console, use the Delegated Administrator Console online help.
- Calendar Server Utilities: `csdomain` and `csattribute` (for Schema version 1 environments.)

  Installed with Calendar Server. You can add or delete attributes with `csdomain`, but there is no `modify` command. Use `csattribute` to modify the value of an existing attribute. In addition, should the need arise, use `ldapmodify` to add or delete object classes in domains created with `csdomain`.

  For information about `csdomain` and `csattribute`, see Appendix D, "Calendar Server Command-Line Utilities Reference."

For information about particular object classes and attributes, see the*Sun Java System Communications Services 6 2005Q4 Schema Reference*.

For an overview of multiple domains and other introductory material, see Chapter 10, "Setting Up a Multiple Domain Calendar Server 6.3 Environment."

Caution – Calendar Server does not support using the Access Manager Console for domain administration.

13.2 Creating New Calendar Server Domains

This section contains conceptual information and instructions for adding domains to your Calendar Server deployment. You can use either schema with multiple domains. However, if you have the choice, use Schema version 2 to take advantage of the superior tool set.

This section contains the following topics:

- "13.2.1 Overview of Creating Calendar Server Domains" on page 264
- "13.2.2 To Add a Domain in Schema Version 2 Mode" on page 265
- "13.2.3 To Add a Domain in Schema Version 1 Mode" on page 265

13.2.1 Overview of Creating Calendar Server Domains

Calendar Server software has multiple domains enabled by default. Do not change the value of the following `ics.conf`parameter: `service.virtualdomain.support="yes"`.

Once you have completed the preparation work described in Chapter 10, "Setting Up a Multiple Domain Calendar Server 6.3 Environment," you can add new domains.

Each domain has a set of attributes and preferences that you can set. 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.

13.2.2 To Add a Domain in Schema Version 2 Mode

This section contains instructions on how to add a domain in Schema version 2 mode.

You can use either the Delegated Administrator Console or Utility:

- Console — Use the Create New Organization wizard on the Organization List page.

  For more information, see the Delegated Administrator Console online help.
- Utility — Use the `commadmin domain create` command.

  For example, to create the domain `sesta.com`, issue the following command:

  ```
  commadmin domain create -D calmaster
      -d sesta.com -w calmasterpassword -S cal
      -B backend.sesta.com
  ```

  For information about the Delegated Administrator Utility, see the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

13.2.3 To Add a Domain in Schema Version 1 Mode

This section contains a simplified sample instruction for using the `csdomain` utility.

Use `csdomain create` when creating a domain in Schema version 1. For example, to create `west.sesta.com`, use the following command.

```
csdomain create west.sesta.com
```

For instructions on how to configure for multiple domains, see Chapter 10, "Setting Up a Multiple Domain Calendar Server 6.3 Environment."

13.3 Enabling Cross Domain Searches

This section contains the instructions for enabling cross domain searches.

This section covers the two tasks you must do to enable cross domain searches:

- "13.3.1 Adding Names of Domains Allowed to Search This Domain" on page 266 in the LDAP entry for each of the domains allowed to search this domain.
- "13.3.2 Adding Names of Domains to be Searched by This Domain" on page 268 when users in this domain send invitations to events.

This can be done using either of the following tools: `ldapmodify` (for either Schema mode), or Delegated Administrator Console or Utility (for Schema version 2).

13.3.1 Adding Names of Domains Allowed to Search This Domain

Each domain LDAP entry specifies access permissions in ACE's, which are defined in the *domainAccess* parameter of the *icsExtendedDomainPrefs* attribute. Two different ways to allow external domains to search this domain are:

- “13.3.1.1 To Allow Specific Domains to Search This Domain” on page 266
- “13.3.1.2 To Allow All External Domains to Search This Domain” on page 267

The construction of ACI's is explained more fully in “1.8 Access Control for Calendar Server Version 6.3” on page 51.

13.3.1.1 To Allow Specific Domains to Search This Domain

This can be done three ways:

- Using `ldapmodify`, create the following ACE string in the *domainAccess* preference of the *icsExtendedDomainPrefs*:

  `@`*domain_being_allowed*`^a^lsfr^g`

  Form the ACE by specifying the domain allowed to search this domain, followed by sufficient permissions to allow the search.

Caution – Only one instance of the `domainAccess` property is allowed. If you change the value using `ldapmodify`, you must ensure that you do not inadvertently create a duplicate of this property.

Unlike how the system reads the `ics.conf` file sequentially, and honors the value of the attribute that it finds last, for LDAP entries, the system uses the first instance it finds. Since the LDAP search mechanism does not guarantee the entry contents will be served in any specific order, an older version of the property might be retrieved first and Calendar Server software wouldn't look any farther.

- Using Delegated Administrator Utility command `commadmin domain modify`, add ACE strings specifying the *domainAccess* preference in *icsExtendedDomainPrefs* attribute.

  For example, in a Schema version 2 environment, `sesta.com` allows searches from `siroe.com`:

```
commadmin domain modify -D admin
   -w adminpassword -X hostmachine_1 -d sesta.com
   -A +icsextendeddomainprefs:"domainAccess=@@d^a^slfrwd^g;
      @siroe.com^a^lsfrwd^g;anonymous^a^r^g;@^a^s^g"
```

- Using Delegated Administrator Console, when creating or editing an organization's properties, you can add domains to the `Allow Invitations From Users in These Organizations` list.

  This updates the *domainAccess* preference in the *icsExtendedDomainPrefs* attribute.

Note – While you can specify the exact permissions given to the domains in the first two methods just listed, the last one, using the Delegated Administrator Console, does not allow the administrator as much control. The list of permissions is preset. The permissions given are: free-busy access, and event scheduling access. The user can't see event details unless the owner of that calendar has set permissions to allow all users to read it.

13.3.1.2 To Allow All External Domains to Search This Domain

There are three ways to allow all external domains to search this domain:

- Using `ldapmodify`, create the following ACE string in the *domainAccess* preference of the *icsExtendedDomainPrefs*:

  `@^a^slfr^g`

  Form the ACE by specifying that all domains have sufficient access to perform searches.

- Using Delegated Administrator Utility command `commadmin domain modify`, add ACE strings specifying the *domainAccess* preference in *icsExtendedDomainPrefs* attribute.

  For example, in a Schema version 2 environment, `sesta.com` allows searches by all domains:

```
commadmin domain modify -D admin
   -w adminpassword -X hostmachine_1 -d sesta.com
   -A +icsextendeddomainprefs:"domainAccess=@@d^a^slfrwd^g;
      anonymous^a^r^g;@^a^slfr^g"
```

  Note – The characters `@@d` refer to the domain of the primary owner.

- Using Delegated Administrator Console, when creating or editing an organization's properties, you can add domains to the `Allow Invitations From Users in These Organizations` list.

  This updates the *domainAccess* preference in the *icsExtendedDomainPrefs* attribute.

Note – While you can specify the exact permissions given to the domains in the first two methods just listed, the last one, using the Delegated Administrator Console, does not allow the administrator as much control. The list of permissions is preset. The permissions given are: free-busy access, and event scheduling access. The user can't see event details unless the owner of that calendar has set permissions to allow all users to read it.

13.3.2 Adding Names of Domains to be Searched by This Domain

This section contains instructions for adding names of domains to be searched.

There are three ways to do add external domains to be searched by this domain:

- Using `ldapmodify`, add one instance of *icsDomainNames* for each external domain that can be searched by users in this domain.

  For example, `sesta.com` searches in both `siroe.com` and `example.com` when performing cross domain searches. Use `ldapmodify` (for either Schema version 1 or Schema version 2) to create the following `LDIF`:

  ```
  dn: dc=sesta, dc=com, o=internet
  changetype: modify
  add: icsDomainNames
  icsDomainNames:siroe.com
  icsDomainNames:example.com
  ```
- Using Delegated Administrator Utility command `commadmin domain modify`, specify the option `-A` to add names of domains to be searched.

  For example:

  ```
  commadmin domain modify -D admin
     -w adminpassword -X hostmachine_1 -d sesta.com
     -A +icsDomainNames:siroe.com
     -A +icsDomainNames:example.com
  ```
- Using Delegated Administrator Console, when creating or editing an organization's properties, you can add domains to the `Invite Calendars in These Organizations` list.

  This adds *icsDomainNames* attributes to the domain LDAP entry. Add one attribute for each external domain to be searched when users in this domain send invitations to an event.

  For more information, see the Delegated Administrator Console online help.

CHAPTER 14

Administering Users, Groups, and Resources

This chapter describes how to use Delegated Administrator and the Calendar Server utilities to manage users, groups, and resources.

This chapter contains the following sections:

14.1 Creating Calendar User LDAP Entries

This section contains instructions for creating new user entries.

This section contains the following topics:

14.1.1 To Create New Calendar Users in Schema Version 2 Mode

This section describes how to create new calendar users for Schema version 2 LDAP entries.

You can use either the Delegated Administrator Console or Utility:

- Delegated Administrator Console

  In the Delegated Administrator Console, use the Create New User wizard. (Click New in the User List page for the organization where the user is to reside.) For more information, see the Delegated Administrator Console online help.
- Delegated Administrator Utility

  Use the commadmin utility user create command. For example, to add user jdoe in the sesta.com domain:

  ```
  commadmin user create -D calmaster -F John -n sesta.com
     -k hosted -l jdoe -w calmasterpassword -W jdoepassword -L Doe -S cal
     -B red.sesta.com -E jdoe@sesta.com
  ```

  For details on all the available options for the commadmin utility, refer to the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*

14.1.2 To Create New Calendar Users For Schema Version 1 Mode

Use the csuser utility. For example, to add user jdoe in the sesta.com domain:

```
csuser -m jdoe@sesta.com -d sesta.com create jdoe
```

14.2 Creating Calendar Group LDAP Entries

This section describes how to create new group LDAP entries

This section contains the following instructions:

- "14.2.1 To Create New Calendar Groups for Schema Version 2 Mode" on page 271
- "14.2.2 To Create New Calendar Groups for Schema Version 1 Mode" on page 271

14.2.1 To Create New Calendar Groups for Schema Version 2 Mode

Groups are named lists of users, resources, or other groups (nested groups). Groups can be either static or dynamic.

Tip – Groups do not contain both static and dynamic members. If an empty group is created, the default is a static group.

You can use one of the following tools:

- Delegated Administrator Console — From the Groups page, click New. The Create New Group wizard appears. The Calendar Service Details screen follows the Mail Service Details screen. You can also assign a service package to the group on the Calendar Service Details screen.

  For more information on the Console, see the Delegated Administrator Console online help.
- Delegated Administrator Utility — Use `commadmin group create`.

  For example:

  ```
  commadmin group create -D chris -n sesta.com -w bolton
  -G testgroup -d sesta.com -m lorca@sesta.com -S mail -M achiko@varrius.com
  ```

  For details on all the available options for the `commadmin` utility, refer to the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*

14.2.2 To Create New Calendar Groups for Schema Version 1 Mode

Add the group LDAP entry directly. Use the Directory Server LDAP commands found in *Sun ONE Directory Server Resource Kit 5.2 Tools Reference*.

The group LDAP entry should include the `icsCalendarGroup` object class, which is an extension of the `GroupofUniqueNames` object class. The following are attributes that can be included:

Attribute	Description
groupid	The is the only required attribute for a group. It is the unique identifier for the group, similar to the `uid` for a user.
icsSecondaryowners	Co-owners of the group.
icsDefaultacl	ACL string for the new group calendar.
`icsCalendar`	The `calid` of the default calendar for this group. It is not required for a group to have a default calendar.
icsStatus	The status of the group calendar. Possible values are: `active`, `inactive`, `deleted`.
icsTimezone	Time zone for the group.
icsDWPHost	The name of the back-end host where the default calendar resides.
icsDoublebooking	Whether or not the default calendar allows multiple events to be scheduled in the same time period. This overrides the domain level preference, bit 15 of *icsAllowRights*. See "To Configure Calendar Server for Groups" on page 125 for domain level defaults for groups.
icsAutoaccept	Whether or not invitations will automatically be accepted for the default calendar.
mail	The email address for this group.
owner	Distinguished Name for the owner's LDAP entry of the group. Must be single valued.

Note – The primary owner is specified by the attribute *owner* from the `GroupOfUniqueNames` object class.

For example, a group LDAP entry might contain:

```
dn: groupid=mygroup, ou=group, o=sesta.com
objectclass:groupofuniquenames
objectclass:icsCalendarGroup
groupid:mygroup
owner:uid=jdoe, ou=people, o=sesta.com
icsSecondaryowners:uid=pfox, ou=people, o=sesta.com
icsStatus:active
uniqueMember: uid=wsmith, ou=people, o=sesta.com
```

For more information about object classes and attributes, see the *Sun Java System Communications Services 6 2005Q4 Schema Reference*.

14.3 Creating Calendar Resource LDAP Entries

This section describes how to create new resources.

Use one of the following ways to create a calendar resource entry:

- "14.3.1 To Create New Calendar Resources for Schema Version 2 Mode" on page 273
- "14.3.2 To Create New Calendar Resources for Schema Version 1" on page 273

14.3.1 To Create New Calendar Resources for Schema Version 2 Mode

This section contains instructions for creating new resource LDAP entries in Schema version 2 mode.

You can use either the Delegated Administrator Console or Utility:

- Delegated Administrator Console

  In the Delegated Administrator Console, use the Create New Resource wizard. (Click New in the Calendar Resources tab for the organization where the resource is to reside.) For more information, see the Delegated Administrator Console online help.
- Delegated Administrator Utility

  Use the commadmin utility rescource create command to create an LDAP entry. For example, to add the conference room Conference_Room_100, use the following command:

  ```
  commadmin resource create -D calmaster
     -w calmasterpassword -n sesta.com -c room100
     -N Conference_Room_100
  ```

  You must then use csresource to create the actual resource calendar. For information on how to create a resource calendar, see "15.5 Creating Calendars" on page 298

  For details on all the available options for the commadmin utility, refer to the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*

14.3.2 To Create New Calendar Resources for Schema Version 1

Use the csresource utility to create both the LDAP entry and the resource calendar. For example, to add a projector, p101, use the following command:

```
csresource -m p101@siroe.com -c p101 create Projector_101
```

For more information on `csresource`, see the "D.15 `csresource`" on page 438.

14.4 Adding the mail Attribute to User, Group and Resource LDAP Entries

This section contains conceptual information and instructions for enabling LDAP entries for mail service.

This section covers the following topics:

- "14.4.1 Overview of Adding Mail Service to Calendar Server LDAP entries" on page 274
- "14.4.2 To Check if the mail Attribute Exists in the LDAP Entry" on page 274
- "14.4.3 To Add the Mail Attribute to Existing User, Group and Resource LDAP Entries for Calendar Server Version 6.3" on page 275

14.4.1 Overview of Adding Mail Service to Calendar Server LDAP entries

Calendar Server requires users, groups and resources to have the *mail* attribute, which contains the email address of the user, group or resource. This enables people to search for calendars using either an email address or a `calid`. When you create new users with Delegated Administrator, it adds the *mail* attribute automatically. This happens even if the user has not been assigned mail service. However, if your users and resources were created in a version of Calendar Server when the *mail* attribute was not required, you might have to add the *mail* attribute to your existing user and resource LDAP entries.

Note – Adding the *mail* attribute does not enable email notifications for user calendars.

Calendar Server does not support email notifications for group, or resource calendars.

To enable email notification for user calendars, add the following two attributes to the user's LDAP entry:

- `icsExtendedUserPrefs:ceNotifyEnable=1`
- `icsExtendedUserPrefs:ceNotifyEmail=jdoe@sesta.com`

14.4.2 To Check if the mail Attribute Exists in the LDAP Entry

If you do not know if your users, groups and resources have the *mail* attribute, for Schema version 2 environments, use Delegated Administrator to check for mail services.

For Schema version 1 environments, use the `csattribute list` command with the `-v` (verbose) option.

For example, to check if the conference room resource Room100 has the *mail* attribute, you would issue the following command:

```
csattribute -v list Room100
```

The output tells if the *mail* attribute is present:

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

If the *mail*attribute exists, you do not have to add it. If the attribute does not exist, then add it as shown in the section that follows.

14.4.3 To Add the Mail Attribute to Existing User, Group and Resource LDAP Entries for Calendar Server Version 6.3

If you are converting existing LDAP entries to calendar enabled entries, you must add the *mail* attribute to each user, group and resource LDAP entry that does not contain it.

To add the *mail* attribute to existing users, groups and resources, use one of the following methods:

- Use Delegated Administrator Utility for a Schema version 2 Environment.

  Use the `commadmin user|resource|group modify -A` option.

  For example: `commadmin group modify -A +mail:jdoe@sesta.com`
- Use the Calendar Server "D.3 csattribute" on page 405 utility for a Schema version 1 environment.

  The following example adds the LDAP *mail* attribute for an existing conference room named Room100 on the sesta.com server:

  ```
  csattribute -a mail=Room100@sesta.com add Room100
  ```
- Use `ldapmodify` to add the attribute directly to the LDAP entries for either Schema version.

14.5 Administering Existing Users

This section contains conceptual information and instructions for administering user entries in an LDAP database. It does not include creating user entries. For information on creating user entries, see"14.1 Creating Calendar User LDAP Entries" on page 269.

Administer users with either the Delegated Administrator Utility or Console for Schema version 2 LDAP user entries, or the csuser utility for Schema version 1 LDAP user entries.

The administrative tasks covered in this section are the following:

- “14.5.1 Displaying Calendar User Information” on page 276
- “14.5.2 Disabling a Calendar User” on page 277
- “14.5.3 Enabling a Calendar User” on page 278
- “14.5.4 Adding Calendar Service to a User” on page 279
- “14.5.5 Deleting Calendar Service from a User LDAP Entry” on page 280
- “14.5.6 Setting Up Email Aliases for a Calendar User” on page 281
- “14.5.7 Verifying that a User has Calendar Service” on page 282
- “14.5.8 Deleting Users from the LDAP Database” on page 283
- “14.5.9 Renaming Calendar Users” on page 284
- “To Disable Users from Having Publicly Writable Calendars” on page 285

14.5.1 Displaying Calendar User Information

This section shows two command examples using the Calendar Server utilities command, `csuser list`, to obtain a list of all calendar users, or to display a particular user's calendar attributes (from the LDAP user entry).

This section contains the following topics:

- “14.5.1.1 To Display All Users Enabled for Calendaring” on page 276
- “14.5.1.2 To Display a Particular User's Calendar Attributes ” on page 276

14.5.1.1 To Display All Users Enabled for Calendaring

To display all users enabled for calendaring, issue the following command-line utility:

```
csuser list
```

14.5.1.2 To Display a Particular User's Calendar Attributes

To display all of the calendar attributes for a single user, issue the following command-line utility:

```
csuser -v list fully-qualified-user-name
```

For example, if the user is `jsmith` who belongs in the `sesta.com` domain, the command-line would be the following:

```
csuser -v list jsmith@sesta.com
```

14.5.2 Disabling a Calendar User

The purpose of disabling a user is to prevent the user from logging into Calendar Server. This is handled differently depending on which user management tool you used to create the user. Users created in the Delegated Administrator Console should be administered using it also. Likewise, if you assigned calendar service to the user with Delegated Administrator Utility, use it to remove the service. Each handles the situation a bit differently.

This section contains the following topics:

14.5.2.1 To Disable a User with Delegated Administrator Console

In the Delegated Administrator Console, it is not possible to merely temporarily inactivate a user. You must remove calendar services from the user. To do this, select the user from the User List page. In the Properties for this user, delete the service package with calendar service in it. This disables the user for calendar, including setting the user's `icsStatus` to *inactive*.

Note – If the package also contains other services, you will have to reassign those services using another package that does not contain calendar.

14.5.2.2 To Disable a User with Delegated Administrator Utility (`commadmin user delete`)

To prevent a user from accessing calendar services, remove the service from the user's LDAP entry, as shown in the example that follows:

```
commadmin user delete jsmith -S cal
```

This removes calendar service from the user without completely removing the LDAP entry. In addition, this command changes the user's `icsStatus` to *inactive*.

14.5.2.3 To Disable a User with Calendar Server Utilities (`csuser disable`)

The `disable` command prohibits a user from accessing calendar data, but it does not remove calendar service from the user's LDAP entry or the Calendar Server database. The utility signals the user being disabled by adding `icsAllowedServiceAccess="http"` to the user LDAP entry.

For example, to disable `jsmith` from accessing Calendar Server:

```
csuser disable jsmith
```

If jsmith is currently logged into Calendar Server, jsmith retains access to calendar data until he logs off.

14.5.2.4 To Remove Calendar Service from a User with Calendar Server Utilities

To remove calendar service from a user, use the csuser utility reset command.

For example, to remove calendar service from jsmith:

```
csuser reset jsmith
```

Doing this removes all of the calendar attributes from the user's LDAP entry, including icsCalendarUser (object class), *icsSubscribed*, *icsCalendarOwned*, *icsCalendar*, and *icsDWPHost* (if using LDAP CLD). A Calendar Server administrator will not be able to create calendars on the user's behalf.

Note –

Calendar Service is restored to the user when one of the following happens:

- The user logs back into Calendar Server (with autoprovisioning turned on).
- The Calendar Server administrator issues a csuser enable command. In this case, the *icsDWPHost* attribute is not restored with the command. You must add it separately.
- The Calendar Server administrator specifically adds the object class and attributes to the user LDAP entry.
- You have recently migrated to Schema version 2 and use the Delegated Administrator to add calendar service.

14.5.3 Enabling a Calendar User

This section contains information about how to enable calendar service for a user.

When users are created, they are normally enabled for calendar service. However, it is possible that a user can be disabled. To re-enable the user for calendar services, you must use one of the methods in this section.

Caution – Enabling a user is implemented slightly differently, between the Delegated Administrator Console and the Utility. Thus, you should use the same tool for both enabling and disabling a user. Do not disable using one tool and re-enable using another.

This section covers the following methods of enabling a user:

- “14.5.3.1 To Enable a User with Delegated Administrator Console” on page 279
- “14.5.3.2 To Enable a User with Delegated Administrator Utility” on page 279
- “14.5.3.3 To Re-Enable a User with Calendar Server Utilities” on page 279

14.5.3.1 To Enable a User with Delegated Administrator Console

You can not disable a user from the Console. You can remove calendar service and then re-add them. To re-add the service, select the user from the User List page and use the Assign Service Package wizard to add the calendar service package to the user's LDAP entry. The user is automatically enabled.

Note – This is the same procedure used to add calendar service (“14.5.4 Adding Calendar Service to a User” on page 279).

14.5.3.2 To Enable a User with Delegated Administrator Utility

Delegated Administrator Utility can enable a user with either one of the following choices:

- Enabling the user by changing the `icsStatus` to *active*.

  `commadmin user modify -A icsStatus:active` to enable the user.
- Adding the calendar service to the user LDAP entry.

  ```
  commadmin user modify -S cal
  ```

Caution – Be sure to use the same method to enable and disable a user. If you tried to enable a user with Delegated Administrator Console, after having disabled the user with Delegated Administrator Utility (changing the *icsStatus* only), the system will not be able to add service since the user already has service and the user will still be disabled.

14.5.3.3 To Re-Enable a User with Calendar Server Utilities

To re-enable a user for calendar service, use `csuser enable` to remove `icsAllowedServiceAccess="http"` from the user's LDAP record.

14.5.4 Adding Calendar Service to a User

It is not necessary to add calendar service to a user created with the old (Schema version 1) Calendar Server utilities. However, with (Schema version 2) Delegated Administrator, calendar service can be added and removed from a user's LDAP entry.

To add calendar service to an existing user, use one of the following tools:

- "14.5.4.1 To Add Calendar Service to a User with Delegated Administrator Console" on page 280 (for Schema version 2)
- "14.5.4.2 To Add Calendar Service to a User with Delegated Administrator (`commadmin user create`)" on page 280(for Schema version 2)
- "14.5.4.3 To Add Calendar Service with Calendar Server Utilities" on page 280 (for Schema version 1).

14.5.4.1 To Add Calendar Service to a User with Delegated Administrator Console

You can add calendar services to both a new user and an existing user:

- When a new user is created, using the New User wizard, it assigns the user a service package that includes calendar service. The user is automatically enabled.
- For an existing user, select the user from the User List page and use the Assign Service Package wizard to select a service package with calendar service. The user is automatically enabled.

14.5.4.2 To Add Calendar Service to a User with Delegated Administrator (`commadmin user create`)

When creating a new user, add calendar services as illustrated in the example that follows:

```
commadmin user create jsmith -S cal
```

If you did not add calendar services when the user was created, you can add calendar services to the user later, using a modify command, as illustrated in the following example:

```
commadmin user modify jsmith -S cal
```

14.5.4.3 To Add Calendar Service with Calendar Server Utilities

If you used `csuser create` when you created the user entry, the utility gives calendar service to the user by adding *icsCalendarUser* and its attributes to the user LDAP entry.

14.5.5 Deleting Calendar Service from a User LDAP Entry

One way to deny calendar service to a user is to remove the service from the user entry. Another way is to disable the user temporarily. These are covered in the earlier section titled "14.5.2 Disabling a Calendar User" on page 277.

14.5.6 Setting Up Email Aliases for a Calendar User

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

You can add the attribute in the following three ways:

- "To Set Up Email Aliases with Delegated Administrator Console" on page 281
- "14.5.6.1 To Set Up Email Aliases with Delegated Administrator Utility" on page 281
- "14.5.6.2 To Set Up Email Aliases with Calendar Server Utility csattribute" on page 281

, using `commadmin user modify -A` or by directly updating LDAP with `ldapmodify`.

Note – To enable these changes, you might also need to rebuild alias tables or configurations. Refer to the documentation for Messaging Server (or your email product) as well as your site's own documentation and procedures regarding changes to mail services. Messaging Server documentation is available on this at: `http://docs.sun.com/coll/1312.2`.

▼ To Set Up Email Aliases with Delegated Administrator Console

1 **Choose the organization where the user resides.**

2 **Search for the user.**

3 **Display the user's properties by clicking the user's name.**

4 **Edit the Mail Services Details to add aliases.**

See Also The Delegated Administrator online help.

14.5.6.1 To Set Up Email Aliases with Delegated Administrator Utility

Email aliases can be set up for calendar users, just like messaging users, by adding *mailalternateaddress* to the user's LDAP entry. To add the attribute using Delegated Administration Utility, use `commadmin user modify -A mailalternateaddress:value`.

14.5.6.2 To Set Up Email Aliases with Calendar Server Utility csattribute

To add email aliases to a user, use the `csattribute add -a` command to add *mailalternateaddress* attributes to the user entry.

For example, to add two aliases for a user named John Smith with these values:

- With a *mail* attribute of: `johnsmith@sesta.com`
- Email aliases: `johns@sesta.com` and `jsmith@sesta.com`

The commands would look like the following examples:

```
csattribute -a mailalternateaddress=johns@sesta.com add johnsmith@sesta.com

csattribute -a mailalternateaddress=jsmith@sesta.com add johnsmith@sesta.com
```

14.5.7 Verifying that a User has Calendar Service

This section provides instructions for verifying calendar service.

Use the following tools to verify that a user has calendar service.

- "14.5.7.1 To Check if a User has Calendar Service with Delegated Administrator Console" on page 282
- "14.5.7.2 To Check if a User has Calendar Service with Delegated Administrator Utility" on page 282
- "14.5.7.3 To Check if a User has Calendar Service with Calendar Server Utility `csuser`" on page 282

14.5.7.1 To Check if a User has Calendar Service with Delegated Administrator Console

If there is a Calendar Service Details page, then they have calendar services. Or, look in the service package details to see what kinds of services are listed.

14.5.7.2 To Check if a User has Calendar Service with Delegated Administrator Utility

Use the following command to list all the directory properties associated with the user:

```
commadmin user search
```

14.5.7.3 To Check if a User has Calendar Service with Calendar Server Utility `csuser`

Use the following command to check if the user is enabled for calendar service:

```
csuser check
```

14.5.8 Deleting Users from the LDAP Database

Use either Delegated Administrator or the Calendar Server Utilities to delete a user from LDAP.

Use one of the two methods that follow to delete users from the LDAP database:

- "Deleting Users in Schema Version 2 Using Delegated Administrator" on page 283
- "14.5.8.1 Deleting Users in a Schema Version 1 Environment" on page 284

Caution – There is no `undelete` command.

Once users in a domain are deleted using Delegated Administrator, they must be purged and re-added from scratch. User names can not be reused until the purge happens.

▼ Deleting Users in Schema Version 2 Using Delegated Administrator

You can mark users for deletion with either Delegated Administrator interface. However you can not actually remove users from LDAP (purge) using the Delegated Administrator Console. You must use the Delegated Administrator Utility for that. The following task lists the steps for deleting a user from LDAP. The user is not actually removed from LDAP until the last step is complete.

1 **Mark a user entry for deletion.**

For Delegated Administrator Console: Select the users to delete in the User List page and click Delete.

For Delegated Administrator Utility: Use the `commadmin` user delete command. For example:

```
commadmin user delete -D chris -n siroe.com
-w bolton -l jsmith
```

In both cases the *icsStatus* attribute in the user LDAP entry is changed from `active` to `deleted`.

2 **Use the Calendar Server Utility `csclean` to remove all calendars belonging to all deleted users in one or all domains, as shown in the following example:**

```
csclean clean "*"
```

Or to remove calendars belonging to all deleted users in one domain, specify the actual domain, as shown in the following example: `csclean clean sesta.com`

Tip – If you inadvertently purge the users from LDAP before deleting the users' calendars, you can remove them later using the `cscal` utility, as described in "15.6 Managing User Calendars" on page 304.

3 **Purge the domain of all users marked for deletion, using Delegated Administrator Utility command `commadmin domain purge`.**

For example:

```
commadmin domain purge -D chris -d sesta.com -n siroe.com -w bolton
```

In this example, all users in `sesta.com` that are marked as deleted will be purged, that is, permanently removed.

Tip – Run this utility manually from time to time to clean up your LDAP directory. For more information about this command, see the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

14.5.8.1 Deleting Users in a Schema Version 1 Environment

To remove the specified user's LDAP entry and the user's default calendar, use the Calendar Server utility `csuser` with the `delete` command.

For example, to delete the LDAP entry and the default calendar for user `jsmith` use the following command:

```
csuser delete jsmith
```

If you wish to remove the other calendars belonging to this user, you must use `cscal` as described in "15.6 Managing User Calendars" on page 304.

14.5.9 Renaming Calendar Users

If one or more user ID's need to be changed, run the `csrename` utility.

This utility performs the following steps:

- Converts the user ID's in the Calendar Server LDAP attributes (the ones with the `ics` prefix). The LDAP directory is updated in place.
- Renames the users in events and tasks on the Calendar Server database files. It writes the new database to a destination directory. Existing database files are not modified.

Note – Be aware that changing even one user ID causes the whole database to be rewritten. So this is a "costly" utility to run.

For further information on the `csrename` utility, see Appendix D, "Calendar Server Command-Line Utilities Reference."

14.5.10 Turning the Publicly Writable Calendars Feature Off

Publicly Writable Calendars is a Calendar Server feature. You can turn this feature on or off. It is enabled by default. The following task shows how to edit the configuration file to change the setting.

With this feature enabled, calendars can be scheduled (written to) when an invitation is generated. The event will automatically be added to the attendees' calendars.

With this feature disabled, calendar owners will get an email notification only when an invitation is generated. The event will not be automatically added to the attendees' calendars. Only the owners are allowed to add events and tasks to the calendar.

▼ To Disable Users from Having Publicly Writable Calendars

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit the following** `ics.conf` **parameter as shown in the following table:**

Parameter	Description and Default Value
service.wcap. *allowpublicwritablecalendars*	Enables users to have publicly writable calendars. This is enabled by default (set to “`yes`”).

6 **Save the file as** `ics.conf`**.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

14.6 Administering Calendar Server Resources

This section contains conceptual information and instructions on administering calendar resources.

After your resources are added, you can administer them using Delegated Administrator or `csresource`.

This section contains the following topics:

- "14.6.1 Retrieving LDAP Information for Resources" on page 286
- "To Enable Resources" on page 287
- "To Disable Resources" on page 287
- "To Delete Resources" on page 287
- "14.6.2 To Set Up a Bitbucket Channel for Resource Email" on page 288

14.6.1 Retrieving LDAP Information for Resources

This section contains instructions for retrieving resource LDAP information.

You can retrieve resource properties from the LDAP resource entry with one of three tools:

- "To Retrieve Resource Information Using Delegated Administrator Console" on page 286
- "14.6.1.1 To Retrieve Resource Information Using Delegated Administrator Utility" on page 286
- "To Retrieve Resource Information Using `csresource`" on page 287

▼ To Retrieve Resource Information Using Delegated Administrator Console

1. **In the Delegated Administrator Console, click the Calendar Resources tab.**

2. **Use the Search Results drop-down box to select one of the following options:**

   - Search Calendar Resource by Resource ID
   - Search Calendar Resource by Calendar Resource Name

3. **Type the value you want to search for.**

4. **Click Search.**

14.6.1.1 To Retrieve Resource Information Using Delegated Administrator Utility

Use the `commadmin resource search` command to retrieve LDAP information for a resource.

For example, to search for the resource `CF101` in the `sesta.com` domain, use the following command:

```
commadmin resource search -D serviceadmin -w serviceadmin -n sesta.com \s
-d sesta.com -u CF101
```

▼ To Retrieve Resource Information Using `csresource`

You can retrieve the LDAP entry information for a resource or for all resources using the `csresource` utility.

1. **Change to the `sbin` directory.**

2. **Use the `csresource list` command to list one or all resources.**

   For example, list all the information about all the resources:

   ```
   ./csresource -v list
   ```

   Or, list all the information about a specific resource, `CF101`:

   ```
   ./csresource
   ```

▼ To Enable Resources

1. **Change to the `sbin` directory.**

2. **Use the `csresource enable` command to enable one or more resources.**

   For example, to enable the `ConfRm12` resource:

   ```
   ./csresource -v enable ConfRm12
   ```

▼ To Disable Resources

1. **Change to the `sbin` directory.**

2. **Use the `csresource disable` command to disable one or more resources. For example, to disable the *ConfRm12* resource:**

   ```
   ./csresource -v disable ConfRm12
   ```

▼ To Delete Resources

1. **Change to the `sbin` directory.**

2. **Use the `csresource delete` command to delete one or more resources. For example, to delete the *ConfRm12* resource:**

   ```
   ./csresource -v delete ConfRm12
   ```

14.6.2 To Set Up a Bitbucket Channel for Resource Email

This section contains directions for setting up a bitbucket channel for both Messaging Server and Sendmail. The bitbucket channel is a way to discard 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.

This section contains the following procedures:

- "To Set up the Messaging Server Bitbucket Channel" on page 288
- "To Set up a Sendmail Bitbucket Channel" on page 288

▼ To Set up the Messaging Server Bitbucket Channel

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 *csattribute* utility:**

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

▼ To Set up a Sendmail Bitbucket Channel

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 *csattribute* utility:**

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

14.7 Managing User and Resource LDAP Attributes

Manage LDAP attributes used by Calendar Server, with the "D.3 csattribute" on page 405 utility, or `ldapmodify`. Attributes can be listed, added, or deleted with `csattribute`. To modify an attribute, use `ldapmodify`.

This section contains the following topics:

- "To List LDAP Entry Attributes" on page 289
- "To Add an LDAP Entry Attribute" on page 289
- "To Delete an LDAP Entry Attribute" on page 289
- "14.7.1 To Modify an LDAP Entry Attribute" on page 290

▼ To List LDAP Entry Attributes

1 **Log in as the user or group under which Calendar Server is running (such as** `icsuser` **and** `icsgroup`**) that was specified during installation, or as** `root`

2 **Change to the** `sbin` **directory.**

3 **Use the** `csattribute list` **command to list the attributes for a user or a resource. For example, to list the attributes for** `tchang@sesta.com`**, issue the following command:**

```
./csattribute -t user -d sesta.com list tchang
```

▼ To Add an LDAP Entry Attribute

1 **Log in as the user or group under which Calendar Server is running (such as** `icsuser` **and** `icsgroup`**) that was specified during installation, or as** `root`

2 **If you want this attribute change to be recognized immediately, stop Calendar Server. Otherwise, you do not have to stop Calendar Server.**

3 **Change to the** `sbin` **directory.**

4 **Use the** `csattribute add` **command to add an attribute to a user or a resource. For example, to add the LDAP attribute** *icsCalendar* **with the value** `Conference_Schedule` **to the user** `tchang`**:**

```
./csattribute -a icsCalendar=Conference_Schedule add tchang@sesta.com
```

▼ To Delete an LDAP Entry Attribute

1 **Log in as the user or group under which Calendar Server is running (such as** `icsuser` **and** `icsgroup`**) that was specified during installation, or as** `root`

2 **If you want this attribute change to be recognized immediately, stop Calendar Server. Otherwise, you do not have to stop Calendar Server.**

3 **Change to the** `sbin` **directory.**

4 **Use the** `csattribute delete` **command to delete an attribute from a user or a resource. For example, to delete the LDAP attribute** *icsCalendar* **with the value** `Conference_Schedule` **from the user** `tchang`**:**

```
./csattribute -a icsCalendar=Conference_Schedule -t user -d sesta.com delete
tchang
```

14.7.1 To Modify an LDAP Entry Attribute

To modify an LDAP entry attribute, use `ldapmodify`. For example, to change the status of user with `uid=tchang`, use `ldapmodify` as shown:

```
dn:uid=tchang,ou=people,o=sesta.com
 changetype: modify
 add: objectclass
 objectClass: icsCalendarUser
 add: icsStatus
 icsStatus: active
```

Note – If your site is using the LDAP CLD plug-in, do not attempt to move a user's calendars from one back-end host to another by changing the value of *icsDWPHost*, using `csattribute`. Modifying *icsDWPHost* does not cause the calendar to be moved to the new back-end host. For instruction on how to move a calendar from one back-end server to another, see "15.6 Managing User Calendars" on page 304.

◆ ◆ ◆ CHAPTER 15

Administering Calendars

This chapter contains topics with instructions on how to use Calendar Server command-line utilities to create and manage calendars.

This chapter contains the following topics:

15.1 Calendar Administration Overview for Calendar Server Version 6.3

The Delegated Administrator does not create or manage calendars. You must use the Calendar Server utilities described in Appendix D, "Calendar Server Command-Line Utilities Reference."

Before creating calendars, you must know the following information:

- There are three types of calendars, user calendars, resource calendars and group calendars.

  User calendars are for scheduling human activity. Resource calendars are for scheduling the use of inanimate objects, such as conference rooms, or video equipment. Group calendars are for scheduling activities for a group of users.
- All types of calendars are identified by a unique calendar identifier (`calid`).
- Create user calendars with `cscal`. (Alternately, you can allow autoprovisioning at login time. See "15.3 Automatic Creation of Calendars" on page 294.

- Create resource calendars with `csresource`. (There is no autoprovisioning of resource calendars.)
- Create group calendars

To run `cscal` or `csresource`, you must log in as a user who has administrative rights to the system where Calendar Server is running. You must run these commands from the `/opt/SUNWics5/cal/sbin` directory. That is, you must change to the `sbin` directory; you can not run them from another directory by specifying the path.

15.2 Creating Calendar Unique Identifiers (calid's)

Each calendar in the Calendar Server database is identified by a unique calendar identifier (ID) or `calid`. When creating calendars, you are required to specify the `calid`.

This section contains the following topics:

15.2.1 Calid Syntax

Each calendar in the database is identified by a unique calendar ID (`calid`). The following `calid` syntax has three parts:

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

The three parts are:

userid A user ID that is unique for the domain in this Calendar Server instance.

domain The name of the user's domain.

With a single domain, the domain part is optional since there is no ambiguity about which domain the user is in.

With multiple domains, if the domain part is not specified, then Calendar Server uses the value specified in the `ics.conf` parameter *service.defaultdomain* for the domain. If the user is not in the default domain, the domain part must be specified.

For more information about multiple domain environments, see Chapter 10, "Setting Up a Multiple Domain Calendar Server 6.3 Environment," and Chapter 13, "Administering Calendar Server Domains."

calendar-name — An optional calendar name that is unique to the specific user. Although an owner has only one default calendar, it is possible to have other calendars for various purposes. Each of these non-default calendars is distinguished by its calendar name. For example, if user John Doe has a `uid` `jdoe`, his default calendar might be `jdoe@sesta.com`. An auxiliary calendar he uses to keep track of baseball games for the Little League team he coaches might be identified with the following `calid`: `jdoe@sesta.com:baseball`.

15.2.2 Calendar ID Creation Rules

This section describes the rules for creating a calendar ID (`calid`).

When creating a `calid`, keep in mind the following rules:

- Calendar ID's are case sensitive. For example, `JSMITH` is not equivalent to `jsmith`. (This 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 (!)

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

    Examples of proper calendar ID's are:

    ```
    jsmithjsmith:private_calendar
    jsmith@calendar.sesta.com:new-cal
    ```

15.2.3 Converting Non-Domain calid's to Multiple Domain Format calid's

If you have `calid`'s that were created before you had domains, and you now want to convert them to domain specific `calid`'s, there is a utility, `csvdmig`, that can be used to add the domain part to your existing `calid`'s. See “3.6 csvdmig” on page 106 for instructions on how to use the utility.

If you do not add domain names to the existing `calid`'s, the system will assume they belong to the default domain.

15.3 Automatic Creation of Calendars

This section contains conceptual information and instructions for using the Calendar Server feature for automatically creating calendars upon a user's first login.

Automatic calendar creation is enabled by default. With it enabled, the system automatically creates calendars under two circumstances:

- The first time a user logs in, the user's LDAP entry in updated to add calendar service, and a default calendar is created. The user entry must already exist in the LDAP directory. If it does not, an error is returned.
- If appropriately configured, the first time a user, group, or resource is invited to an event and there is no existing default calendar, a default calendar is created.

For the configuration information necessary to implement automatic creation of calendars in this circumstance, see “To Configure Calendar Server for Groups” on page 125.

This section covers the following topics:

- “15.3.1 Creating calids” on page 294
- “To Enable Autoprovisioning of Calendars” on page 295
- “To Disable Autoprovisioning of Calendars” on page 295

15.3.1 Creating calids

Calendar Server creates the calendar ID (calid) for new default calendars from the user ID and the domain name.

For example, John Smith's user ID is jsmith, and his LDAP entry resides in the sesta.com domain. The first time he logs into Calendar Server, the system automatically creates a default calendar with jsmith@sesta.com as the calid. Each subsequent calendar John Smith creates has a calid with jsmith@sesta.com: prepended to the calendar name. For example, if John Smith later creates a new calendar named meetings, the calid for the new calendar is jsmith@sesta.com:meetings.

If a user, group, or resource without a default calendar is listed in the attendee list of an event, the system looks up the uid in LDAP in the event owner's domain as the event owner. If no domain is assigned to the owner, the default domain is assumed. The system constructs a calid by appending the domain to the uid.

If the system can't find the uid in the event owner's domain, it will search any other domains the event owner is allowed to search. For more information, see “11.2 Cross Domain Searching in Calendar Server 6.3 Systems” on page 244.

▼ To Enable Autoprovisioning of Calendars

The auto-creaton of calendars is enabled by default. However, if you need to turn it on again after disabling it, perform the steps that follow:

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3 **Change to the** `/etc/opt/SUNWics5/cal/config` **directory.**

4 **Save your old** `ics.conf` **file by copying and renaming it.**

5 **Edit one or more of the following parameters in the Calendar Server configuration file,** `ics.conf`**, as shown in the following table:**

Parameters	Description and Default Value
local.autoprovision	Set to “yes”, allows default calendar creation to occur automatically when the user logs in the first time. autoprovisioning is enabled by default. To turn this feature off, set the value to “no”.

6 **Verify that the user's LDAP entry is enabled for calendar.**

The entry must contain the `icsCalendarUser` object class. Add the class to the user's LDAP entry if it is not there.

7 **If your site is using multiple domains, the user's domain must also be calendar enabled before autoprovisioning will work. The domain entry must contain the** `icsCalendarDomain` **object class.**

8 **Save the file.**

9 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Disable Autoprovisioning of Calendars

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4 **Save your old `ics.conf` file by copying and renaming it.**

5 **Edit one or more of the following parameters in the Calendar Server configuration file, `ics.conf`, as shown in the following table:**

Parameters	Description and Default Value
local.autoprovision	Setting the parameter to `no` disables autoprovisioning of user calendars.

6 **Save the file.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

Note – If autoprovisioning is disabled, calendars must be explicitly created for users before they can successfully log in.

15.4 Calendar Access Control

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 section covers the following topics:

- "15.4.1 Configuration Parameters for Access Control" on page 296
- "15.4.2 Public and Private Events and Tasks Filter" on page 297
- "15.4.3 Command-Line Utilities for Access Control" on page 297

15.4.1 Configuration Parameters for Access Control

The following table describes the configuration parameters in the `ics.conf` file that Calendar Server uses for access control.

TABLE 15–1 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"`

15.4.2 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 — These are confidential events and tasks. 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.

15.4.3 Command-Line Utilities for Access Control

The following table describes the Calendar Server command-line utilities that allow you to set or modify ACLs for access control.

TABLE 15–2 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 or resource calendars.
`csresource`	Use the `csresource` utility with the `-a` option to set ACLs for resource calendars.

`commadmin user`
`csuser`
	For Schema version 2, use Delegated Administrator Console, or Delegated Administrator Utility, `commadmin`, to change the default ACL used when a user calendar is created.
	For Schema version 1, use the `csuser` utility with the `-a` option to change the default ACL used when a user creates a calendar.

Note – To set access rights in the Delegated Administrator Console, from the Organization Properties page (also from the Create New Organization wizard), click the Advanced Rights button to see the list of access rights that can be administered from the console.

15.5 Creating Calendars

This section contains conceptual information and instructions on how to create calendars.

This section contains the following topics:

- "15.5.1 Creating a User Calendar with the `cscal`Utility" on page 298
- "15.5.2 Configuring the Calendar Server for Resources" on page 301
- "15.5.3 Creating Resources and Resource Calendars" on page 302

15.5.1 Creating a User Calendar with the `cscal`Utility

This section contains the following topics and examples:

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
```

The string `-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.

15.5.1.1 Overview of Creating a New Calendar

To create a new calendar, use the cscal utility with the create command. The user or resource entry must already exist in the LDAP directory. Refer to Chapter 14, "Administering Users, Groups, and Resources"for information on adding users and resources to your LDAP directory.

If your site is using the LDAP Calendar Lookup Database (CLD) plug-in, you must create all of the calendars for a particular user or resource on the same back-end server, as indicated by the *icsDWPHost* LDAP attribute in the user or resource entry. 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 Chapter 5, "Configuring Calendar Database Distribution Across Multiple Machines in Calendar Server Version 6.3."

15.5.1.2 Creating New Calendars

To create a new calendar, the minimal command is the following:

```
cscal -o uid  create calid
```

For example, for the user John Smith with a unique ID and calendar ID of jsmith the command would look like the following:

```
cscal -o jsmith create jsmith
```

The command has the following parts:

cscal	The name of the utility.
-o	The unique ID (uid) of the primary owner for this calendar.
create	The command to create the new calendar.
calid	The calendar ID to be assigned to this calendar.

For more information about the cscal utility, see "D.5 cscal" on page 409 also located in this guide.

Tip – The default access control settings are defined by *calstore.calendar.default.acl* in the ics.conf file.

15.5.1.3 Creating Another Calendar for a User

You can create multiple calendars for any user. However, they are always identified as sub-calendars of the default calendar. The fully qualified name of the new calendar will have the default calendar name on the left of a colon separator and the new calendar's name on the right of the colon separator.

The following example demonstrates how to create another (non-default) calendar for a user, John Smith, with the name of the new calendar as `Personal`:

```
cscal -o jsmith@sesta.com create Personal
```

The parts of the command are as follows:

`cscal` The name of the utility.

`-o jsmith@sesta.com` The unique ID (`uid`) of the primary owner for this calendar.

`create` The command to create the new calendar.

Personal The second half of the calendar ID (`calid`) to be assigned to this calendar.

The fully qualified calendar ID is `jsmith@sesta.com:Personal`.

15.5.1.4 Creating a Calendar with a Viewable Name

This example shows how to give a separate viewable name, "Hobbies", to the Personal non-default calendar created in the previous example.

```
cscal -o jsmith@sesta.com -n Hobbies create Personal
```

`-o` `jsmith@sesta.com` Specifies the user ID of the primary owner.

`-n` `Hobbies` Specifies the viewable name of the calendar.

`Personal` The name of this new additional calendar for John Smith.

The entire `calid` becomes: `jsmith@sesta.com:Personal`.

15.5.1.5 Creating a Calendar with Other Properties

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

```
cscal -n Hobbies -o jsmith -g sports -k yes -y rjones create Personal
```

The command has the following parts:

`cscal` The name of the utility.

`-o jamsith@sesta.com` The unique ID (`uid`) of the primary owner for this calendar.

`-g sports` This option associates the calendar, Personal, with a category named `sports`.

`-y`	The value `rjones@sestas.com` specifies another owner of the calendar.
`-k yes|no`	This option enables or disables double booking of events in one time slot. A value of `yes` enables double booking. A value of `no` would disable double booking.
`create`	The command to create the new calendar.
`Personal`	The calendar ID to be assigned to this calendar.

15.5.2 Configuring the Calendar Server for Resources

A resource calendar is associated with things that can be scheduled, such as meeting rooms, notebook computers, overhead projectors and other equipment. Resource calendars require access control lists.

As shown in table Table 15–3, two configuration parameters in the `ics.conf` file apply to resource calendars:

resource.default.acl	A default access control list.
resource.allow.doublebook	A parameter that allows or disallows doublebooking.

To change the default values for these parameters (shown in table Table 15–3), edit the `ics.conf` file. Changes to the default values will apply only to new resource calendars; it will not change the values for existing resources.

For Schema version 1, use the Calendar Server Utility `cscal` to change values for an existing resource calendar. The `csresource` utility does not have a `modify` command.

For Schema version 2, use the Delegated Administrator Utility command `commadmin resource modify`. The Delegated Administrator Console does not allow you to change these values for calendar resources.

Note – The Calendar Server notification software is not programmed to send notifications to resources, only to users.

TABLE 15–3 Resource Calendar Configuration Parameters in the `ics.conf` file

Parameter	Description and Default Value
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.
resource.allow.doublebook	This parameter determines if a resource calendar allows doublebooking. Doublebooking allows a resource calendar to have more than one event scheduled for the same time. The default value is `"no"`— Do not allow doublebooking. To allow doublebooking for a resource calendar, use the `-k` option when you create the calendar using the `csresource` utility create command.
resource.invite.autoprovision	The default value is `"yes"`.
resource.invite.autoaccept	The default value is `"yes"`.

15.5.3 Creating Resources and Resource Calendars

Tip – If the value of the `ics.conf` parameter *resource.invite.autoprovision* is `"yes"`, the resource calendar will be created upon first invitation. That is, if this resource does not already have a default calendar, the first time its scheduled in an invitation, a resource calendar will be created.

To create a resource, use one of the following methods:

Calendar Server Utility (Schema version 1)
Use `csresource create`

This utility creates both the LDAP entry and the default calendar for the resource.

If there is an existing LDAP entry for the resource, `csresource` creates only the calendar. It will not create a duplicate LDAP entry.

For example, to create a resource LDAP entry and calendar with the calendar ID `aud100`, viewable name `Auditorium`with the default settings, use the following command:

```
csresource -m aud100@siroe.com -c aud100 create Auditorium
```

Delegated Administrator Utility and Calendar Server utility

Use a combination of two commands:

- The Delegated Administrator Utility command `commadmin resource create` to create the LDAP entry.
- The Calendar Server Utility command `csresource create` to create the default calendar.

Delegated Administrator Console

To create the LDAP resource with the Console, select the organization where this resource will reside from the Organizations List. From the Calendar Resources page for this organization, click New to bring up the Create New Calendar Resource Wizard.

For more information about the Delegated Administrator Utility, see *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

For more information about the Delegated Administrator Console, see the online help.

For more information about `csresource`, see Appendix D, "Calendar Server Command-Line Utilities Reference."

15.5.4 Allowing Double Booking of Resource Calendars

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.

The following command creates a resource LDAP entry and calendar, 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@sesta.com` as another owner:

```
csresource -m aud100@siroe.com -c aud100 -k yes
    -o bkamdar -y jsmith@sesta.com create Auditorium
```

15.5.5 Limiting Access to Resource Calendars

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.

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

15.6 Managing User Calendars

This section contains instructions on how to manage user calendars using the Calendar Server utility "D.5 cscal" on page 409.

This section covers the following administrative tasks:

15.6.1 To Display Calendars

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

The following examples demonstrate three different tasks using `cscal`.

- 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
  ```

15.6.2 To Delete a Calendar

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.

The following two examples demonstrate different tasks you can accomplish with `cscal delete`:

- To delete a specific calendar with the calendar ID `jsmith@sesta.com:meetings`:

  ```
  cscal delete jsmith@sesta.com:meetings
  ```

- To delete all calendars whose primary owner is `jsmith@sesta.com`:

  ```
  cscal -o jsmith@sesta.com delete
  ```

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 17, "Backing Up and Restoring Calendar Server Data."

15.6.3 To Remove Calendars of Deleted Users

If you have deleted one or more users with the Calendar Server Utility command `csuser delete`, or with Delegated Administrator Console or Utility, calendars owned by that user might still be present in the database.

There are two ways to remove users' calendars. The method to use depends on the tool you used to delete the user:

Calendar Server utility `csuser`

The `csuser` utility removes the user from the LDAP directory and removes the user's default calendar, but not any other calendars the user might own. For instructions on how to use `cscal` to remove these calendars, see "To Remove All Calendars of a User Deleted with csuser in Calendar Server Version 6.3" on page 306.

Delegated Administrator Console and Utility

Delegated Administrator does not remove any calendars. Use Delegated Administrator to mark users for deletion and then use Calendar Server Utility `csclean` to remove calendars for user's marked for deletion.

For instructions on how to remove deleted users' calendars using `csclean`, see "To Remove All Calendars for Users Deleted by Delegated Administrator" on page 306.

For instructions on using Delegated Administrator Utility, see the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide*.

For instructions on using Delegated Administrator Console, see the online help.

▼ To Remove All Calendars of a User Deleted with csuser in Calendar Server Version 6.3

1. **Run the `cscal list` command to find all of the calendars for the deleted owner's `uid`.**

   `cscal -o` *owner list*

2. **Use the `cscal` command to remove all the calendars for this owner.**

   `cscal -o` *owner delete*

3. **Verify that all the calendars have been removed by running `csuser list` again.**

Note – Use this procedure if you used `commadmin` to mark the user as deleted, and the user's LDAP entry has already been purged.

▼ To Remove All Calendars for Users Deleted by Delegated Administrator

Delegated Administrator does not remove calendars. Use the `csclean` utility to remove all calendars for any users marked as deleted with Delegated Administrator.

1. **Use `csclean` to remove all calendars for users marked as deleted but not yet purged.**

   For example, to remove all the calendars for users marked as deleted in the `sesta.com` domain in the last 10 days, the command would be as follows:

   ```
   csclean -g 10 clean sesta.com
   ```

2. **If the user has already been purged from the LDAP, then you must use `cscal`.**

   For instructions, see "To Remove All Calendars of a User Deleted with csuser in Calendar Server Version 6.3" on page 306.

15.6.4 To Enable a Calendar

To allow users to access their calendars, you must first enable the calendars using the `cscal enable` command.

The following examples, demonstrate how to enable calendars:

- To enable calendar `jsmith@sesta.com:meetings` using the default configuration settings:

  ```
  cscal enable jsmith@sesta.com:meetings
  ```
- To enable the calendar `jsmith@sesta.com:meetings` but not allow doublebooking:

  ```
  cscal -k no enable jsmith@sesta.com:meetings
  ```

15.6.5 To Disable 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@sesta.com:meetings`, use the following command:

```
cscal disable jsmith@sesta.com:meetings
```

15.6.6 To Modify 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@sesta.com` as another owner:

```
cscal -a "@@o^c^wd^g" -y RJones@sesta.com modify AllAdmins
```

The following explains the two command variables used in the preceding example:

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

15.6.7 To Remove Properties From a Calendar

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

The three examples that follow show how to remove different properties:

- To remove a description from jsmith@sesta.com:meetings:

  ```
  cscal -d "" modify jsmith@sesta.com:meetings
  ```
- To remove all categories from the jsmith@sesta.com:meetings calendar:

  ```
  cscal -g "" modify jsmith@sesta.com:meetings
  ```
- To remove “other owners” from jsmith@sesta.com:meetings:

  ```
  cscal -y "" modify jsmith@sesta.com:meetings
  ```

15.6.8 To Recover a “Lost” Default Calendar

If a user’s default calendar is not visible to the Communications Express user interface client, but still exists in the database, you can recover the calendar and make it visible again by updating two attributes in the user’s LDAP entry.

To recover a calendar, make sure the value of the following attributes in the user's LDAP entry is the user's fully qualified calid:

- icsCalendar
- icsSubscribed

For Schema version 2, use one of the following methods to update the attributes:

- Use the ldapmodify Directory Server utility.
- Use the Calendar Server Utility command csuser reset.
- Use the Delegated Administrator Utility command commadmin user modify.
- Use the Delegated Administrator Console to add the default calendar name by editing the User Properties page.

For Schema version 1, use the csattribute add command to update the attributes.

▼ To Move a User Calendar to a Different Back-End Server

To move a user 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 “D.19 csuser” on page 448 utility. For example to disable the user with the user ID and calid bkamdar:**

   ```
   csuser disable bkamdar
   ```

2 **On the original server, export each of the user's calendars from the calendar database to a file using the "D.10 `csexport`" on page 429 utility. For example:**

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

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

4 **On the new server, for each of the calendars exported, import the calendar from the file to the calendar database using the "D.11 `csimport`" on page 430 utility. For example:**

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

5 **On the LDAP directory server, update the calendar owner's *icsDWPHost* LDAP attribute to point to the new back-end server using the "D.3 csattribute" on page 405 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 "D.19 `csuser`" on page 448 utility for a user 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`.

Note – If you are using the CLD cache option, after moving 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 finding a calendar after it has been moved.

To clear the CLD cache, follow these steps:

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

15.7 Administering Resource Calendars

This section describes how to administer resource calendars using the `csresource` utility.

The following are procedures for administering resource calendars:

- "15.7.1 To Display Resource Calendars and Attributes" on page 310
- "15.7.2 To Modify a Resource Calendar" on page 310
- "15.7.3 To Disable or Enable a Resource Calendar" on page 311
- "15.7.4 To Delete a Resource Calendar" on page 311
- "To Move a Resource Calendar to a Different Back-End Server" on page 311

15.7.1 To Display Resource Calendars and Attributes

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

For example, use the utility to perform the following tasks:

- 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
  ```

15.7.2 To Modify a Resource Calendar

This section describes how to modify a resource calendar. You must use the "D.5 `cscal`" on page 409 utility command because the `csresource` utility does not have a `modify` command.

For example, the following command performs two tasks simultaneously:

- It sets the owner `uid` to `tchang`.
- It specifies another owner whose`uid` is `mwong`.

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

In this example, the `cscal` utility requires that resource's `calid` (`aud100`) is specified rather than the calendar name (`Auditorium`).

15.7.3 To Disable or Enable 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
```

15.7.4 To Delete a Resource Calendar

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

For example, to delete the Auditorium resource calendar, issue the following command:

```
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.

▼ To Move a Resource 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 resource using the "D.15 `csresource`" on page 438 utility. For example to disable the resource with the common name `Auditorium`:**

```
csresource disable Auditorium
```

2 **On the original server, export each of the resources calendars from the calendar database to a file using the "D.10 `csexport`" on page 429 utility. For example:**

```
csexport -c aud100 calendar aud100.ics
```

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

4 **On the new server, for each calendar exported, import the calendar from the file to the calendar database using the "D.11 `csimport`" on page 430 utility. For example:**

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

5 **On the LDAP directory server, update the calendar owner's** *icsDWPHost* **LDAP attribute to point to the new back-end server using the "D.3 csattribute" on page 405 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 resource using the "D.15 `csresource`" on page 438 utility. For example:**

```
csresource 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`.

Note – If you are using the CLD cache option and you have 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 finding a calendar after it has been moved. To clear the CLD cache, follow these steps:

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

15.8 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://CommunicationsExpresshostname:
CommunicationsExpressport/uwc/
   ?calid=calid-1[; ... ;calid-n]
```

For multiple calendars, separate each calendar ID (`calid`) with a semicolon (`;`).

For example, to link to the default calendar for `jsmith@sesta.com`, and `jdoe@siroe.com`, enter:

```
http://calendar.sesta.com:8080/uwc/?calid=jsmith@sesta;jdoe@siroe.com
```

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

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

15.9 Importing and Exporting Calendar Data in Calendar Server 6.3 Databases

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.

15.9.1 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@sesta.com from the file jsmith.ics that was saved in iCalendar (text/calendar MIME) format:

```
csimport -c jsmithcal@sesta.com calendar jsmith.ics
```

To import data into the calendar jsmithcal@sesta.com from a file named jsmith.xml that was saved in XML (text/xml MIME) format:

```
csimport -c jsmithcal@sesta.com calendar jsmith.xml
```

15.9.2 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.

The following examples show how to use the export utility:

- To export the calendar with the calendar ID (calid) jsmithcal@sesta.com in iCalendar (text/calendar MIME) format to a file named jsmith.ics:

  ```
  csexport -c jsmithcal@sesta.com calendar jsmith.ics
  ```
- To export the calendar jsmithcal@sesta.com in XML (text/xml MIME) format to a file named jsmith.xml:

  ```
  csexport -c jsmithcal@sesta.com calendar jsmith.xml
  ```

CHAPTER 16

Administering Calendar Server Databases with the csdb Utility

Calendar Server keeps many database files in multiple directories. You must protect your database files either by implementing the automatic back up process described in Chapter 9, "Configuring Automatic Backups (csstored)," or by implementing your own system of backups. You can administer the database files using the `csdb` utility.

This chapter describes how to manage Calendar Server databases using the `csdb`utility, and includes the following sections:

- "16.1 Using the csdb Utility to Manage Calendar Databases" on page 315
- "16.2 Administering Databases with the csdb Utility" on page 317

16.1 Using the csdb Utility to Manage Calendar Databases

To administer database files, use the Calendar Server utility `csdb`. This section contains topics:

- "16.1.1 Identifying the Three Logical Database Groups" on page 315
- "16.1.2 Targeting Specific Database Groups with the csdb Utility" on page 316

16.1.1 Identifying the Three Logical Database Groups

The calendar database utility `csdb` treats the database files as three logical databases (groups):

- "16.1.1.1 Calendar Database Group (csdb)" on page 315
- "16.1.1.2 Session Database Group (`sessdb`)" on page 316
- "16.1.1.3 Statistical Database Group (`statdb`)" on page 316

16.1.1.1 Calendar Database Group (csdb)

The `caldb` database consists of all the `.db` files and the `_db.*` files found in the database directory. The following is the default location for the calendar database files (as well as the `cld_cache` and `ldap_cache` subdirectories):

```
/var/opt/SUNWics5/csdb
```

If you prefer, you can specify a different directory when running the Calendar Server configuration program (`csconfigurator.sh`). For information about the configuration program, refer to Chapter 2, "Initial Runtime Configuration Program for Calendar Server 6.3 software (csconfigurator.sh)"

The following table describes the various calendar database (`caldb`) files.

TABLE 16–1 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 18, "Administering the Delete Log Database"

16.1.1.2 Session Database Group (`sessdb`)

The session database consists of all files located in the following directories: `/opt/SUNWics5/cal/lib/admin/session/` and `/opt/SUNWics5/cal/lib/http/session/`

16.1.1.3 Statistical Database Group (`statdb`)

The statistical database consists of all files found in the `counter` directory:

```
/opt/SUNWics5/cal/lib/counter/
```

16.1.2 Targeting Specific Database Groups with the csdb Utility

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

`-t caldb` The calendar database group.

`-t sessdb`	The session database group.
`-t statdb`	The statistics database group.

Tip – If you do not include the `-t` option, `csdb` operates on all three databases. Two commands, `check` and `rebuild`, operate only on the calendar database, `caldb`.

16.2 Administering Databases with the csdb Utility

This section describes how to use the “D.8 `csdb`” on page 417 utility to perform the following administrative tasks:

- “To List Status for a Database Group” on page 317
- “To Check for Corruption in the Calendar Database Group” on page 319
- “To Rebuild the Calendar Database Group (caldb), Without the GSE Database” on page 319
- “To Rebuild the Calendar Database Group, Including the GSE Database” on page 321
- “16.2.1 To Delete a Database Group” on page 324

Note – 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 Appendix D, “Calendar Server Command-Line Utilities Reference.”

To List Status for a Database Group

To view the status of a database group (`caldb`, `sessdb`, `statdb`), use the `csdb` utility `list` command.

To list database status:

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. **Change to the `/sbin` directory. For example, on Solaris Operating Systems, enter:**

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

4. **Run the `list` command against one or all of the database groups. For example, to list the status and statistics for all three database groups:**

   ```
   ./csdb list
   ```

The code that follows shows sample output:

```
Sleepycat Software: Berkeley DB 4.1.25: (December 19, 2002)

Calendar database version: 3.0.0 [BerkeleyDB]
Total database size in bytes: 57344

Session database version: 1.0.0 [BerkeleyDB]
Total database size in bytes: 0

Counter database version: 1.0.0 [Memory Mapped Files]
Total database size in bytes: 118792
```

Or, you can choose to use the verbose mode. For example:

```
./csdb -v list
```

The following sample code shows the verbose output:

```
Sleepycat Software: Berkeley DB 4.1.25: (December 19, 2002)

Calendar database version: 3.0.0 [BerkeleyDB]
Total database size in bytes: 57344
Total number of calendars:    2
Total number of events:       0
Total number of tasks:        0
Total number of alarms:       0
Total number of gse entries:  0
Total number of master component entries:  0
Total number of deletelog entries:  0
Total logfile size in bytes:  5779919

Session database version: 1.0.0 [BerkeleyDB]
Total database size in bytes: 0
Total logfile size in bytes:  0

Counter database version: 1.0.0 [Memory Mapped Files]
Total database size in bytes: 118792
```

Or, use the `-t` option to specify one target database group (`caldb`, `sessdb`, or `statdb`). For example, to view database status and statistics for only the calendar database:

```
csdb -t caldb list
```

▼ To Check for Corruption in the Calendar Database Group

Use the `check` command to scan for corruptions in the calendar database, including calendar properties (calprops) and events and todos (tasks). If the `check` command finds an inconsistency that cannot be resolved, it reports the situation in its output.

The `check` command does not check for corruption in the alarm or group scheduling engine (GSE) databases.

To check for database 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.*`) or log (`log.*`) files.**

4. **Change to the *cal-svr-base*`/SUNWics5/cal/sbin` directory. For example, on Solaris Operating 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. **After you run the `check` command, review the output file.**

   If your database is corrupted, you can choose to replace it with your hot backup copy. Alternately, you can choose to try to rebuild the corrupted one by running the `rebuild` command.

▼ To Rebuild the Calendar Database Group (caldb), Without the GSE Database

To recover a damaged calendar database (`caldb`), use the `csdb` utility `rebuild` command. The `rebuild` command scans all of the calendar databases for corruption. If the `rebuild` command finds an inconsistency, it generates a rebuilt calendar database (`.db` files) in the *cal-svr-base*`/SUNWics5/cal/sbin/rebuild_db` directory.

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

In the instructions that follow, the `rebuild` command does not rebuild the group scheduling engine (GSE) database.

To rebuild the calendar databases without the GSE 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 the database (`.db`) files and the log (`log.*`) files. You don't need to copy any share (`__db.*`) files.**

4 **Change to the *cal-svr-base*`/SUNWics5/cal/sbin` directory. For example, on Solaris Operating Systems, enter:**

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

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 you made of your calendar database:**

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

If you don't specify a database directory, the `rebuild` command uses the database

in the current directory. In the preceding example, the */tmp/* parameter specifies the destination directory for the rebuilt database.

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:

a. `./csdb rebuild db_0629`

   ```
   Then check for corruption. If this backup copy is also corrupt, then run
   rebuild on the next backup copy.
   ```

b. `./csdb rebuild db_0615`

   ```
   Then check for corruption. If this backup copy is also corrupt, then run
   rebuild on the next backup copy.
   ```

c. `./csdb rebuild db_0601`

   ```
   ... etc.
   ```

The `rebuild` command writes the rebuilt database to the `cal-svr-base/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, copy the rebuilt database (`.db`) files and the transaction log (`log.*`) file(s) from the `rebuild_db` directory to your production database.**

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

9 **Restart Calendar Server.**

▼ To Rebuild the Calendar Database Group, Including the GSE Database

If you have implemented group scheduling at your site, then you should include the GSE database in the rebuild.

To rebuild both the calendar databases and the GSE database:

1 **Determine if the GSE database has any entries by running the** `csschedule -v list` **command and then let the GSE finish processing the entries.**

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

3 **Stop Calendar Server.**

4 **Make a copy of your calendar database, if you haven't already done so.**

Copy the database (`.db`) files and the log (`log.*`) files. You don't need to copy any share (`__db.*`) files.

5 **Change to the** *cal-svr-base*`/SUNWics5/cal/sbin` **directory.**

For example, on Solaris Operating Systems, enter:

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

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

6 **Run the** `rebuild` **command on the copy of your calendar database:**

```
./csdb -g rebuild /tmp/db /tmp/
```

If you don't specify a database directory, `rebuild` uses the database in the current directory. In the preceding example, the */tmp/* parameter specifies the destination directory for the rebuilt database.

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

---

7 **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
```

8 **After you have verified that `rebuild` was successful, copy the rebuilt database (`.db`) files from the `rebuild_db` directory to your production database.**

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

10 **Restart Calendar Server.**

Example 16–1 Sample Rebuild Output

The sample output shows the events and the todos databases scanned twice each. This is not an error. It scans the first time to verify the information in the `calprops` database and then scans again to make sure `calprops` is accessible from the calendar database.

The following example shows the command and the output that it generated:

```
# ./csdb -g rebuild
Building calprops based on component information.
Please be patient, this may take a while...
Scanning events database...
512 events scanned
Scanning todos database...
34 todos scanned
Scanning events database...
512 events scanned
Scanning todos database...
34 todos scanned
Scanning deletelog database...
15 deletelog entries scanned
Scanning gse database...
21 gse entries scanned
Scanning recurring database...
12 recurring entries scanned
Successful components db scan
Calendar database has been rebuilt
Building components based on calprops information.
Please be patient, this may take a while...
Scanning calprops database to uncover events...
25 calendars scanned
Scanning calprops database to uncover todos...
25 calendars scanned
```

```
Successful calprops db scan
Calendar database has been rebuilt
```

16.2.1 To Delete a Database Group

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.

CHAPTER 17

Backing Up and Restoring Calendar Server Data

If you have chosen not to use the automatic backup facility provided by Calendar Server (using csstored), then you need to implement a backup procedure to protect your data. This chapter describes how to use Calendar Server tools and other Sun tools to perform a manual backup and restore of calendar database files.

To back up and restore Calendar Server data in the /var/opt/SUNWics5/csdb directory, use these command-line utilities:

- The csbackup command backs up the calendar database, a specific calendar, or a user's default calendar. The directories to be backed up must be owned by the runtime user (icsuser), or you will receive an error message when you attempt to restore the data.
- The csrestore command restores the calendar database, individual calendars, or a user's default calendar that was saved using csbackup.

Note – If you have an existing custom script that uses the Berkeley database tools (such as, db_recover), you might find that the tools do not work after upgrading to Calendar Server version 6.3. In earlier versions of Calendar Server software, the tools were compiled with a static library. They are now compiled with a dynamic library.

To accommodate this change, alter your custom script to use the dynamic link library, as follows: Change the global variable LD_LIBRARY_PATH to the name of the dynamic library (libdb-4.2.so).

This chapter includes these sections:

Caution – Calendar Server version 2 data is not compatible with the current product. Do not try to restore calendar data backed up by the Calendar Server version 2 backup utility, because data loss can occur.

If you have version 2 calendar data that you want to move to the current release, you must contact technical support for the appropriate migration utilities.

17.1 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:

- "To Back Up the Calendar Database to a Directory" on page 326
- "To Back Up a Specific Calendar to a File" on page 327
- "To Back Up a User's Default Calendar to a File" on page 327

▼ To Back Up the Calendar Database to a Directory

1. **Log in as the owner of the database files (such as** icsuser**).**

2. **Use the** csbackup **utility** database **command.**

   For example, to back up the calendar database to a directory named backupdir:

   ```
   csbackup -f database backupdir
   ```

3. **Verify the correct version of the database was backed up by checking the** ics50caldb.conf **version file in the backup directory.**

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 nonexistent target backup directory and let csbackup create the directory for you.

▼ To Back Up a Specific Calendar to a File

1 **Login as the database owner (icsuser).**

2 **To backup a calendar to a file in iCalendar or XML format, use the csbackup utility calendar command.**

The filename extension (.ics or .xml) of the backup file indicates the format.

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

```
csbackup -c jsmithcal@sesta.com calendar backupdir/jsmith.ics
```

Or, to backup the calendar jsmithcal@sesta.com in XML format (text/XML) to the file jsmith.xml in the backupdir directory:

```
csbackup -c jsmithcal@sesta.com calendar backupdir/jsmith.xml
```

▼ To Back Up a User's Default Calendar to a File

1 **Login as the database owner (icsuser).**

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

For example, to back up the fault calendar for user jsmith@sesta.com in iCalendar (text/calendar MIME) format to a file named jsmith.ics in the backup directory:

```
csbackup -a jsmith@sesta.com defcal backupdir/jsmith.ics
```

Or, to back up the default calendar for user jsmith@sesta.com in XML (text/xml MIME) format to a file named jsmith.xml in the backup directory:

```
csbackup -a jsmith@sesta.com defcal backupdir/jsmith.xml
```

17.2 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:

- "To Restore the Calendar Database" on page 328

- "To Restore a Calendar From a Backup Directory" on page 328
- "To Restore a Calendar From a File" on page 328
- "To Restore a User's Default Calendar" on page 329

▼ To Restore the Calendar Database

1 Log in as the database owner (`icsuser`).

2 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
```

▼ To Restore a Calendar From a Backup Directory

1 Log in as the database owner (`icsuser`).

2 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@sesta.com` from the backup database directory `backupdir`:

```
csrestore -c jsmithcal@sesta.com calendar backupdir
```

▼ To Restore a Calendar From a File

1 Log in as the database owner (`icsuser`).

2 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 filename extension (`.ics` or `.xml`) of the backup file indicates the format in which the calendar was saved.

For example, to restore the calendar `jsmithcal@sesta.com` that was saved in iCalendar (text/calendar MIME) format to the file `jsmith.ics` located in the `backupdir` directory:

```
csrestore -c jsmithcal@sesta.com calendar backupdir/jsmith.ics
```

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

```
csrestore -c jsmithcal@sesta.com calendar backupdir/jsmith.xml
```

▼ To Restore a User's Default Calendar

1. **Log in as the database owner (`icsuser`).**

2. **To restore a user's default calendar that was saved to a backup file using the `csbackup` utility, use the `csrestore` utility `defcal` command.**

   The filename extension (`.ics` or `.xml`) of the backup file indicates the format in which the calendar was saved.

   For example, to restore the default calendar for user `jsmith@sesta.com` that was saved in iCalendar (`text/calendar` MIME) format to a file named `jsmith.ics` located in the backup directory `backupdir`:

   ```
   csrestore -a jsmith@sesta.com defcal backupdir/jsmith.ics
   ```

   To restore the default calendar for jsmith 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@sesta.com defcal backupdir/jsmith.xml
   ```

17.3 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:

- "17.3.1 StorEdge or Legato Tools" on page 330
- "To Back Up Calendar Data Using Sun StorEdge Enterprise Backup Software or Legato Networker" on page 330
- "To Restore Calendar Data Using Sun StorEdge Enterprise Backup Software or Legato Software" on page 331

17.3.1 StorEdge or Legato Tools

Calendar Server provides the following 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` utility.
`legrestore.sh`	Script that calls the `csrestore` utility.

▼ To Back Up Calendar 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 `-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 save set 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` command 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. 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.

▼ To Restore Calendar Data Using Sun StorEdge Enterprise Backup Software or Legato Software

To restore calendar 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 18

Administering the Delete Log Database

Calendar Server includes the Delete Log database (`ics50deletelog.db`) to store deleted events and todos (tasks).

In early releases, Calendar Server did not maintain a database of deleted events and tasks. User interfaces that store a local copy of the calendar events and tasks had difficulty trying to determine which events had been deleted. Client software was forced to compare the unique identifiers (`uid`) or recurrence identifiers (`rid`) of all events or todos (tasks) with the Calendar Server copy of the data in order to determine which components had been deleted. This limitation directly affected installations that used WCAP commands to develop a client user interface (UI). To solve this limitation, the delete log database was created.

Delete logs need to be managed, as any database file must be. The following sections describe management of deletelog files:

18.1 Creating the Delete Log Database

Calendar Server automatically 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`).

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.

18.2 Querying the Delete Log Database

This section describes how to query 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.

The following information explains when and how to specify each mode.

- Expanded Mode (`recurring = 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* = `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, the `fetch_deletedcomponents` command does not return master components associated with the deleted recurring instances that are still active. To return active master components, use the `fetchcomponents_by_lastmod` WCAP command. The `fetch_deletedcomponents` command should be used in conjunction with the `fetchcomponents_by_lastmod` command.

For more about WCAP commands, see the *Sun Java System Calendar Server 6.3 WCAP Developer's Guide*.

18.3 Purging the Delete Log Database

This section describes how to purge the delete log database. Calendar Server provides two types of purging of the delete log database, automatic and manual.

This section contains the following topics:

18.3.1 Tuning Delete Log Purging

Before purging the delete log database, you need to be very aware of the end users you are serving. If your end users are using Communications Express, the default parameter settings should be adequate. If , however, they are using client user interfaces that store a local copy of the events and tasks, such as the Connector for Microsoft Outlook, or the Sync Tool, then you must adjust the settings of the automatic purge configuration parameters to fit their needs. Typically, they need the delete log to include up to 30 days or more of entries. This will cause the size of the delete log to increase dramatically. A failure to make this adjustment could cause problems with the database. The purge interval should also be adjusted to fit the needs of the users. For instance, there might not be a point in running purge every minute when your Delete log database is holding 30 days of data before it will be allowed to be purged. Things will age out daily, though, so a daily purge would be reasonable.

Similar problems can occur running `cspurge` manually. If too much is removed from the Delete log, it can cause users of the Connector for Microsoft Outlook and the Sync Tool to get out of sync with the server database.

Waiting too long to purge the Delete log database can cause the files to grow very large. Then, when a giant purge occurs, daily transaction logs grow greatly, reflecting the fact that each item purged is a transaction recorded in those logs and then archived to the archive and hot backups. These large anomalies in the transaction logs can make it appear as if there is a system problem and cause time to be lost figuring out what happened.

18.3.2 Automatic Purge of the Delete Log Database

If you wish, you can have Calendar Server automatically purge entries in the Delete Log database at a specified interval. By default the automatic purge is disabled.

The following `ics.conf`parameter controls the automatic purge feature.

TABLE 18–1 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 `518400` seconds (6 days).

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 "18.3.2 Automatic Purge of the Delete Log Database" on page 335 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.

18.3.3 Manual Purge of the Delete Log Database

You can choose to manually purge entries in the Delete Log database (`ics50deletelog.db`), using the `cspurge` utility:

Usage of this utility is as follows:

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

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

To run `cspurge`, 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 "D.13 `cspurge`" on page 434.

18.4 Using Calendar Server Utilities for the Delete Log Database

The following table lists the Calendar Server utilities that support the delete log database (`ics50deletelog.db`).

TABLE 18–2 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 Appendix D, "Calendar Server Command-Line Utilities Reference."

CHAPTER 19

Administering Calendar Server Time Zones

This chapter describes how Calendar Server software defines and processes time zones.

This chapter contains the following sections:

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`

19.1 Overview of Calendar Server Time Zones

This section contains an overview of time zones as implemented by Calendar Server software.

The `timezones.ics` file contains the representation of the time zones supported by Calendar Server. The file is located in the following directory:

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

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 Example 19–1. 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 subcomponents: `STANDARD` for standard 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 Communications 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 Java System Calendar Server 6.3 WCAP Developer's Guide*.

EXAMPLE 19–1 America/Los_Angeles Time-Zone Representation in the `timezones.ics` File

The following example shows the America/Los_Angeles time zone representation in the `timezones.ics` file.

19.2 Managing Calendar Server Time Zones

This section contains conceptual information and instructions on how to manage time zones.

This section contains the following topics:

- "19.2.1 Adding a New Time Zone" on page 340
- "19.2.2 Modifying an Existing Time Zone" on page 341

19.2.1 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 Communications Express user interface. For example, you might want to add a new time zone for America/Miami.

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. If your new time zone has Daylight Savings Time (DST), try to find a similar block to copy.

19.2.2 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:**

   `/etc/optSUNWics5/config/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`/SUNWics5/cal/html/`*language*`/i18n.xsl`

   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`/SUNWics5/cal/html/change_timezone.xml`

   cal-svr-base`/SUNWics5/cal/html/new_cal.xml`

   cal-svr-base`/SUNWics5/cal/html/new_group.xml`

   For information about the entries in these files, see "19.2.1 Adding a New Time Zone" on page 340.

4. **If the change affects the default time zone for user preferences, modify the "*icsTimeZone*" entry in the following file:**

   cal-svr-base`/SUNWics5/cal/html/default_user_prefs.xml`

   Note – Steps 2, 3, and 4 are required only if you use the Calendar Express user interface.

   ---

5. **Stop (if necessary) and then restart Calendar Server for your time zone changes to take effect.**

◆ ◆ ◆ CHAPTER 20

Using Instant Messaging Pop-up Reminders

Calendar Server is integrated with Sun Java System Instant Messaging 6.0 (or later) to provide automatic pop-up reminders for both calendar events and tasks.

This chapter contains conceptual information and instructions on how to configure pop-up reminders.

This chapter contains the following sections:

- "20.1 Pop-up Reminders Overview" on page 343
- "20.2 Configuring Pop-up Reminders" on page 344

20.1 Pop-up Reminders Overview

This section contains the conceptual information you need to understand how pop-up reminders work in Calendar Server software.

This section contains the following topics:

- "20.1.1 Configuration Concepts for Calendar Pop-up Reminders " on page 343
- "20.1.2 How Pop-up Reminders Work" on page 344

20.1.1 Configuration Concepts for Calendar Pop-up Reminders

This section describes what must be configured to make pop-up reminders work.

Users can receive Instant Messenger pop-up reminders for upcoming events and tasks on their calendars.

To enable these pop-up reminders, two things must happen:

- The administrator must configure Calendar Server and Instant Messaging Server to allow pop-up notifications.
- The end user must specify email reminders in the Options tab of Communications Express, which sets an alarm in the Event Notification System.
- The end user must enable calendar reminders in Instant Messenger.

With pop-ups enabled, when an impending event or task nears, the alarm set in the Event Notification System causes Calendar Server to send an email notification and Instant Messaging to display a pop-up reminder.

A Calendar Server administrator can choose to configure either email notifications or pop-up reminders or both for end users. For example, to turn email reminders off, set the following parameter in the `ics.conf` file:

```
caldb.serveralarms.binary.enable= "no"
```

20.1.2 How Pop-up Reminders Work

This section describes how pop-up reminders work.

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 task notification in `text/xml` or `text/calendar` format to ENS.
3. The Instant Messaging JMS subscriber receives the calendar event or task 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. If the recipient is available, Instant Messenger generates an HTML pop-up reminder on the end user's desktop based on the message.

20.2 Configuring Pop-up Reminders

This section contains instructions for configuring pop-up reminders for Calendar Server software.

This section includes the following configuration instructions:

- "To Configure Instant Messaging Server" on page 345
- "To Configure Calendar Server" on page 347
- "To Configure Instant Messenger" on page 347

▼ To Configure Instant Messaging Server

The high level list of tasks necessary to configure Instant Messaging for Pop-ups that follows is for your convenience. To configure Instant Messaging, refer to the Instant Messaging documentation available at:

`http://docs.sun.com/coll/1309.2`

1 **Install the new package `SUNWiimag`.**

Before you can use Instant Messaging for Pop-ups, the Instant Messaging package must be installed using the Java Enterprise System installer.

2 **On the machine where Instant Messaging is installed, change to the following directory:**

```
cd /etc/opt/SUNWiim/default/config
```

3 **Edit one or more of the parameters in the `iim.conf` file as shown in the following table.**

The parameter values shown assume you want pop-up reminders for both events and tasks. If these parameters do not already exist in your `iim.conf` file, add them.

Parameter	Description and Appropriate Value to Use
JMS Consumers Section	
jms.consumers	Name of alarm. Set the value to *cal_reminder*.
jms.consumer.cal_reminder.destination	Destination of the alarm. Set the value to enp:///ics/customalarm
jms.consumer.cal_reminder.provider	The name of the provider. Set to ens. This must be the same as the name in *jms.providers* in the JMS Providers section.
jms.consumer.cal_reminder.type	The type of alarm to set. Set the value to `topic.`
jms.consumer.cal_reminder.param	The alarm parameter. Set the value to `"eventtype=calendar.alarm" (including the quotes)`

jms.consumer.cal_reminder.factory	C++ factory name. Set the value to:
	`com.iplanet.im.server.`
	`JMSCalendarMessageListener`

Parameter	Description and Appropriate Value to Use
JMS Providers Section	
jms.providers	The name of the provider. Set value to `ens`. This must be the same as the value listed in the JMS Consumers Section for `jms.consumer.cal_reminder.provider.`
jms.provider.ens.broker=cal.example.com	Port number that ENS listens on. Set to the port specified in the `ics.conf` file parameter `service.ens.port`. The default is 57997.
jms.provider.ens.factory	C++ factory to use. Set to *com.iplanet.ens.jms.EnsTopicConnFactory*
Calendar Server General Parameters	
iim_agent.enable	Enables the Calendar agent. Set the value as follows including the quotes: `iim_agent.enable="true"`
iim_agent.agent-calendar.enable	Loads a component that enables the Calendar agent. Set the value as follows including the quotes: `iim_agent.agent-calendar.enable="true"`
agent-calendar.jid	The JID of the Calendar agent. Set this value as follows: `agent-calendar.jid=calimbot.`*server*`.`*domain*
agent-calendar.password	The Calendar agent password. Set this value as follows: `agent-calendar.password=`*password*
iim_server.components	Set this value as follows: `iim_server.components=agent-calendar`

4 **Change to the directory where the `imadmin` command-line utility is located:**

```
cd /opt/SUNWiim/sbin
```

5 **Start the Calendar agent using `imadmin`:**

```
imadmin start agent-calendar
```

The Calendar agent is an Instant Messaging component that provides pop-up functionality to Calendar Server users. Using tools provided with Instant Messaging, you can start, stop, restart, or check the status of the Calendar agent as well as monitor its activity through log files.

Note – If you have scripts that include the `stop`, `start`, and `refresh` commands, add the calendar agent to them.

For more information about `imadmin and the Calendar agent`, see the *Sun Java System Instant Messaging 7 2005Q1 Administration Guide*.

▼ To Configure Calendar Server

Before You Begin Confirm that the `ics.conf` parameters shown in the following table have the values shown. If they do not, or you wish to customize them, perform the following steps:

1 **Log in as an administrator with configuration privileges.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4 **Save your old `ics.conf` file by copying and renaming it.**

5 **Edit the `ics.conf` parameters as shown in the following table:**

Parameter	Description and Default Value
caldb.serveralarms	Enables calendar alarms to be queued. The default is “`yes`” (enabled).
caldb.serveralarms.contenttype	Output format for alarm content. The default is "`text/xml`".
caldb.serveralarms.dispatch	Enables calendar alarms to be dispatched. The default is “`yes”.`
caldb.serveralarms.dispatchtype	The type of server alarm to dispatch. The default is "`ens`".
caldb.serveralarms.url	This is the URL for alarm retrieving alarm contents. The default is "`enp:///ics/customalarm`".

6 **Save the file as `ics.conf`.**

7 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

▼ To Configure Instant Messenger

To receive pop-up reminders for Calendar Server events and tasks, end users must configure their Instant Messenger as follows:

1 **On the Main window, from the Tools menu, select Settings.**

2 **On the Settings window, click the Alerts tab.**

3 **Check the Show Calendar Reminders option.**

4 **Click OK.**

CHAPTER 21

Tuning Calendar Server Performance

This chapter contains conceptual information and instructions for tuning Calendar Server performance.

To improve the performance of Calendar Server, consider the following options:

21.1 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

This attribute is used to search for the default calendar for a calendar user or resource. Specify presence (*pres*), equality (*eq*), and substring (*sub*) index types.

icsCalendarOwned

This attribute is used to search for other calendars owned by the user. Specify presence (*pres*), equality (*eq*), and substring (*sub*) index types. See also "21.2 Improving Calendar Search Performance in a DWP Environment" on page 350.

mail, *mailAlternateAddress* These attributes specify a user's primary and alternate email addresses. See also "14.1 Creating Calendar User LDAP Entries" on page 269 and "14.5.4.3 To Add Calendar Service with Calendar Server Utilities" on page 280.

For information about adding directory server indexes, refer to Directory Server documentation found at:

`http://docs.sun.com/coll/1316.1`

21.2 Improving Calendar Search Performance in a DWP Environment

When you are in a DWP environment, that is, the calendar database is distributed across multiple back-end servers, searching for a calendar in the calendar database can be time consuming. It can be faster to look in the LDAP entry first and find out directly which DWP host the calendar resides on.

This section contains the following topics:

- "To Enable Calendar Searches to Look at LDAP" on page 350
- "To Improve Search Performance by Indexing" on page 351

▼ To Enable Calendar Searches to Look at LDAP

To enable calendar searches to look at the LDAP directory first, and the calendar database second, perform the following steps:

1. **Log in as an administrator with configuration privileges.**

2. **Stop Calendar Server services by issuing the** `stop-cal` **command.**

3. **Change to the configuration directory,** `/etc/opt/SUNWics5/cal/config`**.**

4. **Set the** *service.calendarsearch.ldap* **parameter in the** `ics.conf` **file to** `"yes"`**, which is the default, as shown below:**

   service.calendarsearch.ldap="yes"

5. **Restart Calendar Services as follows:**

   ```
   start-cal
   ```

Note – If you are allowing anonymous access to public calendars, you might prefer to disable calendar searches from looking at LDAP. In fact, Communications Express expects the parameter value to be "no".

▼ To Improve Search Performance by Indexing

1 **To determine if the calendar search performance can be improved by indexing, 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 a search dialog in .

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.

2 **Index appropriate LDAP attributes, or at least,** *icsCalendarOwned*, **by running** *comm_dssetup.pl.*

The `comm_dssetup.pl`, indexes this attribute and many others to improve performance in various ways. If you have not run `comm_dssetup.pl`, or ran it but did not perform the indexing, you can run the utility again to do the indexing, or you can use Directory Server tools to perform the indexing.

For information on how `comm_dssetup.pl` does indexing, see *Attribute Indexes* in the *Sun Java System Communications Suite 5 Installation and Configuration Guide*.

For information about adding directory server indexes, refer to Directory Server documentation found at:

http://docs.sun.com/coll/1316.1

21.3 Improving Performance of Calendar Searching by Disabling Wildcard Searches

By default, wildcard searches are disabled in Calendar Server. That is, when you search for a calendar using the graphical user interface, or when you issue `search_calprops.wcap` in your custom interface, it searches for an exact match to the search string passed in with the WCAP command.

If you have enabled wildcard searches by uncommenting the following line in the `ics.conf` file (by removing the exclamation point ("!") at the beginning), you may be experiencing a negative impact on performance.

```
!service.calendarsearch.ldap.primaryownersearchfilter =
"(&(|(uid=*%s*)(cn=*%s*))(objectclass=icsCalendarUser))"
```

To test the impact of wildcard searches on performance, comment out the line again by inserting the exclamation point ("!") in front of it.

21.4 Improving Performance of the CLD Plug-in

Before the system accesses a calendar from the calendar database, it must determine which back-end machine stores that user's calendars. To find the appropriate back-end machine, the system searches the LDAP directory for the user's entry and picks up the `icsDWPHost` attribute. This search is time consuming, and it must be performed for every access to the calendar data. Every user session can result in many accesses of the database and thus many searches of the LDAP. To save time and enhance performance, enable the CLD cache by editing the `ics.conf` file as follows:

```
caldb.cld.cache.enable="yes"
```

The LDAP data cache stores the user ID and its associated icsDWPHost attribute. Before searching the LDAP for a user's entry, the system checks the cache for the user's ID. If it is in the cache, it picks up the back-end host name from the icsDWPHost attribute stored there. If it is not in the cache, the system performs the LDAP search and copies the user ID and attribute into the CLD cache. Subsequently, accesses to the user's calendar data will be faster, since it will now find the user ID in the cache.

21.5 Improving Performance of the LDAP Data Cache

With the LDAP data cache enabled, you can tune it using the `ics.conf` parameters, adjust one or more of the parameters found in the following table.

Note – The LDAP data cache is enabled by default. You can disable it by setting: `local.ldap.cache.enable="no"`

TABLE 21–1 `ics.conf` Parameters Used to Customize LDAP Data Caching

Parameter	Description/Value
local.ldap.cache *.checkpointinterval*	The number of seconds for the checkpoint thread to sleep between checkpoints. The default is `“60”`. In a high activity LDAP, you might want to decrease the interval to keep the cache as current as possible. At the same time, remember that the more often you refresh the cache, the more system overhead you introduce.
local.ldap.cache. *circularlogging*	Specifies whether to remove the LDAP data cache database log files after they have been processed. The default is `“yes”`. Do not change this parameter unless you have a custom clean up routine that will remove the old log files.
local.ldap.cache. *logfilesizemb*	Specifies the maximum size in megabytes of checkpoint file. The default is `"10”` megabytes. If you have a high activity LDAP, this file could fill up before the checkpoint interval is over. Try to set the value to a number that is close to the actual size of the logs according to your experience
local.ldap.cache. *maxthreads*	Specifies the maximum number of threads for the LDAP data cache database. The default is `“1000”`. In a high activity LDAP, you might want to increase the number of threads. This could cause increased CPU utilization. Decrease the number of threads only if your LDAP activity is minimal.
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). If your cache is filling up too fast (high activity), you can decrease the TTL time. However, this could increase the overall number of LDAP database accesses, which could slow the system down overall.
local.ldap.cache. *cleanup.interval*	Specifies the interval in seconds between each cache database cleanup. The default is `“1800”` seconds (30 minutes). The system removes expired entries. The time interval does not have to be the same as the entry TTL time. But synchronizing them can make it more efficient.
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”`. For performance enhancement, use this only in debug mode.

TABLE 21–1 `ics.conf` Parameters Used to Customize LDAP Data Caching *(Continued)*

Parameter	Description/Value
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). This is only active if *local.ldap.cache.stat.enable* is enabled. Decreasing the interval can help you pinpoint problems. Increasing the interval helps decrease system load.

Note – Communications Express expects data caching to be disabled.

21.6 Tuning the LDAP SDK Cache

There are a couple of parameters that control how long an item stays in the cache, and how large the cache can be.

To tune the cache, edit one or more of the parameters as shown in the following table.

TABLE 21–2 `ics.conf` Parameters for Configuring the LDAP SDK Cache

Parameter	Description and Default Value
service.ldapmemcachettl	This is not currently implemented. You must manually remove the contents of the `ldap_cache` directory and then restart Calendar Server. If *service.ldapmemcache* is "`yes`", this parameter is used to set 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. The default is “`30`”.
service.ldapmemcachesize	If *service.ldapmemcache* is "`yes`", this parameter is used to set the maximum amount of memory in bytes that the cache will consume. If “`0`”, the cache has no size limit. The default is “`131072`”.

21.7 Tuning Automatic Backups

You must balance the number of backups you keep on disk with the need to not exceed available disk space. To help manage the amount of disk space your archival and hot backups take, you can change the settings of various `ics.conf` parameters that determine how many copies of the backups you keep at one time and where the disk space threshold is that will trigger clean up of the older copies.

There are three types of parameters that can be adjusted for the each backup type, archival and hot backup:

- `mindays` – The minimum number of days worth of backups held on disk.

- `maxdays` – The maximum number of days worth of backups held on disk.
- `threshold` – The percentage of disk space used. This is used as a trigger point.

Calendar Server keeps backups for the maximum number of days possible without going over the threshold on disk space. So if the current backup is going to push the disk usage above the threshold, the system will purge the oldest backup copy and see if disk space usage goes below the threshold. It will continue to purge old backup copies until either of the following conditions is met: removing another backup copy would bring the number of backups on disk below the minimum number of backup copies, or the disk space usage falls below the threshold.

Therefore, you can manage the amount of disk space backups use with the threshold parameter. And conversely, you can manage how many backups you keep on disk by adjusting the amount of disk space and copies allowed.

The following table lists the `ics.conf` parameters that control the disk space and number of backups kept on disk.

TABLE 21–3 `ics.conf` Parameters Used to Set Number of Backups Held on Disk

`ics.conf` **Parameters**	**Default Setting**	**Description**
caldb.berkeleydb.hotbackup.mindays	3	Minimum number of days of hot backups held on disk.
caldb.berkeleydb.hotbackup.maxdays	6	Maximum number of days of hot backups held on disk.
caldb.berkeleydb.hotbackup.threshold	70	Percent of disk space used for hot backups. Triggers purge of oldest copies when exceeded.
caldb.berkeleydb.archive.mindays	3	Minimum number of days of archival backups held on disk.
caldb.berkeleydb.archive.maxdays	6	Maximum number of days of archival backups held on disk.
caldb.berkeleydb.archive.threshold	70	Percent of disk space used for archival backups. Triggers purge of oldest copies when exceeded.

21.8 Using Load Balancing Across Multiple CPU's

By default, load balancing is enabled in Calendar Server. Calendar Server achieves load balancing using the following algorithm: Processes accept one connection out of every `n` connections, where `n` is the number of processes.

To disable load balancing, add the *service.http.loadbalancing* parameter to the `ics.conf` file and set it to "`no`". Then restart Calendar Server for the change to take effect.

21.9 Controlling the Number of Processes Running for Each Service

If a server has multiple CPU's, by default Calendar Server distributes the HTTP Service (`cshttpd` processes) and Distributed Database Service (`csdwpd` processes) across the CPU's.

If you want to control the number of processes that run for each service, you can edit the *service.http.numprocesses* and *service.dwp.numprocesses* parameters. By default, these parameters are set to the number of CPU's for the server during installation, but you can reset these values. For example, if a server has 8 CPU's, but you want a `cshttpd` and `csdwpd` process to run in only 4 CPU's, set the parameters as:

```
service.http.numprocesses="4"
 service.dwp.numprocesses="4"
```

21.10 Using Timeout Values

This section contains conceptual information and instructions for tuning Calendar Server performance using timeout values for various `ics.conf` parameters.

The following types of timeouts exist:

- `Timeout Values for csadmind`
- "21.10.2 HTTP Timeout Values for End Users" on page 357
- "21.10.3 GSE Queue Timeout Value" on page 357

For information about editing `ics.conf` parameters, see "E.1 Editing the `ics.conf` Configuration File" on page 455.

21.10.1 Timeout Values for `csadmind`

The following table describes the Calendar Server timeout parameters in the `ics.conf` file used by the Administration (`csadmin`) service.

TABLE 21–4 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).

TABLE 21–4 HTTP Timeout Values for the Administration Service (`csadmin`) *(Continued)*

Parameter	Description
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).

21.10.2 HTTP Timeout Values for End Users

The following table describes the Calendar Server HTTP timeout parameters in the `ics.conf` file that apply to end users.

TABLE 21–5 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).

21.10.3 GSE Queue Timeout Value

The following `ics.conf` file parameter 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 scans the queue again. Therefore, this setting only takes effect 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. However, if the queue is getting too large because of an increased volume of events, the time can be decreased to allow the queue to be processed faster. This may serve to slow down overall performance, but events will be updated sooner.

CHAPTER 22

Troubleshooting Calendar Server 6.3 Software

This chapter covers how to set up logging and what to do for some common problems.

The chapter contains the following topics:

22.1 Turning on Debugging Information for Calendar Server 6.3 Software

This section contains conceptual information and instructions on using logs and debugging to troubleshoot problems with your Calendar Server deployment.

While there is no one `ics.conf` parameter that puts the whole system in "debug mode", this section describes some ways to get useful debug information:

Note – Be sure to turn off excess logging and monitoring when not needed as it will negatively impact performance.

22.1.1 Increase Logging Level

Use the parameter shown in the following table to increase the verbosity of logging:

Parameter	Description and Default Value
logfile.loglevel	Set to `DEBUG` to get all levels logged, including `CRITICAL`, `ALERT`, `ERROR`, `WARNING`, `NOTICE`, and `INFORMATION`. This applies to all logs.

22.1.2 Enable Logging Access to the LDAP Cache

To log all accesses of the LDAP data cache and print out the log (report) set the `ics.conf` parameters shown in the following table:

Parameter	Description and Default Value
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`" (no statistics logged). Set to "`yes`" to enable logging of statistics. For performance enhancement, use this only in 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). This is only active if logging is enabled. Decreasing the interval can help you pinpoint problems. Increasing the interval helps decrease system load.

22.1.3 Clearing the LDAP Cache

There is currently no logic in Calendar Server to expire LDAP cache data. You must manually remove the contents of the `ldap_cache` directory and restart Calendar Server.

▼ To Clear the LDAP Cache

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

22.1.4 WCAP Command and HTTP Access Logging

Two configuration parameters that facilitate debugging enable logging of incoming commands and HTTP accesses. Add one or both of these parameters to the `ics.conf` file to activate the logging:

- `service.http.commandlog = "yes"` — The `cshttpd` process creates a file, `http.commands`, in the logs directory. The log contains each `.shtml` or `.wcap` command received by the server, including all parameters of each command.
- `service.http.commandlog.all = "yes"` — The `cshttpd` process creates a file, `http.access`, in the logs directory. The log contains each HTTP request received by the system.

Caution – The log files can grow very quickly and fill the available disk space. Monitor them carefully to avoid problems. Choose a period of low activity on your systems to run with these commands enabled. Performance will drop noticeably if run during peak hours. Always disable the two commands when you are finished troubleshooting.

22.1.5 Monitor the System Using the Calendar Server 6.3 csstats Utility

Use the `csstats list` command to display statistical information from counter objects defined in the `counter.conf` file.

For more information on the csstats utility, see Appendix D, "Calendar Server Command-Line Utilities Reference."

22.2 Troubleshooting LDAP Issues

This section contains conceptual information on troubleshooting LDAP issues.

If you are creating a multiple domain environment for the first time, you must create the DC tree in LDAP by adding the appropriate entries for domains, containers, users, groups and resources. If the DC tree does not already exist when using a Calendar Server utility, such as `cscal`, you might see the following error message: `"Initialization failed .... exiting"`.

Be sure that your DC tree contains at least one (default) domain under the DC tree root. Create the DC tree structure using instructions found in "13.2 Creating New Calendar Server Domains" on page 264.

22.3 Troubleshooting Migration Utilities

Calendar Server offers several utilities for migrating calendar databases and LDAP directories.

This section contains the following topics:

22.3.1 What to do Before Calling Technical Support

In general, if you have trouble using the migration utilities, you should contact technical support.

Before calling, gather the following information:

- Back-up copies of the databases in question.
- Copies of all the pertinent logs.
- Any error output messages, including cores.

22.3.2 Where to Find the Migration Utilities

The various migration utilities and their documentation can be found at the locations indicated in the list that follows:

Schema Migration Utility (`commdirmig`

This utility is bundled with Delegated Administrator, which is a separately installable component. It migrates your LDAP directory from Schema version 1 to Schema version 2. For information about this utility, see the*Sun Java Communications Suite 5 Schema Migration Guide*.

Calendar Server 6.2 to 6.3 migration utility`csmigrate`

This utility can be found in the `sbin` directory after the software is installed.

Calendar Server 5 to Calendar Server 6.2 migration utility (`cs5migrate`

This utility can be found in the `sbin` directory after the software is installed.

Calendar Server multiple domain database preparation utility (`csmig`)

This utility can be found in the `sbin` directory after the software is installed.

Documentation for this utility can be found in Chapter 3, "Database Migration Utilities for Calendar Server 6.3," which includes a troubleshooting section.

Calendar Server non-domain to multiple domain migration utility (`csvdmig`)

This utility can be found in the `sbin` directory after the software is installed.

Documentation for this utility can be found in Chapter 3, "Database Migration Utilities for Calendar Server 6.3." Use this utility to prepares your calendar database and LDAP directory entries for multiple domains.

Note – Always run `csmig` before `csvdmig`.

Calendar Server 2 to Calendar Server 6 Migration Utility (`ics2migrate`)
This utility is installed with Calendar Server. Documentation can be found in Chapter 3, "Database Migration Utilities for Calendar Server 6.3." Use this utility to migrate your Calendar Server 2 databases to be compatible with Calendar Server 5.

Netscape Calendar Server 4 to Calendar Server 5 Migration Utility (`ncs4migrate`)
This utility is available only from technical support. The utility package includes documentation. This utility migrates Netscape Calendar Server 4 to Calendar Server 5. These migrations tend to require special attention because of the lack of uniformity in the source database. It is not unusual for a lot of manual This utility is available only from technical support. The utility package includes documentation. This utility migrates Netscape Calendar Server 4 to Calendar Server 5. These migrations tend to require special attention. It is not unusual for a lot of work on the source file to be necessary before the utility can be run. You might consider using Professional Services to help you plan your migration.

22.4 Non-Database Troubleshooting for Calendar Server

This section covers various troubleshooting ideas for non-database problems.

The following topics are covered in this section:

- "22.4.1 One cshttpd Process is Accepting Too Many Connections and Taking 100% of CPU Time" on page 364
- "Fixing start-cal Problems" on page 364
- "22.4.2 Fixing stop-cal Problems" on page 364
- "22.4.3 Can't Connect to Back-End Server" on page 366
- "22.4.4 Can't Find Calendar" on page 366
- "22.4.5 Can't Create Calendar on Back-End Machine" on page 367
- "22.4.6 Get "Unauthorized" When Trying to Log In Using Proxy Authentication" on page 367
- "22.4.7 Troubleshooting Searches that Don't Complete Properly" on page 368

Tip – In addition, there is a trouble shooting section for SSL in the SSL chapter:

"7.2 Troubleshooting SSL for Calendar Server 6.3 Software" on page 215

22.4.1 One cshttpd Process is Accepting Too Many Connections and Taking 100% of CPU Time

If one `cshttpd` process is accepting too many connections and taking 100% of CPU time, you might have disabled load balancing. To re-enable it, change the value of `ics.conf` parameter *service.http.loadbalancing* to `"yes"`.

▼ Fixing start-cal Problems

If not all of the calendar services started when you issued `start-cal`, the ones that did start must be stopped before restarting. For example, if `enpd`, `csnotifyd`, and `csadmind` started, but not `cshttpd`, then `enpd,`, `csnotifyd`, and `csadmind` must be stopped.

To start calendar services:

1. **Log in as an administrator with configuration privileges.**

2. **Issue the `stop-cal` command.**

3. **If the `stop-cal` command fails to stop all Calendar Server services, there might be some child processes still running. To handle this, see "22.4.2 Fixing stop-cal Problems" on page 364.**

4. **Once you are sure all Calendar Server processes are stopped, use the `start-cal` command to start all services. For example:**

   cal-svr-base`/SUNWics5/cal/sbin/start-cal`

22.4.2 Fixing stop-cal Problems

This section contains some conceptual information and instructions for fixing `stop-cal` problems.

There are two separate issues to consider when Calendar Server shuts down:

- "To Stop Child Processes" on page 364
- "To Recover After an Improper Shutdown" on page 365

▼ To Stop Child Processes

After issuing `stop-cal`, it is possible that some child processes were not stopped. 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 individually, using the following procedure:

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** `kill -15` **command to kill the process. For example:** `kill -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 kill -9 command to kill it.
For example: kill -9 9875
```

Note – On Linux systems with Calendar Server running, if you search for calendar processes using the `ps` command, the results might appear confusing. In Linux, the `ps` command returns the list of threads running rather than the list of processes. There is no known workaround to display only the processes.

▼ To Recover After an Improper Shutdown

If Calendar Server was not properly shutdown, perform the following steps:

1 **Perform the steps in the previous procedure, "22.4.2 Fixing stop-cal Problems" on page 364.**

2 **Manually delete all files in the LDAP data cache database directory.**

These left over files could cause database corruptions. To delete the files:

a. **Change to the LDAP data cache directory.**

The default is `/opt/SUNWics5/csdb/ldap_cache`, but use the directory pointed to by the *local.ldap.cache.homedir.path* parameter in the `ics.conf` file.

b. **Remove all files in the directory.**

For example: `rm *.*`

c. **Check to make sure all files were removed.**

For example: `ls`

3 **Restart Calendar Server.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

For instructions on how to configure LDAP data caching, see "4.8 Configuring LDAP for Calendar Server Version 6.3" on page 144. For more information about the LDAP data cache, see the*Sun Java Communications Suite 5 Deployment Planning Guide*.

22.4.3 Can't Connect to Back-End Server

1. Ping the back-end server to see if it is responding.

   If it is not responding, determine why it is failing. When it is functioning again, proceed to the next step in this task.
2. Clear the CLD cache. See "12.5 Clearing the CLD Cache in Calendar Server Version 6.3" on page 258.

   If you are using the CLD cache option and you have updated a server name for an `ics.conf` parameter, 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.
3. Stop the server with the `stop-cal` command.
4. Restart Calendar Server using `start-cal`.

22.4.4 Can't Find Calendar

If you are using the CLD cache option and you have moved one or more calendars to different back-end servers (or changed the name of the back-end server), might not be able to see the calendars on the new server.

If this happens, perform the following steps:

1. Clear the CLD cache. See "12.5 Clearing the CLD Cache in Calendar Server Version 6.3" on page 258.

   The CLD cache will be out of date if you moved one or more calendars to different back-end servers. To refresh it, you need to clear the cache so it will be rebuilt.
2. If that fails, confirm that you followed the correct procedure for moving calendar. This information can be found at:

   "15.6 Managing User Calendars" on page 304.

   Then, clear the cache.

22.4.5 Can't Create Calendar on Back-End Machine

If you try to create a calendar on a designated back-end machine, and you get the following error message: `Invalid DWP Host Server`, it means one of two things. Either your server is not configured properly, or the calendar owner has already been assigned to a different back-end server.

This section contains information on how to fix these two problems:

- "22.4.5.1 Back-End Machine Not Configured Properly" on page 367
- "22.4.5.2 Calendar Owner Assigned to a Different Back-End Machine" on page 367

22.4.5.1 Back-End Machine Not Configured Properly

Look at the `ics.conf` file for the back-end server in question.

Verify that the following settings exist:

```
service.dwp.enable = "yes"
caldb.cld.type = "directory"
local.hostname = "back-end hostname"
```

22.4.5.2 Calendar Owner Assigned to a Different Back-End Machine

Look at the user's LDAP entry and see if there is an *icsDWPHost* attribute present. The value of *icsDWPHost* must match the back—end server name on which you are attempting to create the calendar. You can not create a calendar for this user on a different back-end server.

22.4.6 Get "Unauthorized" When Trying to Log In Using Proxy Authentication

This section contains suggestions for possible reasons for failure. Follow the suggested steps and retry the login.

1. Perform one or more of the following steps to fix this error:
   - Verify that *service.http.allowadminproxy* is set to "`yes`".
   - Verify that the *admin-user* has Calendar Server administrator privileges.
   - Verify that the *admin-password* is correct.
   - Verify that the *calendar-user* is a valid Calendar Server user.
2. Retry the log in.

22.4.7 Troubleshooting Searches that Don't Complete Properly

This section contains conceptual information and instructions for troubleshooting searches that don't complete properly.

The *nsslapd-sizelimit* and *nsLookthroughLimit* attributes in your LDAP Directory Server configuration must be large enough so that searches complete properly. If *nsSizeLimit* is not large enough, truncation can occur and no results will be displayed. If *nsLookthroughLimit* is not large enough the search may not complete.

This section covers the following topics:

- "To Determine if Limit Attributes Have Appropriate Values" on page 368
- "To Set the Limit Attributes to Appropriate Values" on page 368

▼ To Determine if Limit Attributes Have Appropriate Values

1 To determine if these attributes 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 search dialog in the user interface.

2 If the LDAP server returns an error, the *nsSizeLimit* or the *nsLookthroughLimit* parameters might not be large enough.

▼ To Set the Limit Attributes to Appropriate Values

The DN for these attributes is:

```
dn: cn=config,cn=ldbm databases,cn=plug ins,cn=config
```

1 Use `ldapmodify` to dynamically set the value of *nsLookthroughLimit*.

You do not have to stop and restart Directory Server to change this attribute.

The default value is `5000`. You might want to increase this value if searches are not reporting results. However, this could slow down the LDAP server.

It is possible to set the limit to `-1`, which causes no limit to be used. However, do this with caution as it could conceivably cause the system to hang.

2 If you want to set *nsslapd-sizelimit* to a higher value, you must perform the following steps:

a. Stop the Directory Server.

b. **Edit the `dse.ldif` file.**

c. **Restart the Directory Server.**

Note – For information on how to use `ldapmodify` and edit the `dse.ldif` file, see Directory Server documentation found at:

`http://docs.sun.com/coll/1316.1`

22.5 Dealing with Calendar Server Database Issues

This section covers various issues involving the Calendar Server (Berkeley Database) databases:

This section contains the following topics:

- "22.5.1 Finding Berkeley Database Tools" on page 369
- "Detecting and Fixing Database Deadlocks" on page 370
- "22.5.2 Detecting Database Corruption" on page 371
- "22.5.3 Dealing with Transaction Log Files Suddenly Very Large and Numerous" on page 372
- "22.5.4 Preventing Service Interruptions When Your Database is Corrupted (Read-only Mode)" on page 373
- "22.5.5 Handling Common Database Failures" on page 374
- "22.5.6 Rebuilding a Corrupted Calendar Database" on page 376
- "22.5.7 Using the Dump and Load Procedure to Recover a Calendar Database" on page 379
- "22.5.8 Restoring an Automatic Backup Copy" on page 381
- "22.5.9 Repairing Custom Backup Scripts" on page 385

22.5.1 Finding Berkeley Database Tools

Many of the troubleshooting steps you will want to take require having access to the Berkeley database utility programs. While a version of these utility programs is available in the Calendar Server bundle, they are not supported. You might want to obtain more information directly from Sleepycat Software (`http://www.oracle.com/database/berkeley-db/index.html`.

This section covers the following topics:

- "22.5.1.1 To Access the Berkeley Database Utilities" on page 369
- "22.5.1.2 List of Available Tools" on page 370

22.5.1.1 To Access the Berkeley Database Utilities

Set and export the `LD_LIBRARY_PATH` environment variable to reflect the following directory:

cal-svr-base`/SUNWics5/cal/tools/unsupported/bin/`

22.5.1.2 List of Available Tools

The following table lists some of the commonly used Berkeley database tools (utility programs).

Berkeley Database Tools	Description
`db_archive`	Writes the path names of log files that are no longer in use to the standard output, one pathname per line.
`db_checkpoint`	A daemon process that monitors the database log and periodically calls the checkpoint routine to checkpoint it.
`db_deadlock`	Traverses the database environment lock region and aborts a lock request each time it detects a deadlock or a lock request that has timed out.
`db_dump`	Writes the specified file to standard output in a flat-text format understood by the `db_load` utility.
`db_load`	Reads from the standard input and loads it into the database file specified. If the file does not already exist it creates it.
`db_printlog`	Debugging utility that dumps log files in human-readable format.
`db_recover`	Restores the database to a consistent state after an unexpected application, database, or system failure.
`db_stat`	Displays statistics for the database environment.
`db_verify`	Verifies the structure of one or more files and the databases they contain.

▼ Detecting and Fixing Database Deadlocks

If the Berkeley database is in a deadlock state, you must reset the database. It is important to detect this condition as early as possible.

To enable the system to periodically check the databases to detect a deadlock state and inform the Administrator:

1. **Log in as an administrator with permission to change the configuration.**

2. **Stop Calendar Server services by issuing the `stop-cal` command.**

3. **Change to the `/etc/opt/SUNWics5/cal/config` directory.**

4. **Save your old `ics.conf` file by copying and renaming it.**

5. **Edit the `ics.conf`, if necessary, to have the following value:**

```
local.caldb.deadlock.autodetect=”yes”
```

Note – When this parameter is set to "yes", the db_deadlock daemon is launched that will monitor the lock region.

22.5.2 Detecting 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:

- "22.5.2.1 Database Corruption Basics" on page 371
- "22.5.2.2 Monitoring Log Files" on page 371
- "To Check for Calendar Database Corruption" on page 372

22.5.2.1 Database Corruption Basics

No one can guarantee corruption free databases. But you can minimize data loss and operational downtime. Closely monitoring the database and calendar server is key to detecting corruption early. Frequent and complete backups are the key to recovering from corruption once it is found.

There are two levels of corruption possible in a calendar database:

- Application level–Offending entries in one of more database files prevent the server from running when they are operated upon.
- Database level–Corruptions in the Berkeley database pages cause various problems. One common symptom is looping while running csdb check. Another common symptom is an error message like the following:

  ```
  "illegal page type or format",
  or "page 97895 doesn't exist, create flag not set."
  ```

22.5.2.2 Monitoring Log Files

Monitor the Calendar Server log files, including the alarm logs, for any error messages that might indicate database corruption.

You should inspect the log files on a regular basis for 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.

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.

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

▼ To Check for Calendar Database Corruption

Use the `check` command to scan for corruptions in the calendar database, including calendar properties (`calprops`) and events and todos (tasks). If the `check` command finds an inconsistency that cannot be resolved, it reports the situation in its output.

The `check` command does not check for corruption in the alarm or group scheduling engine (GSE) databases.

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.*`) or log (`log.*`) files.

4. **Change to the *cal-svr-base*`/SUNWics5/cal/sbin` directory.**

   For example, on Solaris Operating Systems for the default directory, enter:

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

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

   ```
   ./csdb check dbdir /tmp/check.out
   ```

   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.**

   (See "22.5.6 Rebuilding a Corrupted Calendar Database" on page 376.)

22.5.3 Dealing with Transaction Log Files Suddenly Very Large and Numerous

It is possible that your automatic purge configuration settings do not properly account for the client user interface your end users prefer. The sudden appearance of large numbers of large transaction log files can be simply the result of a long delay in purging Delete log records. If this delay is purposefully done to accommodate users of the Connector for Microsoft Outlook or the Sync Tool, then the appearance of the large and numerous transaction log files is to be expected. No further action is required. Eventually the system will catch up. However, if your end users are using the Communications Express client, returning the automatic purge settings to their defaults should fix the problem.

22.5.4 Preventing Service Interruptions When Your Database is Corrupted (Read-only Mode)

This sections covers how to keep your corrupted database accessible while you are in recovery mode and includes the following topics:

- "22.5.4.1 Using Read-only Mode" on page 373
- "To Put a Database in Read-only Mode" on page 373

22.5.4.1 Using Read-only Mode

If you are encountering database corruption, one way to prevent service interruptions is to put your database in read-only mode. This mode allows end users to read database entries, but does not allow additions, modifications, or deletions. If an end user attempts to add, modify or delete any calendar data, the system gives an error message. In addition, administrator tools that add, modify or delete calendar events and todos will not work while the database is in read-only mode.

Note – If the database is corrupted to the point that it can't be read, you must interrupt service long enough to restore a backup. The quickest way to restore a backup is to have a good hot backup. See "22.5.8.1 Before You Restore" on page 381.

▼ To Put a Database in Read-only Mode

1 **Log in as an administrator with permission to change the configuration.**

2 **Stop Calendar Server services by issuing the `stop-cal` command.**

3 **At a command line, change to the directory where the `ics.conf` is located:**

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

4 **Specify read-only mode for the calendar, by setting the parameter as follows:**

```
caldb.berkeleydb.readonly=”yes”
```

5 **Restart Calendar Server by issuing the `start-cal` command.**

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

You must restart the services in order for the ics.conf changes to take effect.

22.5.5 Handling Common Database Failures

This section covers a few of the common database failures and includes some suggested remedies. It contains the following topics:

- "csadmind Won't Start or Crashes During Startup" on page 374
- "Services Hung, and End Users Can't Connect–Orphaned Locks" on page 376
- "csdb rebuild Never Finishes–Database Looping" on page 376

▼ csadmind Won't Start or Crashes During Startup

Since csadmind is the service that handles both the group scheduling engine (GSE) and the alarm dispatch engine, this could have been caused by offending entries in the GSE queue or the alarm queue.

Remedies:

1. **If csadmind is not running, issue the stop-cal command immediately.**

   Leaving calendar server running could cause transaction logs to accumulate, which could further corrupt the database, and could take much longer to reconcile the transaction log files to the database.

2. **Verify that all Calendar Server processes are stopped.**

   For instructions on how to verify that all processes are stopped, see "To Stop Child Processes" on page 364.

3. **Try restarting csadmind again by issuing the start-cal -csadmind command.**

   If it starts successfully, make sure the two queues are functioning by performing the following steps:

   a. **Checking the GSE queue using csschedule.**

   b. **Checking the alarm queue using dbrig.**

   For instructions on running csschedule and dbrig, see Appendix D, "Calendar Server Command-Line Utilities Reference."

4. **If csadmind crashes with a dump, analyze the pstack.**

   If you notice any GSE related functions in the trace (they will have the letters GSE in them), look at the first entry in the GSE queue and the referenced entry in the events database. Most of the time, the event referred to in the GSE entry is the offending entry. To fix this problem:

   a. **Remove the GSE entry using csschedule.**

b. **Remove the offending event from the database using `cscomponents`.**

For instructions on running `csschedule` and `cscomponents`, see Appendix D, "Calendar Server Command-Line Utilities Reference."

5 **If the entries are not corrupted, then it could be a special case that the calendar server could not handle.**

Take the following steps:

a. **Take a calendar environment snapshot of the corrupted database, and contact customer support.**

To create an environmental backup:

i. **Use the `db_checkpoint` utility found at:**

```
cal-svr-base/SUNWics5/cal/tools/unsupported/bin/db_checkpoint
```

ii. **Run `db_archive -s`.**

Use the `-s` option to identify all the database files and copy them to a removable medium, such as CD, or DVD, or tape.

iii. **Run `db_archive -l`.**

Use the `-l` option to identify all the log files and copy unapplied log files to a removable-medium device.

b. **To avoid service interruptions, place your calendar database into a read-only state temporarily, and revert to a hot backup copy.**

- Placing your calendar database into a read-only state temporarily prevents any add, modify or delete transactions from taking place. End users will get an error message when they try to add, modify or delete any calendar data. Administrator tools that add, modify or delete calendar events and todos also will not work while the database is in read-only mode.

  To put your calendar database in read-only mode, edit the `ics.conf` file and set the following parameter to `"yes"`, as shown:

  ```
  caldb.berkeleydb.readonly="yes"
  ```

- Revert to a hot backup copy, using the instructions found in "22.5.8 Restoring an Automatic Backup Copy" on page 381.

  With `csstored` configured and enabled, a hot backup is available that should be within minutes of being uptodate. You should always verify your hot backup copy to make sure it is not corrupt also. (Run `db_verify`.)

6 **If all else fails, perform the dump and reload procedure to see if it can salvage the database.**

This procedure is described in "22.5.7 Using the Dump and Load Procedure to Recover a Calendar Database" on page 379.

▼ Services Hung, and End Users Can't Connect–Orphaned Locks

This condition may be caused by a control thread, which holds a Berkeley DB database page lock, quitting without releasing the lock. To confirm the problem, run `pstack` on `cshttpd` processes and `csadmind`. (`pstack` is a standard UNIX utility found at: `/usr/bin/pstack`) It should show threads that are waiting to acquire a lock.

To fix the problem, restart Calendar Server, as follows:

1 **Change to the directory where `start-cal` resides.**

```
cd cal-svr-base/SUNWics5/cal/sbin
```

2 **Issue the `start-cal` command.**

```
./start-cal
```

▼ `csdb` rebuild Never Finishes–Database Looping

Database looping is usually caused by corruption in the database files. Since it is a database corruption, it can be unrecoverable. There are several options:

1 **Revert to the hot backup.**

If the corruption occurred recently, you can use one of your hot backups.

2 **Use your catastrophe archival recovery process.**

For a suggested process, see "22.5.8 Restoring an Automatic Backup Copy" on page 381.

3 **Use the dump and reload procedure, "22.5.7 Using the Dump and Load Procedure to Recover a Calendar Database" on page 379.**

22.5.6 Rebuilding a Corrupted Calendar Database

This section describes how to use the `csdb rebuild` command and contains the following topics:

- "22.5.6.1 `rebuild` Overview" on page 377
- "To Rebuild a Calendar Database" on page 377
- "22.5.6.2 Sample Rebuild Output" on page 378

22.5.6.1 `rebuild` Overview

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*/`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 databases, placing them into the `/tmp/db` directory.**

   Copy the database (`.db`) files and the log (`log.*`) files. You don't need to copy any share (`__db.*`) files.

4. **Change to the *cal-svr-base*/`SUNWics5/cal/sbin` directory.**

   For example, on Solaris Operating Systems, for the default directory, 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 path, `rebuild` uses 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*`/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.*`) or log (`log.*`) files from the corrupted database, move them to another directory.**

9 **Restart Calendar Server.**

22.5.6.2 Sample Rebuild Output

The following example shows the command and the output that it generated:

```
# ./csdb -g rebuild
Building calprops based on component information.
Please be patient, this may take a while...
Scanning events database...
512 events scanned
Scanning todos database...
34 todos scanned
Scanning events database...
512 events scanned
Scanning todos database...
34 todos scanned
Scanning deletelog database...
```

```
15 deletelog entries scanned
Scanning gse database...
21 gse entries scanned
Scanning recurring database...
12 recurring entries scanned
Successful components db scan
Calendar database has been rebuilt
Building components based on calprops information.
Please be patient, this may take a while...
Scanning calprops database to uncover events...
25 calendars scanned
Scanning calprops database to uncover todos...
25 calendars scanned
Successful calprops db scan
Calendar database has been rebuilt
```

Note – The preceding sample output shows the events and the todos databases scanned twice each. This is not an error. It scans the first time to verify the information in the calendar properties database and then scans again to make sure calendar properties database is accessible.

22.5.7 Using the Dump and Load Procedure to Recover a Calendar Database

This sections contains the following topics:

- "22.5.7.1 Dump and Load Overview" on page 379
- "To Perform the Dump and Load Procedure" on page 380

22.5.7.1 Dump and Load Overview

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`/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 the last good hot backup or archive backup of your database.

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

For the most up to date version of `cs5migrate,` call Sun technical support.

▼ To Perform the Dump and Load Procedure

1. **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 17, "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
```

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

6 **Repeat the previous two steps for the other corrupted database files.**

That is, run `db_dump` for the other corrupted database files.

7 **Rebuild the recovered database files using the `csdb rebuild` command, as described in "22.5.6 Rebuilding a Corrupted Calendar Database" on page 376.**

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, 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. In the meantime, you might need to revert to the last good backup of your database.

22.5.8 Restoring an Automatic Backup Copy

If you have used the automatic backup feature described in Chapter 9, "Configuring Automatic Backups (csstored)," you can use the hot backup copy when your live database is corrupted.

This sections covers how to restore the two different automatic backups:

- "22.5.8.1 Before You Restore" on page 381
- "To Restore a Hot Backup" on page 381
- "To Restore an Archive Backup" on page 383

22.5.8.1 Before You Restore

Before you restore a backup, be sure that you have:

- Tried to diagnose which transaction caused the corruption of the live database.
- Removed or corrected the corrupting transaction so the new archive will not be corrupted.
- Preserved the corrupted database by copying it to another directory or removable media. This is necessary should you need to contact technical support.

▼ To Restore a Hot Backup

Hot backups should be your first choice of backup when your live database is corrupted. To restore a hot backup, follow these steps:

1 **Identify any log files that were unapplied or open for writing in the corrupted live database directory.**

2 **Close the log that was open for writing. It contains the most recent transactions.**

3 **Create a new (recovery) directory.**

4 **Copy the current hot backup copy into the new recovery database directory.**

5 **Copy the `log.*` files from your corrupted live database directory into your new recovery database directory.**

6 **If you are keeping an archive copy of the database, copy the logs that had not been applied to the live database into the archive directory, so your archive backup copy will be complete.**

7 **Run `db_recover` with the `-c -h` options specified against the new recovery database.**

For example, if your new recovery directory is called `recoverydb`, then the command would be as follows:

```
db_recover -c -h recoverydb
```

8 **Leave the `log.*` files in the new recovery directory.**

The `db_recover` program applied the log files to the new recovery databases, but starting with version 42, the Berkeley DB expects them to remain.

9 **Run `db_verify` against the database files in the new recovery directory. To run `db_verify`:**

a. **Stop the Calendar Server using these commands.**

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

```
./stop-cal
```

b. **Create another copy of the Calendar Server database (`csdb`) using this command.**

```
cp -Rp /var/opt/SUNWics5/csdb /var/opt/SUNWics5/csdb.db_verify
```

c. **Run `db_verify` on the copy of `csdb`.**

Note – Do not run `db_verify` on the original `csdb`.

```
LD_LIBRARY_PATH=/opt/SUNWics5/cal/lib
export LD_LIBRARY_PATH
cd /opt/SUNWics5/cal/tools/unsupported/bin
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50alarms.db
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50calprops.db
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50events.db
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50gse.db
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50journals.db
```

```
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50recurring.db
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50todos.db
./db_verify -o -h /var/opt/SUNWics5/csdb.db_verify ics50deletelog.db
```

Note – Run `db_verify` with `-o` option for `ics50deletelog.db`.

If `db_verify` completes running successfully, you will not get any error messages. If the database file gets corrupted, it throws error messages. For example:

```
./db_verify -h /var/opt/SUNWics5/csdb.db_verify ics50todos.db
db_verify:Page 612: last item on page sorted greater than parent entry
db_verify: Page 612: incorrect next_pgno 885 found in leaf chain (should be 501)
db_verify: Page 0: page 501 encountered a second time on free list
db_verify: DB->verify: ics50todos.db: DB_VERIFY_BAD: Database verification failed
```

10 **Run `csdb -v list` against the new recovery directory.**

11 **If the new recovery directory passed all three preceding recovery steps, replace the old corrupted live database with the new recovery database.**

12 **Copy the new live database into your hot backup directory to function as the new snapshot.**

All new logs will be applied to this copy until the next regular snapshot is taken.

13 **Start Calendar Server.**

14 **If the new recovery directory failed any of the steps, identify an uncorrupted older hot backup as follows:**

a. **Working backward through your hot backups, find the most recent copy that is not corrupted by running `db_verify` and `csdb -v list` on each in turn.**

b. **The first hot backup copy that passes can be restored to your live database directory.**

Replace the corrupted live database with the clean hot backup, as described in "To Restore a Hot Backup" on page 381. (Be sure to read "22.5.8.1 Before You Restore" on page 381 first.)

c. **If none of your hot backups work and you do not have archive backups to try, call technical support. If you do have archive backups, follow the procedure that follows"To Restore an Archive Backup" on page 383. (See also, "22.5.8.1 Before You Restore" on page 381.)**

▼ To Restore an Archive Backup

If you do not have an uncorrupted hot backup, but have archive backups and their transaction logs, you can restore the most current uncorrupted version of the archived database by performing the following steps:

1. **Identify any log files that were unapplied or open for writing in the corrupted live database directory.**

2. **Close the log that was open for writing. It contains the most recent transactions.**

3. **Create a new (recovery) directory.**

4. **Copy the most recent archive copy and its log files into the new recovery database directory.**

5. **Copy any unapplied `log.*` files from your corrupted live database directory into your new recovery database directory.**

6. **Run `db_recover` with the `-c-h` options specified against the new recovery database.**

   For example, if your new recovery directory is called `recoverydb`, then the command would be as follows:

   ```
   db_recover -c -h recoverydb
   ```

7. **Leave the `log.*` files in the new recovery directory.**

   The `db_recover` program applied the log files to the new recovery databases, but starting with version 4.2, Berkeley DB expects the log files to still be there.

8. **Run `db_verify` against the database files in the new recovery directory.**

   Steps in the "To Restore a Hot Backup" on page 381 procedure, explains how to run `db_verify`.

9. **Run `csdb -v list` against the new recovery directory.**

10. **If the new recovery directory passed all three preceding recovery steps, replace the old corrupted live database with the new recovery database.**

11. **Copy the new live database into your hot backup directory to function as the new snapshot.**

12. **Start Calendar Server.**

13. **If the new recovery directory failed any of the steps, identify an uncorrupted older archive backup as follows:**

    a. **Working backward through your archive backup copies, find the most recent copy that is not corrupted by running the three recovery programs against each of them in turn: `db_recover -c-h`, `db_verify`, and `csdb -v list`.**

    b. **The first archive copy that passes can be restored to your live database directory.**

       Replace the corrupted live database with the clean archive backup, as shown in "To Restore an Archive Backup" on page 383.

c. If none of your archive backups work, call technical support.

22.5.9 Repairing Custom Backup Scripts

This section includes the following topics:

- "22.5.9.1 Berkeley Tools Now Compiled with a Dynamic Library" on page 385
- "22.5.9.2 To Repair a Custom Backup Script" on page 385

22.5.9.1 Berkeley Tools Now Compiled with a Dynamic Library

If you have created a custom backup script using the Berkeley database tools, such as `db_recover`, you may find that it will no longer work after upgrading to Calendar Server. The reason for this is that the earlier versions of Calendar Server compiled the tools with a static library. The tools are now compiled with a dynamic library, `libdb-4.2.so`.

22.5.9.2 To Repair a Custom Backup Script

To use the new dynamic library with your existing custom scripts, set the following global variable as shown:

```
LD_LIBRARY_PATH=libdb-4.2.so
```

PART V

Appendixes

This part contains the appendixes for the Administration Guide.

- Appendix A, “Directory Configuration Worksheet”
- Appendix B, “Calendar Server Configuration Worksheet”
- Appendix C, “Calendar Server Configuration Worksheet”
- Appendix D, “Calendar Server Command-Line Utilities Reference”
- Appendix E, “Calendar Server Configuration Parameters”

APPENDIX A

Directory Configuration Worksheet

This worksheet helps you collect the information you will be asked for when running `comm_dssetup.pl`. The first column shows you the silent mode options and the expected information that is to follow it. The second column shows you the same option in interactive mode with the default answer.

There is a line provided in the right column for the answer you want to give. It applies to both the silent and interactive modes. For silent mode, use the answers as the value that follows the option. For interactive mode, enter your value at the prompt.

For examples and instructions on how to run `comm_dssetup.pl`, see the *Sun Java System Communications Suite 5 Installation and Configuration Guide*.

TABLE A–1 Directory Server Setup Script (comm_dssetup.pl) Worksheet

Silent Mode Options	Interactive Dialog and Defaults
`-i` yes | no	Add new Directory Server indexes (yes/no). Default: yes Your value:
`-R` yes | no	Reindex now (yes/no). Default: yes Your value:
`-c` Directory Server Root	Directory Server root path name. Default: `/var/opt/Sun/dsins` Your value:

TABLE A–1 Directory Server Setup Script (comm_dssetup.pl) Worksheet *(Continued)*

Silent Mode Options	Interactive Dialog and Defaults
`-d` Directory Server Instance	Directory Server instance subdirectory. Default: none Your value:
`-r` DC Root Suffix	DC Tree root suffix. Default: `o=internet` Your value:
`-u` User and Group Base Suffix	User/Group root suffix. Default: `o=usergroup` Your value:
`-s` yes | no	Update schema (yes/no). Default: yes Your value:
`-D` Directory Manager DN	Directory Manager Distinguished Name (DN). Default: "`cn=Directory Manager`". Your value:
`-w` Directory Manager DN Password	Directory Manager DN password. Default: none. Your value:
`-b` yes | no	Use this directory to store both configuration and user data (yes) or configuration data only (no). Default: yes Your value:
`-t 1|1.5|2`	Schema version: ■ Option 1 – Schema version 1 ■ Option 1.5 – Schema version 2 Compatibility Mode ■ Option 2 – Schema version 2 Native Mode Default: 1 Your value:

TABLE A–1 Directory Server Setup Script (comm_dssetup.pl) Worksheet *(Continued)*

Silent Mode Options	Interactive Dialog and Defaults
`-m yes|no`	Do you want to modify the directory server? Default: yes no- prints out script but does not execute it.
`-S` *PathtoSchemaFile*	Path to the directory where the schema files are located. Default: `./schema` Your value:

APPENDIX B

Calendar Server Configuration Worksheet

This appendix contains the following worksheets to help you keep track of the information you need to run the Calendar Server configuration program, which is described in Chapter 2, “Initial Runtime Configuration Program for Calendar Server 6.3 software (csconfigurator.sh)”

B.1 Administration, User Preferences and Authentication Screen Worksheet

TABLE B–1 Administration, User Preferences and Authentication Screen Worksheet

Option	Description
LDAP Server Host Name	Host name of the LDAP directory server you are using for user authentication. Default: current host. Your value:
LDAP Server Port	Port number that the LDAP server listens on. Default: 389. Your value:

TABLE B–1 Administration, User Preferences and Authentication Screen Worksheet *(Continued)*

Option	Description
Base DN	Entry in the LDAP directory used as the starting point from which searches will occur. Default: `o=host.com`. 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:
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:

B.2 Email and Email Alarms Worksheet

TABLE B–2 Email and Email Alarms Worksheet

Option	Description
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:

TABLE B–2 Email and Email Alarms Worksheet *(Continued)*

Option	Description
SMTP Host Name	Host name of the SMTP server where email alarm messages should be sent. Default: Current host. Your value:

B.3 Runtime Configuration Worksheet

TABLE B–3 Runtime Configuration Worksheet

Option	Description
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 CPU's on the server where you are installing Calendar Server. Your value:
Runtime User ID	UNIX user name under which Calendar Server will run. Default: `icsuser`. Your value:
Runtime Group ID	UNIX group under which Calendar Server will run. Default: `icsgroup`. Your value:

TABLE B–3 Runtime Configuration Worksheet *(Continued)*

Option	Description
Calendar Server Startup	Start after successful installation. Default: Checked. Your value: Start on system startup. Default: Checked. Your value:

B.4 Database, Logs, and Temporary Files Directories Worksheet

TABLE B–4 Database, Logs, and Temporary Files Directories Worksheet

Option	Description
Database Directory	Directory where Calendar Server should create and store the calendar database files. Default: `/var/opt/SUNWics5/csdb` Your value:
Logs Directory	Directory where Calendar Server writes log files. Default: `/var/opt/SUNWics5/logs` Your value:
Temporary Files Directory	Directory where the Calendar Server writes temporary files. Default: `/var/opt/SUNWics5/tmp` Your value:

APPENDIX C

Calendar Server Configuration Worksheet

C.1 Calendar Server Configuration Worksheet

The following table lists the values you set when you run the Calendar Server configuration program (`csconfigurator.sh`).

TABLE C–1 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:
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:

TABLE C–1 Calendar Server Configuration Worksheet *(Continued)*

Component	Description and Comments
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 CPU's on the server where you are installing Calendar Server. Your value:

TABLE C–1 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 D

Calendar Server Command-Line Utilities Reference

For a short description of each utility, see "D.2 Short Description of Command-Line Utilities" on page 403.

Calendar Server provides command-line utilities not included in the Delegated Administrator bundled with Access Manager.

These Calendar Server utilities can be invoked from batch, shell, and scripting programs such as Perl. Some of these utilities (csuser, csresource and csdomain) have been superseded by the Delegated Administrator utility, but the rest are still used, even in a Schema version 2 environment. For Schema version 1, you must continue to use csuser, csresource and csdomain, and not use Delegated Administrator.

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/`SUNWics5/cal/sbin`

All of the utilities must be started from the `sbin` directory, with the exception of `start-cal` and `stop-cal` which can be run from any directory, if the full path is specified.

Note – Error messages from these administrative tools are written to the `admin.log` file found in the `csdb` directory.

This chapter provides the following information:

D.1 Running the Command-Line Utilities

Run the command-line utilities while logged in as the user and group where Calendar Server is running, or as `root`. This was specified during installation; the defaults are `icsuser` and `icsgroup`.

For example, if your Calendar Server base directory is *cal-svr-base*, to run the `cscal` utility `list` command, you would do the following after logging in:

```
cd cal-svr-base/SUNWics5/cal/sbin
./cscal list
```

D.1.1 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 semicolon 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.

D.1.2 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.

D.1.3 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.

D.2 Short Description of Command-Line Utilities

The following table gives a short description the Calendar Server command-line utilities.

TABLE D–1 Calendar Server Command-Line Utilities Summary

Utility	Description
“D.3 csattribute” on page 405	Manages the LDAP attributes of a calendar user or resource for Schema version 1.

TABLE D–1 Calendar Server Command-Line Utilities Summary *(Continued)*

Utility	Description
"D.4 csbackup" on page 407	Backs up individual calendars, users, and the calendar database.
"D.5 cscal" on page 409	Manages calendars and their properties.
"D.6 csclean" on page 414	Removes user and resource calendars for Calendar Server users whose status attribute (inetUserStatus) has been marked as "deleted" by Delegated Administrator.
"D.7 cscomponents" on page 415	Manages calendar components: events and tasks (todos).
"D.8 csdb" on page 417	Manages the calendar database.
"D.9 csdomain" on page 420	Manages Calendar Server attributes in the domain LDAP entry for Schema version 1.
"D.10 csexport" on page 429	Exports a calendar in iCalendar (.ics) or XML (.xml) format.
"D.11 csimport" on page 430	Imports a calendar in iCalendar (.ics) or XML (.xml) format.
"D.13 cspurge" on page 434	Allows the manual purge of entries in the Delete Log database (ics50deletelog.db).
"D.14 csrename" on page 436	Allows the renaming of user ID's. Causes the whole database to be rewritten.
"D.15 csresource" on page 438	Manages calendar resources such as conference rooms and equipment.
"D.16 csrestore" on page 441	Restores individual calendars, users, and the calendar database.
"D.17 csschedule" on page 444	Manages scheduling entries in the Group Scheduling Engine (GSE) queue.
"D.18 csstats" on page 446	Displays counters in a Calendar Server.
"D.19 csuser" on page 448	Manages calendar users for Schema version 1.
"D.20 start-cal" on page 452	Starts all Calendar Server processes.
"D.21 stop-cal" on page 453	Stops all Calendar Server processes.

D.3 csattribute

The csattribute utility only works in Schema version 1 mode. It manages Calendar Server user or resource LDAP entry attributes. 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 when trying 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 Chapter 5, “Configuring Calendar Database Distribution Across Multiple Machines in Calendar Server Version 6.3”

D.3.1 Requirements

- You must be using Schema version 1.
- Calendar Server can be running or stopped.
- You must be logged in as the user or group under which Calendar Server is running (such as icsuser and icsgroup) that was specified during installation, or as root.

D.3.2 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
```

The following table describes the commands available for csattribute.

TABLE D–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.

The following table describes the `csattribute` utility command options.

TABLE D–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 domain. Default is taken from the *service.defaultdomain* parameter in the `ics.conf` file.

D.3.3 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
  ```

D.4 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 "D.16 csrestore" on page 441.

D.4.1 Requirements

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

D.4.2 Syntax

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

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

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

The following table describes the commands available for csbackup.

TABLE D–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*/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 nonexistent 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.

The following table describes the `csbackup` utility command options.

TABLE D–5 csbackup Utility Command Options

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

TABLE D–5 csbackup Utility Command Options *(Continued)*

Option	Description
-b *basedn*	The base DN to be used for this user. The default is taken from the setting *service.schema2root*, 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 “15.2 Creating Calendar Unique Identifiers (calid's)” on page 292.
-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 17, “Backing Up and Restoring Calendar Server Data.”

D.4.3 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
  ```

D.5 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

D.5.1 Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- 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`.

D.5.2 Syntax

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

cscal [-q|-v]
      [-o owner’s uid]
      [-O]
      delete|reset calid

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

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

Note – Despite the fact that `cscal` does not check case when you enter the `-o` (owner’s *uid*), the search is case insensitive.

The following table describes the commands available for the `cscal` utility.

TABLE D–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, all calendars for a specific user must reside on the same back-end server, as indicated by the user's *icsDWPHost* LDAP attribute. If you try to create a calendar for the user 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 *uid*.
`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 *uid*.
`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 *uid*.
`list [` *calid* `]`	Lists properties of the calendar specified by *calid*. If *calid* is not specified, lists properties of all calendars. If the `-o` *owner's uid* option is specified, lists all calendars whose primary owner is the specified owner *uid*. **Note –** When any of the permissions have been modified through the Calendar Express user interface, this command lists those ACE letters capitalized. There is no significance to this. The ACEs are case insensitive.
`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.

The following table describes the `cscal` utility command options.

TABLE D–7 `cscal` Utility Command Options

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

TABLE D–7 cscal Utility Command Options *(Continued)*

Option	Description
-a [*aces*]	Access Control Entries (ACE's) for a specified calendar. ACE's 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 "1.8 Access Control for Calendar Server Version 6.3" on page 51.
-c *charset*	Character set. The default is no character set.
-d *description*	Description (a viewable comment about the purpose of the calendar). The default is no description.
-g *category*	Category. Multiple categories must be enclosed in quotation marks ("") and separated by spaces. The default is no category.
-k yes|no	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 database, ics50calprops.db, to determine if doublebooking is allowed. If you need to change the calendar properties for a calendar to allow or disallow doublebooking, 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	Viewable Name. The default is no name.
-o owner	(Lowercase o) Primary owner. The default setting is the unique ID (uid) of the primary owner.
-O	(Uppercase 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.

D.5.3 Possible Problems Creating a Calendar on a Back-End Machine

If you try to create a calendar on a designated back-end machine, and you get the following error message: Invalid DWP Host Server. it means one of two things. Either your server is not configured properly, or the calendar owner has already been assigned to a different back-end server.

D.5.3.1 Back-End Machine Not Configured Properly

Look at the ics.conf file for the back-end server in question. Verify that the following settings exist:

```
service.dwp.enable = "yes"
caldb.cld.type = "directory"
local.hostname = "back-end hostname"
```

D.5.3.2 Calendar Owner Assigned to a Different Back-End Machine

Look at the user's LDAP entry and see if there is an *icsDWPHost* attribute present. The value of *icsDWPHost* must match the back—end server name on which you are attempting to create the calendar. You can not create a calendar for this user on a different back-end server.

D.5.4 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@sesta.com`:

  ```
  cscal -a "@^a^rw^g" -g sports -y jsmith@sesta.com 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 doublebooking:

  ```
  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
  ```

D.6 `csclean`

The `csclean` utility only works in Schema version 2 mode. It removes user and resource calendars for users whose status attribute (*icsStatus*) has been marked as “deleted” by Delegated Administrator.

For Schema version 1, use `csuser` and `cscal` to remove all of the calendars for a deleted user.

D.6.1 Requirements

- You must be using Schema version 2.
- Calendar Server can be running or stopped.
- You must run `csclean` locally on the machine where Calendar Server is installed.
- 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`.

D.6.2 Syntax

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

The following table describes the `csclean` utility command options.

TABLE D–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.

D.6.3 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 clean "*"
  ```

D.7 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.

D.7.1 Requirements

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

- 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.

D.7.2 Syntax

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

The following table describes the commands available for the cscomponents utility.

TABLE D–9 cscomponents Utility Commands

Command	Description
delete *calid*	Deletes events and tasks in the calendar with the specified calendar ID.
list *calid*	Lists events and tasks in the calendar with the specified calendar ID. **Note –** When deleting tasks, you must specify the -soption with an actual DateTime Z String specifying the starting date for removal. If you do not specify a date, or you specify zero (0) as the value for the option, all tasks will be deleted from the calendar.
version	Prints the version of the utility to the screen.

The following table describes the cscomponents utility command options.

TABLE D–10 cscomponents Utility Command Options

Option	Description
-v	Run in verbose mode: Display all available information about the command being performed. Default is off.
-q	Run in quiet mode: ■ Display no information if the operation is successful (errors, if they occur, are displayed). ■ Suppress confirmation prompting for dangerous commands. Default is off.
-e *endtime*	Ending time of the components. An end time of 0 means to the end of time. The default is 0.
-s *starttime*	Starting time of the components. A start time of 0 means from the beginning of time. The default is 0. **Note –** For tasks only, you must specify this option and the starting date specified must be a DateTime Z String. If the option is not specified, or the option is specified but the value is set to zero, all tasks for this calendar will be deleted.

TABLE D–10 cscomponents Utility Command Options *(Continued)*

Option	Description
-t event|task	Type of components (events or tasks) on which the action is performed. Default is both.

D.7.3 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
  ```

D.8 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.

D.8.1 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.
- 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.

D.8.2 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 [-a, -V]
     [-g] [dbdir [dstdir]]
```

The following table describes the commands available for the `csdb` utility.

TABLE D–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 calendar databases in the specified database directory. If a database directory is not specified, the current directory is used. Is not implemented for session or statistics databases.
`check [dbdir]`	Scans a calendar database in the specified database directory to determine if any corruption has occurred and reports the results in its output. If a database directory is not specified, the current directory is used.
`rebuild [dbdir [dstdir]]`	Scans all calendar databases 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. After the databases are rebuilt, `db_verify` runs. The *dstdir* specifies an optional destination directory.
`version`	Displays the version of the utility.

The following table describes the `csdb` utility command options.

TABLE D–12 csdb Utility Command Options

Option	Description
-a	For rebuild command only, rebuilds only the alarms database.
-V	Must be passed in with -a for rebuilding the alarms database. Causes verify to be run against all databases, including alarms.
-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 group: ■ caldb (calendar) ■ sessdb (session) ■ statdb (statistics) Note: If -t is not specified, csdb operates on all database groups, except for the check, recover and rebuild commands, which operate only on caldb (calendar).

D.8.3 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
  ```

- Rebuild the alarms database only:

  ```
  csdb -a -V rebuild
  ```

D.9 csdomain

The csdomain utility manages Calendar Server attributes in the domain LDAP entry. These attributes are part of the icsCalendarDomain object class. Commands are:

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

D.9.1 Requirements

- To run csdomain, 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" , "1.5", or "2").
    - If local.schemaversion = "1" or "1.5", 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 have followed the instructions in Chapter 10, "Setting Up a Multiple Domain Calendar Server 6.3 Environment," before using csdomain to add Organization Tree nodes.

    You must run csdomain locally on the machine where Calendar Server is installed.

  Calendar Server can be running or stopped.
- 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.

D.9.2 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
```

The following table describes the commands available for the `csdomain` utility.

TABLE D–13 `csdomain` Utility Commands

Command	Description
`create`	Create a new domain in the LDAP directory.
`add`	Add a Calendar Server attribute and its associated value in the domain LDAP entry. If you add or update domain 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 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.

The following table describes the `csdomain` utility command options.

TABLE D–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 “D.9.3 LDAP Attributes and Property Names” on page 422.

TABLE D–14 csdomain Utility Command Options *(Continued)*

Option	Description

-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 version 1 – Specifies the node under which all users and resources are created. For example: `o=node2,o=node1,o=sesta` ■ For LDAP Schema version 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`

D.9.3 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.

- "D.9.3.1 *icsAllowRights* Attribute: `csdomain` Utility" on page 422
- "D.9.3.2 *icsExtendedDomainPrefs* Attribute: `csdomain` Utility" on page 424
- "D.9.3.3 Other LDAP Directory Attributes: `csdomain` Utility" on page 427

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

D.9.3.1 *icsAllowRights* Attribute: `csdomain` Utility

"D.9.3 LDAP Attributes and Property Names" on page 422 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* programmatically in bitwise operations, you must first convert its string value to an integer.

TABLE D–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*
2	`allowPublicWritableCalendars`	If set (bit 2=1), do not allow public writable calendars. Corresponding `ics.conf` parameter: *service.wcap.allowpublicwriteablecalendars*
3		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		Not used in the current release.
7		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*

TABLE D–15 *icsAllowRights* LDAP Directory Attribute and Properties *(Continued)*

Bit	Property Name	Description
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*
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*
15–31		Not used in the current release.

D.9.3.2 *icsExtendedDomainPrefs* Attribute: `csdomain` Utility

The following table 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 (for example, value = `0`, or `service.virtualdomain.support=“no”`), or is not present, Calendar Server uses the corresponding `ics.conf` parameter as the default value.

TABLE D–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 = "yes")
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")
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.siteadmin.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.siteadmin.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 "1.8.3 Access Control Lists (ACLs) in Calendar Server Version 6.3" on page 52. This ACL is used for cross domain searches. For more information, see "11.2 Cross Domain Searching in Calendar Server 6.3 Systems" on page 244. **Caution** – Only a single instance of domainAccess is allowed. However, the system does not warn you if there is a duplicate. You must ensure there is only one, whenever you change the value.
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")

TABLE D–16 *icsExtendedDomainPrefs* LDAP Directory Attribute *(Continued)*

Property Name	Description
`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 size of an LDAP group that will be expanded for an invitation. Corresponding `ics.conf` parameter: *calstore.group.attendee.maxsize* (default is "0" – expand the group without regard to size)
`language`	Specifies the language for a domain. Corresponding `ics.conf` parameter: *local.domain.language*

`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)

TABLE D–16 *icsExtendedDomainPrefs* LDAP Directory Attribute *(Continued)*

Property Name	Description
`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 the user interface 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 the user interface 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)
`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)

D.9.3.3 Other LDAP Directory Attributes: `csdomain` Utility

The following table describes other LDAP attributes and properties that you can set with the `csdomain` utility.

TABLE D–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	(undefined)	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.
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 default time-zone ID. For example, America/New_York or Asia/Tokyo. For the supported time zones, refer to the `timezones.ics` file.

D.9.4 Examples

- Create a new domain using LDAP Schema version 1 named `west.sesta.com`:

  ```
  csdomain -v -n o=nodewest,o=sesta create west.sesta.com
  ```
- Create a new domain using LDAP Schema version 2 named `east.sesta.com`:

  ```
  csdomain -v -n nodeeast create east.sesta.com
  ```
- Display a list of Calendar Server LDAP attributes for the domain named `west.sesta.com`:

  ```
  csdomain -v list west.sesta.com
  ```
- Set the time zone to `America/New_York` for the domain named `west.sesta.com`:

  ```
  csdomain -v -a timezone=America/New_York add west.sesta.com
  ```

D.10 `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.

D.10.1 Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- 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`.

D.10.2 Syntax

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

The following table describes the commands available for the `csexport` utility.

TABLE D–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 filename extension: ■ .ics for iCalendar (text/calendar) ■ .xml for XML (text/xml)
version	Display the version of the utility.

The following table describes the csexport utility command options.

TABLE D–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.

D.10.3 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
  ```

D.11 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.

Date calculations for importing a calendar's components use the time zone specified in the *X-NSCP-DTSTART-TZID* associated with the component. If none is present, then the server time zone found in the `ics.conf` is used.

D.11.1 Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- 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`.

D.11.2 Syntax

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

The following table describes the commands available for the `csimport` utility.

TABLE D–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 filename extension: ■ `.ics` for iCalendar (text/calendar) ■ `.xml` for XML (text/xml)
`version`	Display the version of the utility.

The following table describes the `csimport` utility command options.

TABLE D–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.

TABLE D–21 csimport Utility Command Options *(Continued)*

Option	Description
`-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 "15.2 Creating Calendar Unique Identifiers (calid's)" on page 292.

D.11.3 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
  ```

D.12 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 D–23.)
- `list` displays all supported plug-ins.
- `version` displays the version number of the utility currently installed.

D.12.1 Requirements

- Must be run on the local machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- 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`.

D.12.2 Syntax

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

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

The following table describes the commands available for the `csplugin` utility.

TABLE D–22 `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 D–23.)
`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 D–23.)
`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 D–23.)
`version`	Display the version of the utility.

The following table describes the `csplugin` utility command options.

TABLE D–23 `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 D–23 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.

D.12.3 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
  ```

D.13 cspurge

The cspurge utility allows the manual purge of entries in the Delete Log database (ics50deletelog.db).

D.13.1 Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- 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*.

D.13.2 Syntax

```
cspurge [-q|-v]
        -e endtime
        -s starttime
```

The following table describes the `cspurge` utility command options.

TABLE D–24 `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.
`-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.

D.13.3 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
  ```

D.14 csrename

The csrename utility allows you to rename one or more calendar users. This utility renames calendar users as follows:

- Calendar database files–Renames users (user ID's) in the calendar database files and then writes the new database files to a destination directory. The existing calendar database files are not modified.
- LDAP directory server–Converts the user ID's in the Calendar Server LDAP attributes (that is, attributes with the “ics” prefix). The LDAP directory server is modified in place.

The csrename utility is located in the following directory:

cal-svr-base/SUNWics5/cal/sbin

D.14.1 Requirements

Before you run csrename, you must first:

- Create an input mapping file (-m option) for the users you want to convert.
- Create any new users in the LDAP directory server, if necessary.
- Stop Calendar Server.

To run csrename, you must log in as icsuser (or as the Calendar Server runtime user ID specified during configuration). If you run csrename as superuser (root), you might need to reset the permissions for the new database files. To modify the LDAP directory server attributes, you must also have administrative rights for that directory.

If your Calendar Server installation has a front-end/back-end server configuration, you must run csrename on each back-end server.

D.14.2 Syntax

Use the following syntax to run csrename:

```
csrename [-t DestinationDB]
         [-c ConfigFile]
         [-e ErrorFile]
          -m MappingFile
         rename [DB|LDAP]
```

table lists the options for this utility and gives a description of each.

TABLE D–25 Options for `csrename`

Option	Description
`-t` *DestinationDB*	Specifies the destination directory where `csrename` generates the new database with the converted user names. The default is *MigratedDB*. After `csrename` is finished, the *caldb.berkeleydb.homedir.path* parameter in the `ics.conf` file must point to the destination database. Either reset *caldb.berkeleydb.homedir.path* to point to the destination database directory, or move the destination database files to the directory indicated by the parameter.
`-c` *ConfigFile*	An input parameter that specifies a Calendar Server configuration file. The default is the `ics.conf` file. The `csrename` utility uses the *caldb.berkeleydb.homedir.path* parameter in the configuration file to determine the location of the input calendar database. The default location of the calendar database is `/var/opt/SUNWics5/csdb`.
`-e` *ErrorFile*	The file where `csrename` writes any errors or database entries that cannot be resolved. The default is `MigrateError`.
`-m` *MappingFile*	Specifies an input mapping file. The default is `MigrateMapping`. The input mapping file is a text file that maps existing user ID's to new user ID's. You must create the mapping file before you run `csrename`. Specify one entry per line with a space between the old and new values. For example: *tchang tc897675* *jsmith js963123* *bkamdar bk548769* If upon auditing your results, you find that one or more of your intended name changes was omitted, you can fix the error by creating a new mapping file with only the missed names in it and rerunning `csrename`.
`DB|LDAP`	Specifies the database that gets updated: `DB` converts user ID's in the new calendar database only (default). `LDAP` converts user ID's in both the new calendar database and the LDAP directory server attributes.

D.14.3 Examples

- Rename users based on the mapping file named `DBMapFile` and create the new calendar database in the destination directory named `newcalDB`:

  ```
  csrename -t newcalDB -m DBMapFile rename DB
  ```

- Rename users based on values in the mapping file named `NewNames`, create the new calendar database in the destination directory named `NewDB`, and modify the Calendar Server attributes in the LDAP directory server:

  ```
  csrename -t NewDB -m NewNames rename LDAP
  ```

D.15 csresource

The csresource utility creates and manages LDAP entries and calendars for resources, such as conference rooms or equipment. (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

D.15.1 Requirements

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

D.15.2 Syntax

```
csresource [-q|-v]
           [-a aces]
           [-b  basedn]
           [-d domain]
           [-t description]
           [-k yes|no]
           [-o owner]
           [-y otherowners]
            -m email
            -c calid
           create common_name

csresource [-q|-v]
           [-b basedn]
           [-d domain]
           delete|disable|enable [common_name]

csresource [-q|-v]
           [-b basedn]
           [-d domain]
           [-h host]
           list [common_name]
```

Then following table describes the commands available for the csresource utility.

TABLE D–26 csresource Utility Commands

Command	Description
`create common_name`	Create a new resource for a specified calendar ID.
`delete [common_name]`	Delete a resource or, if no resource *common_name* is specified, delete all resources.
`enable [common_name]`	Enable a resource or, if no resource *common_name* is specified, enable all resources.
`disable [common_name]`	Disable a resource or, if no resource *common_name* is specified, disable all resources.
`list [common_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 (" ").

The following table describes the csresource utility command options.

TABLE D–27 csresource Utility Command Options

Option	Description
`-v`	Run in verbose mode: Display all available information about the command being performed. Default is off.
`-q`	Run in quiet mode: ■ Display no information if the operation is successful (errors, if they occur, are displayed). ■ Suppress confirmation prompting for dangerous commands. Default is off.
`-a [aces]`	Access Control Entries (ACE's) for the specified calendar. ACE's 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 "15.4 Calendar Access Control" on page 296.
`-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.

TABLE D–27 csresource Utility Command Options (Continued)

Option	Description
-c calid	The *icsCalendar* attribute. This option is required with the create command. For more information, see "15.2 Creating Calendar Unique Identifiers (calid's)" on page 292.
-d domain	Specifies the name of a 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 database (`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.siteadmin.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	Display the version of the utility.

D.15.3 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 -m room100@sesta.com -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 resource calendar with the viewable name *MeetingRoom100* and allow doublebooking:

  ```
  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
  ```

D.16 `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 – The Calendar Server version 6.3 `csrestore` utility is not compatible with the Calendar Server version 2 `csrestore`utility. Do not try to restore data that was backed up using the `csrestore` in version 2 because data loss can occur.

D.16.1 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.
- 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`.

Note – csrestore does not take care about the user LDAP entries, subscribed or own calendar. You need to manually perform on the User LDAP entire to get the personal calendar back on the multi valued attribute, icsSubscribed.

D.16.2 Syntax

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

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

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

The following table describes the commands available for the csrestore utility.

TABLE D–28 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 filename 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 filename extension: ■ .ics for iCalendar (text/calendar). ■ .xml for XML (text/xml).
version	Display the version of the utility.

The following table describes the csrestore utility command options.

TABLE D–29 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.
`-c` *calid*	The calendar ID to restore. This option is required with the `calendar` command. There is no default. For more information, see "15.2 Creating Calendar Unique Identifiers (calid's)" on page 292.

D.16.3 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
  ```

D.17 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 specified calendar ID.
- delete removes an entry from the GSE queue requested by a specified calendar ID.
- version displays the version number of the utility currently installed.

D.17.1 Requirements

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

D.17.2 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]
```

The following table describes the commands available for the `csschedule` utility.

TABLE D–30 `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.

Then following table describes the `csschedule` utility command options.

TABLE D–31 `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`

TABLE D–31 `csschedule` Utility Command Options *(Continued)*

Option	Description
`-u` *uid*	The unique identifier (UID) of an entry in the GSE queue.

D.17.3 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
  ```

D.18 `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 "E.3 Counters Configuration (counter.conf) File" on page 497.

D.18.1 Requirements

- You must run the utility locally on the machine where Calendar Server is installed.
- Calendar Server can be running or stopped.
- 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`.

D.18.2 Syntax

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

The following table describes the commands available for the `csstats` utility.

TABLE D–32 `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.

The following table describes the `csstats` utility command options.

TABLE D–33 `csstats` Utility Command Options

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

TABLE D–33 csstats Utility Command Options *(Continued)*

Option	Description
-q	Run in quiet mode: ■ Display no information if the operation is successful (errors, if they occur, are displayed). ■ Suppress confirmation prompting for dangerous commands. Default is off.
-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.

D.18.3 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
  ```

D.19 csuser

The csuser utility works in Schema version 1 mode only. It manages calendar users' LDAP entries, and the users' default calendars. Commands are:

- check Checks whether a user is enabled for calendaring.
- create Enables a user for calendaring.

Note – This utility does not enable users for Address Book as is required for Communications Express. This will have to be done manually with ldapmodify.

- delete Deletes a user and the user’s default calendar.

  Tip – If the user has other calendars, they are not deleted. Use cscal to remove other calendars of a deleted user.

- `disable` Prevents a user from logging in to Calendar Server.
- `enable` Allows a user to log on to Calendar Server.
- `list` Lists a user's calendar attributes.
- `reset` Removes all calendar attributes from the LDAP entry, including `icsCalendarUser` (object class), `icsSubscribed`, `icsCalendarOwned`, `icsCalendar`, and `icsDWPHost` (if the user is in an LDAP CLD setup).

Tip – After this command has been issued, the user is no longer enabled for calendar service. If you want to restore calendar services to the user, issue a `csuser enable` command.

If you are using Directory Server, you can also use the `ldapsearch` and `ldapmodify` utilities. For information about these utilities, see the Directory Server documentation on the following Web site:

`http://docs.sun.com/coll/1316.2`

D.19.1 Requirements

- You must be using Schema version 1.
- 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.
- 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`.

D.19.2 Syntax

```
csuser [-q|-v]
       [-a aces]
       [-b basedn]
        -m email address
       [-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
```

The following table describes the commands available for the `csuser` utility.

TABLE D–34 `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`	Disables the specified user ID for calendaring by adding `icsAllowedServiceAcess="http"` to the user's LDAP entry.
`enable userid`	Enables the specified user ID for calendaring by removing `icsAllowedServiceAcess="http"` from the user's LDAP entry.
`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.
`version`	Display the version of the utility.

The following table describes the `csuser` utility command options.

TABLE D–35 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 domain. Default is taken from the *service.defaultdomain* parameter in the `ics.conf` file.

`-a [aces]`	Access Control Entries (ACE's) for a specified calendar. ACE's 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 "E.2.9 Calendar Server Services Configuration" on page 466.

`-f filename`	File name to specify a password for options that require a password (`-y` parameter). If you are running csuser from a script, for added security, specify the password in filename.
`-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 address`	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.

D.19.3 Examples

- Check if the calendar user `jsmith@sesta.com` is enabled for calendaring (if the existing calendar user has access to calendar data for this Calendar Server):

  ```
  csuser check jsmith@sesta.com
  ```
- Create an LDAP user with the user ID `jsmith@sesta.com` 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@sesta.com -d sesta.com
  ```
- Delete the calendar user `jsmith@sesta.com`

  ```
  csuser delete jsmith@sesta.com
  ```
- Disable the calendar user `jsmith@sesta.com` from logging in to Calendar Server:

  ```
  csuser disable jsmith@sesta.com
  ```

Note – This command prevents `jsmith@sesta.com` 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, he retains access to calendar data until he logs off.

- Enable `jsmith@sesta.com` for calendaring (lets existing calendar user log in to Calendar Server):

  ```
  csuser enable jsmith@sesta.com
  ```
- List all calendar attributes for `jsmith@sesta.com`:

  ```
  csuser -v list jsmith@sesta.com
  ```
- List all calendar user ID's prefixed with the string *user*:

  ```
  csuser -v list "user*"
  ```
- Reset all calendar attributes for `jsmith@sesta.com` to the default configuration settings:

  ```
  csuser reset jsmith@sesta.com
  ```
- List all calendar attributes for *tchang* on the back-end server *sesta*:

  ```
  csuser -v -h sesta list tchang
  ```

D.20 `start-cal`

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

- `watcher` — Monitors Calendar Server daemons
- `enpd` — Event Notification Service (ENS)
- `csstored` — Performs maintenance and backup operations on the database

- `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

D.20.1 Requirements

- You must run `start-cal` locally on the machine where Calendar Server is installed.
- You must be logged in as `root`.

D.20.2 Syntax

```
start-cal
```

D.20.3 Example

cal-svr-base`/SUNWics5/cal/sbin/start-cal`

For more information, see "12.1 Starting and Stopping Calendar Server 6.3 Processes" on page 252.

D.21 `stop-cal`

The `stop-cal` utility stops all Calendar Server services.

D.21.1 Requirements

- You must run `stop-cal` locally on the machine where Calendar Server is installed.
- You must be logged in as `root`.

D.21.2 Syntax

```
stop-cal
```

D.21.3 Example

cal-svr-base`/SUNWics5/cal/sbin/stop-cal`

For more information, see "12.1 Starting and Stopping Calendar Server 6.3 Processes" on page 252.

APPENDIX E

Calendar Server Configuration Parameters

Calendar Server configuration parameters are stored in configuration files, including `ics.conf` and `counter.conf`.

This chapter provides the following information:

E.1 Editing the `ics.conf` Configuration File

Calendar Server configuration parameters are stored in the following file:

`/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.

For example: 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.

▼ 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 `/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`. 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.
     - All options must start at the beginning of a line.

4. **After you make changes to parameters in the `ics.conf` file, stop and then restart 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 "12.1 Starting and Stopping Calendar Server 6.3 Processes" on page 252.

E.2 Configuration Parameters (`ics.conf`) File

This section lists the various configuration parameters in the `ics.conf` file. For convenience, they are broken down into functional groups as follows:

Caution – The parameters listed below merely show their default settings. To implement certain features, you might need to change one or more parameters to different values. Refer to the chapters in Part III for instructions on how to implement features, including which parameters to use and which values to assign them.

Note – Duplicate parameters are allowed in the `ics.conf` file. The system takes the value of the last instance of the parameter in the file.

E.2.1 Tips for Customizations to the Configuration File

The configuration file is big. There are many parameters. If you make customizations, it can be difficult to find them or remember why you changed the values. To avoid confusion, add your customizations to the end of the file in a section you create for that purpose. For example, you can create a comment line with the following text: `! My ics.conf Changes`. Then add any new parameters or any parameters that you are modifying, and their values. Add comments to each parameter describing why the change was made, with the current date. This will give you a history of the changes made to the system for later reference.

Every time you start or restart Calendar Server, the system reads the entire configuration file. The more parameters the system must process, the longer it takes to start up the system. If there are many duplicate parameters, it can slow the process noticeably. To avoid this, comment out obsolete duplicate parameters.

E.2.2 Calendar Server Local Instance Configuration Parameters

The following table shows the configuration parameters starting with `local.`, showing each parameter's default value and description.

TABLE E–1 Local Server Instance Configuration Parameters in the `ics.conf` File

Parameter	Default Value	Description
local.autoprovision	`"yes"`	Enables (`"yes"`) or disables (`"no"`) autoprovisioning of the user's calendar.
local.domain.language	`"en"`	Default language for domains in this Calendar Server instance.
local.hostname	`" "`	Host name of the machine on which Calendar Server is installed.
local.installeddir	*cal-svr-base*`/SUNWics5/ cal`	Directory path location where Calendar Server is installed.
local.instancedir	*cal-svr-base*`/SUNWics5/ cal`	Directory path location where configuration files and data for this instance of Calendar Server are installed.
local.instance.lockdir.path	*cal-svr-base*`/SUNWics5/ cal/data/lock`	Specifies the location where lock files for this server instance are stored.
local.instance.counter.path	*cal-svr-base*`/SUNWics5/ cal/lib/counter`	Specifies the location where counter files for this server instance are stored.
local.plugindir.path	`" "`	Directory path location where CSAPI plug-ins for this instance of Calendar Server are installed.
local.rfc822header.allow8bit	`"no"`	Allow (`y`) or not allow (`n`) 8-bit headers in email messages sent by this server.

TABLE E–1 Local Server Instance Configuration Parameters in the `ics.conf` File *(Continued)*

Parameter	Default Value	Description
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.

E.2.3 Calendar Server LDAP Authorization Configuration Parameters

TABLE E–2 LDAP Authorization Configuration Parameters in the `ics.conf` File

Parameter	Default Value	Description
local.authldapbasedn	`" "`	Base DN for LDAP authentication. If not specified, *local.ugldapbasedn* is used.
local.authldaphost	`"localhost"`	Host for LDAP authentication. If not specified, *local.ugldaphost* is used.
local.authldapbindcred	`" "`	Bind credentials (password) for user specified in *local.authldapbinddn*.
local.authldapbinddn	`" "`	DN used to bind to LDAP authentication host to search for user's `dn`. If not specified, or if the value is `" "`, its an 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.

TABLE E–2 LDAP Authorization Configuration Parameters in the `ics.conf` File *(Continued)*

Parameter	Default Value	Description
local.authldapmaxpool	`"1024"`	Maximum number of LDAP client connections that are maintained for LDAP authentication. If not specified, *local.ugldapmaxpool* is used.

E.2.4 Calendar Server LDAP User and Group Search Configuration Parameters

TABLE E–3 LDAP Authorization Configuration Parameters in the `ics.conf` File

Parameter	Default Value	Description
local.lookupldap.search.minwildcardsize	`"3"`	Specifies the minimum string size for wildcard searches in an attendee lookup search. Zero (0) means always do a wildcard search.
local.ugldaphost	`"localhost"`	Host name of machine that stores the LDAP user preferences.

E.2.5 Calendar Server User Preferences Configuration Parameters

TABLE E–4 User Preferences Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
local.enduseradmincred	`" "`	Bind credentials (password) for LDAP user preferences authentication.
local.enduseradmindn	`" "`	DN used to bind to LDAP user preferences host. There is no default DN. If the value is " ", or not specified, anonymous bind is assumed.
local.ugldapbasedn	`" "`	Base DN for LDAP user preferences. Must be specified and cannot be blank.

TABLE E–4 User Preferences Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description

local.ugldapicsextendeduserprefs	`"ceColorSet,`
	`ceFontFace,`
	`ceFontSizeDelta,`
	`ceDateOrder,`
	`ceDateSeparator,`
	`ceClock,`
	`ceDayHead,`
	`ceDayTail,`
	`ceInterval,`
	`ceToolText,`
	`ceToolImage,`
	`ceDefaultAlarmStart,`
	`ceSingleCalendarTZID,`
	`ceAllCalendarTZIDs,`
	`ceDefaultAlarmEmail,`
	`ceNotifyEmail,`
	`ceNotifyEnable,`
	`ceDefaultView,`
	`ceExcludeSatSun,`
	`ceGroupInviteAll"`
		Values for the options in the *icsExtendedUserPrefs* attribute.

local.user.authfilter	`"uid=%u"`	Filter to use for user lookup.

E.2.6 Calendar Server Calendar Store Configuration Parameters

The following table shows the Calendar Store Configuration parameters with each parameter's default value and description. The Calendar Store holds all event and todo records.

TABLE E–5 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.

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 "15.4 Calendar Access Control" on page 296
		To specify Access Control Entries for one or more calendars using the command-line utilities, see "D.5 `cscal`" on page 409.

calstore.calendar.owner.acl	`"@@o^a^rsf^g;`
	`@@o^c^wdeic^g"`
		Specifies the default access control settings for owners of a calendar.

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. Although this parameter regulates the case of the original `calid`, it does not affect related LDAP attributes that set values for the owner, subscribed, and ACE permissions. Setting this parameter to "yes" may affect the use of Communications Express, since Communications Express relies on values set in the LDAP attributes such as icsCalendarOwned to determine the `calid`. Therefore, use caution if you intend to set this parameter to "yes."
calstore.default.timezoneID	`"America/` `New_York"`	Time zone ID to be used when: ■ A time zone ID is not supplied ■ A calendar time zone ID is not found ■ A user time zone ID is not found An invalid value causes the server to use to the GMT (Greenwich Mean Time) time zone.

TABLE E–5 Calendar Store Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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 size of an LDAP group that will be expanded for an invitation. A value of "0" means to expand the group without regard to size. A value of -1 means no expansion of LDAP groups allowed.
calstore.group.freebusy.maxsize	"0"	Maximum number of users present in any legitimate mailing list (group) used for calendar free/busy lookups. The default value allows an unlimited number of users. If a group contains an extremely large number of users, there can be performance issues and/or high memory consumption when doing a free/busy lookup.
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.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. Allowable values are: uid, cn, gid, res, mailto, cap.
calstore.unqualifiedattendee. *fmt2.type*	"mailto"	Specifies how Calendar Server treats strings with an at sign (@), such as jdoe@sesta.com, when performing a directory lookup for attendees of an event. Allowable values are: uid, cn, gid, res, mailto, cap.

TABLE E–5 Calendar Store Configuration Parameters in the `ics.conf` File *(Continued)*

Parameter	Default Value	Description
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. Allowable values are: `uid`, `cn`, `gid`, `res`, `cap`.
store.partition.primary.path	`"."`	Location of primary disk partition where calendar information is stored.

E.2.7 Calendar Log Information Configuration Parameters

The following table shows the calendar log configuration parameters with each parameter's default value and description.

TABLE E–6 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.access.logname	*"httpd.access"*	Name of the current HTTP access log file.
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: `CRITICAL`, `ALERT`, `ERROR`, `WARNING`, `NOTICE`, `INFORMATION`, and `DEBUG`.
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).

TABLE E–6 Calendar Log Configuration Parameters in the `ics.conf` File *(Continued)*

Parameter	Default Value	Description
logfile.minfree *diskspace*	`"5242880"`	Minimum free disk space (in bytes) that must be available for logging.
logfile.notify.logname	`"notify.log"`	Name of log file for the `csnotifyd` service.
logfile.rollovertime	`"86400"`	Number of seconds before log files are rotated.
logfile.store.logname	`"store.log"`	Store log file name.
service.http.commandlog	*"no"*	This parameter is for debugging only. If set to `"yes"`, the system logs all incoming commands into the `http.commands` log file. Do not use this during production runtime. It will fill up the log file very quickly and could cause performance degradation.

E.2.8 Calendar Server Administrator Configuration Parameters

The following table lists the `ics.conf` parameters that pertain to administrators.

TABLE E–7 Configuration Parameters for Administrators

Parameter	Default Value	Description
service.siteadmin. *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.siteadmin. *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.allowgetmodify* *userprefs*	`"no"`	Indicates whether the Calendar Server administrator can get and set user preferences using WCAP commands.
service.admin.ldap.enable	`"yes"`	If `"yes"`, enables LDAP for user authentication of the user specified in *service.siteadmin.userid*.

E.2.9 Calendar Server Services Configuration

The following table shows the various services configuration parameters with each parameter's default value and description.

TABLE E–8 Services Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
service.admin.alarm	"yes"	Enable ("yes") or disable ("no") alarm notifications for administration tools.
local.store. *checkpoint.enable*	"yes"	If "yes", start the csadmind database checkpoint thread.
service.admin. *dbcachesize*	"8388608"	Maximum cache size (in bytes) for Berkeley Database for administration sessions.
local.store. *deadlock.enable*	"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. *maxsessions*	"100"	Maximum number of administration sessions allowed.
service.admin. *maxthreads*	"10"	Maximum number of running threads per administration session.
service.admin. *numprocesses*	N/A	Maximum number of a concurrent administration processes allowed.
service.admin.port	N/A	**CAUTION** Set by the system. Do not change.
service.admin. *resourcetimeout*	"900"	Number of seconds before timing out an administration connection.

TABLE E–8 Services Configuration Parameters in the ics.conf File (Continued)

Parameter	Default Value	Description
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.
service.dcroot	"o=internet"	Root suffix of the DC tree in the directory.
service. *dnsresolveclient*	"no"	If "yes", client IP addresses are checked against DNS if allowed HTTP access.
service.plaintext *loginpause*	"0"	Number of seconds to delay after successfully authenticating a user using plain text passwords.
service.http.admins	"calmaster"	Space separated list of user ID's with administration rights to this Calendar Server.
service.http. *allowadminproxy*	"yes"	If "yes", allow login via proxy.
service.http. *allowanonymouslogin*	"yes"	If "yes", allow anonymous (no authentication) access. This is a special type of login that is allowed only specified, restricted access (usually read only access to public calendars).
service.http *.calendarhostname*	"" (Null)	HTTP host for retrieving HTML documents.
service.http.cookies	"yes"	Tells the server to whether or to support cookies. It must be set to "yes" to enable single sign-on.

TABLE E–8 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.http. *dbcachesize*	"8388608"	Maximum cache size of Berkeley DB for HTTP sessions.
service.http. *domainallowed*	"" (Null)	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). EXCEPT Operator You can use the EXCEPT operator to create exceptions to matching names or patterns when you have multiple entries in a list. For example, the expression: list1 EXCEPT list2 means that anything that matches list2 is matched, unless it also matches list2. For example: ALL: ALL EXCEPT isserver.siroe.com If this were a Deny filter, it would deny access to all services to all clients except those on the host machine, isserver.siroe.com. EXCEPT clauses can be nested. The expression: list1 EXCEPT list2 EXCEPT list3 is evaluated as if it were list1 EXCEPT (list2 EXCEPT list3)
service.http. *domainnotallowed*	"" (Null)	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). EXCEPT Operator. For details about this operator, see the preceding entry, service.http.domainallowed.
service.http. *attachdir.path*	"."	Directory location relative to *local.queuedir* (or an absolute path if specified) where imported files are temporarily stored.
service.http. *ipsecurity*	"yes"	If "yes", all requests that reference an existing session are verified as originating from the same IP address.
service.http.enable	"yes"	If "yes", start the cshttpd service when starting all services and stop cshttpd when stopping all services.

TABLE E–8 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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.listenaddr	"INADDR_ANY"	Specifies the TCP address that HTTP services will listen on for client requests. "INADDR_ANY" indicates any address.
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 (cshttpd) processes that should run on a server. For a server that has multiple CPU's, see "21.8 Using Load Balancing Across Multiple CPU's" on page 355
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		Directory relative to executable where all URL references to files are stored. The default is: /var/opt/SUNWics5/tmp
service.http.uidir.path	"html"	Directory that contains the default calendar client. If allowing only WCAP access, set to "".

TABLE E–8 Services Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.ldapmemcache	"no"	If "yes", use cache in LDAP SDK.
service.ldapmemcachettl	"30"	If *service.ldapmemcache* is "yes", pass in this value to the LDAP SDK. This is the maximum number of seconds that an item can be cached. If 0, there is no limit to the amount of time that an item can be cached.
service.ldapmemcachesize	"131072"	If *service.ldapmemcache* is "yes", pass in this value to the LDAP SDK. This is the maximum amount of memory in bytes that the cache will consume. If 0, the cache has no size limit.
service.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.freebusy. *redirecturl*	““	For migration purposes, when migration is only partially done and calendars are split between the originating database and the Calendar Server target database. The URL of the originating database to look in, if a calendar is not found in the Calendar Server database.
service.wcap.allow *createcalendars*	"yes"	If "yes", allow calendars to be created.
service.wcap.allow *deletecalendars*	"yes"	If "yes", allow calendars to be deleted.
service.wcap.allow *changepassword*	"no"	If "yes", allows user passwords to be changed.
service.wcap.allow *publicwritablecalendars*	"yes"	If "yes", allow users to have publicly writable calendars.
service.wcap.allow *setprefs.cn*	"no"	If "yes", allow the set_userprefs.wcap attribute to modify the user preference cn (LDAP user's common name).

TABLE E–8 Services Configuration Parameters in the `ics.conf` File *(Continued)*

Parameter	Default Value	Description
service.wcap.allow *setprefs.givenname*	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `givenname` (LDAP user's given name).
service.wcap.allow *setprefs.icsCalendar*	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `icsCalendar` (a user's default calendar identifier).
service.wcap.allow *setprefs.mail*	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `mail` (user's email 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.allow *setprefs.sn*	`"no"`	If `"yes"`, allow the `set_userprefs.wcap` attribute to modify the user preference `sn` (LDAP user's surname).
service.wcap.allow *setprefs.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 publicly readable and privately writable. If `"no"`, default user calendars are initially set to privately readable and writable.
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.

E.2.10 Calendar Server SSL Configuration Parameters

The following table shows the `ics.conf` SSL Configuration parameters with each parameter's default value and description. While most of the SSL parameters take the default values, two of the parameters require you to change the value from the system default to the SSL value, as follows:

- `service.http.ssl.usessl="yes"`
- `service.http.ssl.port.enable="yes"`

The table that follow shows the `ics.conf` parameters and their default settings. Verify that your `ics.conf` parameters have the appropriate values:

TABLE E–9 Configuration Parameters for SSL

Parameter	Default Value	Description
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.
service.http.tmpdir	`/var/opt/SUNWis5/tmp` doma	Specifies a temp directory.
service.http.uidir. *path*	`"html"`	Specifies directory where the UI files are found.
service.http.ssl. *cachedir*	`"."`	Specifies the physical path location for the SSL cache.
service.http.ssl. *cachesize*	`"10000"`	Specifies the maximum size of the SSL cache database.
service.http.ssl. *usessl*	`"no"`	For SSL configuration, change this value to `"yes"`. Specifies whether the `cshttpd` process should use the SSL subsystem.
service.http.ssl. *port.enable*	`"no"`	For SSL configuration, change this value to "yes". **Note –** This does not disable the HTTP process from listening to its port. There is no way to actually disable HTTP, but you can assign it to another port that is non-functional. Do *not* set `service.http.enable="no"`. That would disable the HTTPS process also.
service.http.ssl. *port*	`"443"`	Specifies the SSL port number where the `cshttpd` process listens for HTTPS requests from Calendar Server users. Do not set this to the same default port used by HTTP (`"80"`).
service.http.ssl. *securesession*	`"yes"`	Specifies whether to encrypt the entire session.

TABLE E–9 Configuration Parameters for SSL *(Continued)*

Parameter	Default Value	Description
local.ssldbpath	`"/etc/opt/SUNWics5/config"`	Specifies the physical path location of the SSL Certificate Database.
service.http.ssl.certdb.password — This parameter was removed from the `ics.conf` file. It was replaced by a configuration file.	`sslpassword.com`	`sslpassword.conf` is a text file that contains the certificate database password. This file is used by the `certutil` utility but not by Calendar Server. Create `sslpassword.conf` in the following directory: `/etc/opt/SUNWics5/config`
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*	`"rsa_rc4_40_md5, rsa_rc2_40_md5, rsa_des_sha, rsa_rc4_128_md5, rsa_3des_sha"`	Specifies a list of supported or valid SSL ciphers.
service.http.ssl. *ssl3.sessiontimeout*	`"0"`	Specifies the timeout value for the SSL session.

E.2.11 Calendar Server Domain Configuration Parameters

The following table shows the domain configuration parameters with each parameter's default value and description.

TABLE E–10 Configuration Parameters for Multiple Domain Support

Parameter	Default Value	Description
local.domain.language	`"en"`	Default language for domains in this instance of Calendar Server.
local.schemaversion	`"1"`	Specifies the version of the LDAP schema: ■ "1" Sun LDAP Schema version 1. See also *service.dcroot* ■ "2" Sun LDAP Schema version 2. See also *service.schema2root*

TABLE E–10 Configuration Parameters for Multiple Domain Support *(Continued)*

Parameter	Default Value	Description
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.
service.siteadmin.cred	`" "`	Specifies the password of the domain administrator.
service.virtualdomain. *support*	`"yes"`	Enables (`"yes"`) or disables (`"no"`) support for multiple domains. **Caution –** Do not change this parameter to `"no"`. Calendar Server supports multiple domains by default.

E.2.12 Configuration Parameters to Enable Email Notifications

The following three parameters enable or disable the system to send out notifications for cancellations, invitations and replies.

TABLE E–11 Alarm Notification Configuration Parameters in the `ics.conf` File

Parameter	Default Value	Description
ine.cancellation.enable	`"yes"`	Determines whether email notifications are sent to attendees when an event is cancelled.
ine.invitation.enable	`"yes"`	Determines whether email notifications are sent to attendees invited to an event.
ine.reply.enable	`"yes"`	Determines whether email notifications are sent to the organizer when an attendee replies to an invitation.

E.2.13 Alarm Notification Parameters

The following table shows the alarm notification server configuration parameters with each parameter’s default value and description.

TABLE E–12 Alarm Notification Configuration Parameters in the `ics.conf` File

Parameter	Default Value	Description
alarm.diskstat. *msgalarmdescription*	"*percentage calendar partition disk space 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.diskstat.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.msgalarmnoticehost	`"localhost"`	The host name of the SMTP server used to send server alarms.
alarm.msgalarmnoticeport	`"25"`	The SMTP port used to send server alarms.
alarm.msgalarmnoticercpt	`"Postmaster @localhost"`	The email address to whom server alarms sent.
alarm.msgalarmnoticesender	`"Postmaster @localhost"`	The email address used as the sender when the server sends alarms.
alarm.msgalarmnoticetemplate	`""`	The default format used to send email alarms: `"From: %s\nTo: %s\nSubject: ALARM: %s of \"%s\" is 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 there is no service response.

TABLE E–12 Alarm Notification Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
alarm.responsestat. *msgalarmthresholddirection*	"-1"	Specifies whether *alarm.responsestat.* *msgalarmthreshold* is the percentage above or below the threshold. A value of -1 is below, and a value of 1 is above.
alarm.responsestat. *msgalarmwarninginterval*	"24"	Number of hours between sending warning messages about no service response sent out.

E.2.14 Calendar Lookup Database Configuration

The following table shows the Calendar Lookup Database (CLD) parameters with each parameter's default value and description.

TABLE E–13 Calendar Lookup Database (CLD) Parameters in the ics.conf File

Parameter	Default Value	Description
csapi.plugin.calendarlookup	"no"	Enable ("yes") or disable ("no") 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"	Use "local" for a machine where everything is on the same machine, or for a machine that functions only as a back-end machine. Use "directory" for machines that are only a front-end, or for a machine which functions as both the front-end and back-end.
caldb.dwp.server.default	" "	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 the LDAP entry for a user logging into Calendar Server (login.wcap), does not have an icsDWPHost attribute, Calendar Server uses the value of this parameter to add the attribute. If a user LDAP entry 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.

TABLE E–13 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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.
caldb.cld.cache.mempoolsizemb	"4"	Specifies the size in megabytes of shared memory.
caldb.cld.cache.maxthread	"1000"	Specifies the maximum number of database threads.
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 `/var/opt/SUNWics5/csdb/cld_cache` directory.
caldb.cld.cache.checkpointinterval	"60"	Specifies the number of seconds between checkpoints.
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	" "	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.

TABLE E–13 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
caldb.dwp.md5	"no"	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.) ■ "no" disables MD5 hash checking. ■ "yes" 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.
caldb.dwp.server.hostname.port	"59779"	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	none	Base DN to authenticate against if LDAP plug-in is used for the calendar locate mechanism.
caldb.cld.directory.ldaphost	none	Host name of the LDAP server to access if an LDAP plug-in is used for the calendar locate mechanism.
caldb.cld.directory.ldapbindcred	none	Bind credentials (password) for the user specified in the setting *local.authldapbinddn* if an LDAP plug-in is used for the calendar locate mechanism.
caldb.cld.directory.ldapbinddn	none	DN used to bind to for authentication to search for user's DN if an LDAP plug-in is used for the calendar locate mechanism.
caldb.cld.directory.ldapport	"389"	Port number of the LDAP server to access if an LDAP plug-in is used for the calendar locate mechanism.
csapi.plugin.authentication	"no"	If "yes", 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 "no" and *csapi.plugin.authentication* is "yes", only load this specific plug-in. If not specified or blank (" "), load all authentication class plug-ins.
logfile.dwp.buffersize	"0"	Size of Database Wire Protocol service log buffers (in bytes).

TABLE E–13 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
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.
logfile.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.

TABLE E–13 Calendar Lookup Database (CLD) Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
service.dwp.maxthreads	"1000"	Maximum number of concurrently running Database Wire Protocol service threads.
service.dwp.numprocesses	"1"	Maximum number of concurrently running Database Wire Protocol (DWP) service (csdwpd) processes that should run on a server. For a server that has multiple CPU's, see "21.8 Using Load Balancing Across Multiple CPU's" on page 355
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").

E.2.15 Calendar Server LDAP Data Cache Configuration Parameters

The following table describes the configuration parameters in the ics.conf file for the LDAP data cache.

TABLE E–14 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 old cache files. The default is "yes".
local.ldap.cache.homedir.path	Specifies the physical location of LDAP data cache database. The default is /var/opt/SUNWics5/csdb/ldap_cache.

TABLE E–14 LDAP Data Cache Configuration Parameters *(Continued)*

Parameter	Description
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 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).

E.2.16 Group and Resource Calendar Configuration Parameters

To configure group and resource calendars, use the parameters found in the following table.

TABLE E–15 Configuration Parameters for Resource Calendars

Parameter	Default Value	Description
Use the following parameters only for groups:		
local.lookupldap searchattr.groupid	"groupid"	This is a group's unique identifier. Similar to a uid for a user.
group.allow.doublebook	"no"	Determines if a group 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 group calendar is created. After a group calendar is created, Calendar Server checks the calendar properties (ics50calprops.db) to determine if double booking is allowed.

TABLE E–15 Configuration Parameters for Resource Calendars *(Continued)*

Parameter	Default Value	Description
`group.default.acl`	see the description at right	The default access control list for group calendars. The default is: "`@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g`"
`group.invite.autoprovision`	"`yes`"	Specifies whether a group calendar should be created when an invitation is sent to a group that does not yet have a default calendar.
`group.invite.autoaccept`	"`no`"	Specifies whether invitations to a group should be auto-accepted.
`group.invite.expand`	"`yes`"	Specifies whether a group should be expanded when invited, such that all members show on the invitation.
Use the following parameters only for resources:		
`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`		Specifies the default access control permissions used when a resource calendar is created. The default is: "`@@o^a^r^g;@@o^c^wdeic^g;@^a^rsf^g`"
`resource.invite.autoprovision`	"`yes`"|"`no`"	Specifies whether resource calendars should be created automatically when an invitation is sent to a resource that does not have a default calendar yet.
`resource.invite.autoaccept`	"`yes`"|"`no`"	Specifies whether invitations sent to resources should be automatically accepted.
Use the following parameters for both groups and resources:		
`local.lookupldap searchattr.owner`	"`owner`"	Attribute to use for groups and resource owners. Default is "`owner`". Same attribute used as the default for both groups and resources; changing it for one changes it for the other.

TABLE E–15 Configuration Parameters for Resource Calendars *(Continued)*

Parameter	Default Value	Description
local.lookupldap searchattr.coowner		Attribute to use for groups and resource owners. Same attribute used as the default for both groups and resources; changing it for one changes it for the other. Default is "icsSecondaryowners".
local.lookupldap searchattr.defaultacl		Attribute used to hold the default access control strings for groups and resources. Default is "icsDefaultacl"
local.lookupldap searchattr.doublebook		Attribute used at autoprovisioning of group and resource calendars to specify whether more than one event can be scheduled for the same time slot. Default is "icsDoublebooking"
local.lookupldap searchattr.autoaccept		Attribute used at autoprovisioning of group and resource calendars to specify whether events will automatically be accepted. Default is "icsAutoaccept"
local.lookupldap searchattr.timezone		Attribute used at autoprovisioning of group and resource calendars to specify the time zone the calendar will use. Default is "icsTimezone"

E.2.17 Calendar Server Single Sign-on (SSO) Configuration Parameters

- “E.2.17.1 Configuring SSO Through Access Manager” on page 483
- “E.2.17.2 Configuring SSO Through Communications Servers Trusted Circle Technology” on page 484

E.2.17.1 Configuring SSO Through Access Manager

The following table shows the SSO configuration parameters with each parameter’s default value and description when you are using Access Manager.

TABLE E–16 SSO Configuration Parameters in the ics.conf File (Through Access Manager)

Parameter	Default	Description
local.calendar.sso. singlesignoff	”yes”	Enables (”yes”) or disables (”no”) SSO for Calendar Server.

TABLE E–16 SSO Configuration Parameters in the ics.conf File (Through Access Manager) *(Continued)*

Parameter	Default	Description
`local.calendar.sso.` `amcookiename`	`“iPlanetDirectoryPro”`	Specifies the name of the Access Manager SSO cookie.
`local.calendar.sso.` `amnamingurl`	”http://*AccessManager*:*port* /amserver/namingservice”	Specifies the URL of the Access Manager SSO naming service.
`local.calendar.sso.` `amloglevel`	`”3”`	Specifies the log level for Access Manager SSO. Range is from 1 (quiet) to 5 (verbose).
`local.calendar.sso.` `logname`	`“am_sso.log”`	Specifies the name of the Access Manager SSO API log file.

E.2.17.2 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 E–17 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`	none	Verification URL for the value specified for `sso.appid`. For example: `"sso.ics50.url="http://siroe.com:80/ default.html"`
`sso.nnn.ip`	none	IP address of the value specified for `sso.appid`. For example: `sso.ics50.ip=` `"123.12.456.123"`

TABLE E–17 SSO Configuration Parameters in the ics.conf File Using Communications Servers Trusted Circle Technology *(Continued)*

Parameter	Default Value	Description
`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.
`sso.appid.url`	" "	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* identifies 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?`"

E.2.18 Calendar Server Group Scheduling Engine (GSE) Configuration Parameters

The following table shows the Group Scheduling Engine (GSE) configuration parameters with each parameter’s default value and description.

TABLE E–18 Group Scheduling Engine (GSE) Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
gse.autorefresh replystatus	"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.belowthreshold timeout	"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.retryexpired interval	"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.

E.2.19 Calendar Server Berkeley Database Configuration Parameters

The following table lists the ics.conf parameters used to configure database handling, and gives each parameter's default value and description.

TABLE E–19 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. Do not set this to "no" unless you have enabled automatic backups.
`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: `/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 */etc*`/opt/SUNWics5/config` that contains the localized version of the files used to format email notifications. For example, "en" specifies the directory for the English localized version, and "fr" specifies the directory for the French localized version.
`caldb.smtpport`	"25"	Port for SMTP host.

TABLE E–19 Database Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`local.caldb.deadlock.autodetect`	"no"	Periodically checks if the Berkeley database is in a deadlock state and, if so, instructs the database to reset.

E.2.20 Automatic Backups of the Calendar Database

The following table lists the parameters used by the automatic backup process (`csstored`), gives the default value where available, and describes the `ics.conf` parameter.

TABLE E–20 ics.conf Parameters Used by the Automatic Backup Process

Parameter	Default Value	Description
`logfile.store.logname`	*defaultstore*`.log`	Name of the log file.
`logfile.logdir`	"."	Path to the log directory.
`caldb.berkeleydb.homedir.` `path`	none	Path to live database.
`caldb.berkeleydb.archive.` `path`	none	Path to the archive backup.
`caldb.berkeleydb.hotbackup.` `path`	none	Path to the hot backup.
`caldb.berkeleydb.archive.` `enable`	"yes"	Enable/disable automatic archive backups.
`caldb.berkeleydb.hotbackup.` `enable`	"yes"	Enable/disable automatic hot backups.
`caldb.berkeleydb.hotbackup.` `mindays`	"3"	Minimum number of hot backup copies kept on disk.
`caldb.berkeleydb.hotbackup.` `threshold`	"70"	Percentage of used disk space that triggers purging of old hot backup copies.
`caldb.berkeleydb.archive.` `interval`	"86400"	Interval in seconds between backups. Default is 24 hours, which is 86400 seconds.
`caldb.berkeleydb.archive.` `mindays`	"3"	Minimum number of archive backup copies kept on disk.

TABLE E–20 ics.conf Parameters Used by the Automatic Backup Process *(Continued)*

Parameter	Default Value	Description
caldb.berkeleydb.archive.maxdays	"7"	Maximum number of archive backup copies kept on disk.
caldb.berkeleydb.archive.threshold	"70"	Percentage of used disk space that triggers purging of old archive backup copies.
caldb.berkeleydb.circularlogging	"yes"	Enables/disables management of the number of, and disk space occupied by, backup copies.
caldb.berkeleydb.archive.interval	"86400"	Time in seconds between backups. The default is 24 hours, or 86400 seconds.
service.store.enable (not included in ics.conf file)	"yes"	Enables csstored to be started by start-cal. Must be added to the ics.conf file, set to "no", if you want to disable csstored from being started by start-cal.

E.2.21 Calendar Database Parameters for ENS Messages

The following table describes the parameter, default value, and description for each of the parameters used to configure the calendar database. If you require a value other than the default, you must set it in the ics.conf file.

TABLE E–21 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 alarm 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.

TABLE E–21 Calendar Database Parameters for ENS Messages *(Continued)*

Parameter	Default Value	Description
`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.
`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"`.

TABLE E–21 Calendar Database Parameters for ENS Messages *(Continued)*

Parameter	Default Value	Description
`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"`.

E.2.22 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.

The following table shows the Event Notification Server (ENS) configuration parameters in `ics.conf`. with each parameter's default value and description.

TABLE E–22 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 ENS is running.
service.ens.port	"57997"	The port number of the machine on which ENS is running.
service.ens.library	"xenp"	The name of ENS 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 ENS. A value of "-1" causes the alarm thread to retry indefinitely.
service.notify. retryinterval	"3"	Number (in seconds) that csnotifyd waits before attempting to recontact ENS after a connection failure.
service.notify. startupretrytime	"0"	Total number of seconds Calendar Server keeps trying to contact ENS 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. A value of "0" tells the alarm thread to exit immediately if it fails to connect to ENS 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 E–22 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:
		■ If `"yes"`, the system differentiates between `reply`, `refresh`, or `modify` transactions by publishing to the respective topic:
		`caldb.berkeleydb.ensmsg.replyevent`
		`caldb.berkeleydb.ensmsg.refreshevent`
		`caldb.berkeleydb.ensmsg.modifyevent`
		■ If `"no"`, the system publishes all three types (`reply`, `refresh`, `modify`) to the following topic:
		`caldb.berkeleydb.ensmsg.modifyevent`

`caldb.berkeleydb.ensmsg.createevent`	`"no"`	If `"yes"`, create an ENS 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 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 ENS message for the refresh of an event.
`caldb.berkeleydb.ensmsg.replyevent`	`"no"`	Specifies 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 E–22 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 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 ENS's alarm thread waits for an acknowledgment from csnotifyd after publishing an alarm notification. If the timeout 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: ■ If "ens", the server uses the external ENS to send and receive alarms. ■ If "smtp", the server sends alarms as standard SMTP messages and to bypass ENS.
`caldb.serveralarms.` `initthreads`	"10"	Initial number of server alarm threads.
`caldb.serveralarms.` `maxretrytime`	"-1"	How many times the alarm thread will consecutively retry and fail to contact ENS. "-1" causes the alarm thread to retry indefinitely.
`caldb.serveralarms.` `maxthreads`	"10"	Maximum number of server alarm threads.

TABLE E–22 Event Notification Server (ENS) Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
caldb.serveralarms. retryinterval	"5"	Number (in seconds) that the alarm thread (in csadmind) waits before attempting to recontact ENS.
caldb.serveralarms. stacksize	"65536"	Stack frame size for server alarm threads.
caldb.serveralarms.startup retrytime	"0"	Total number of seconds Calendar Server keeps trying to contact ENS 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. If "0", the alarm thread exits immediately if it fails to connect to ENS at startup.
caldb.smtphost	"localhost"	Send alarm emails to this SMTP host.

E.2.23 Calendar Server API Configuration

The following table shows the Calendar Server API (CSAPI) configuration parameters with each parameter's default value and description.

TABLE E–23 CSAPI Configuration Parameters in the ics.conf File

Parameter	Default Value	Description
csapi.plugin.authentication	"no"	If the value is "yes", load only the plug-in specified in csapi.plugin.authentication.name.
csapi.plugin.accesscontrol	"no"	Enable ("yes") or disable ("no") Access Control plug-in.
csapi.plugin.authentication	"no"	If the value is "yes", load only the plug-in specified in csapi.plugin.authentication.name. If the value is "no", 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 "no" and csapi.plugin.authentication is "yes", only load this specific plug-in. If not specified or blank (" "), load all authentication class plug-ins.

TABLE E–23 CSAPI Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
csapi.plugin.database	"yes"	If "yes", load only the plug-in specified in csapi.plugin.database.name. If "no", or if not specified, load all database plug-ins in alphabetical order.
csapi.plugin.database.name	"cs_caldb _berkeley"	If csapi.plugin.loadall is "no" and csapi.plugin.database is "yes", load only this plug-in. If not specified or blank (" "), load all database plug-ins in alphabetical order.
csapi.plugin.datatranslator	"yes"	If "yes", 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_data translatorcsv"	If csapi.plugin.loadall is "no" and csapi.plugin.datatranslator is "yes", load this specific plug-in. If blank (" "), or not specified, load all data translator class plug-ins.
csapi.plugin.dbtranslator	"yes"	Enable ("yes") or disable ("no") database-to-output format plug-ins.
csapi.plugin.dbtranslator.name	"*"	If csapi.plugin.dbtranslator is "yes", then either: ■ If "*", load all the database-to-output format plug-ins. ■ If this value is a library name, load only this specific plug-in. If csapi.plugin.dbtranslator is "no", this setting is ignored.
csapi.plugin.loadall	"no"	If "yes", load all plug-ins found in the plug-ins directory. (Plug-ins have an .so extension.) If "no", 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	"no"	If "yes", 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.

TABLE E–23 CSAPI Configuration Parameters in the ics.conf File *(Continued)*

Parameter	Default Value	Description
`csapi.plugin.userprefs.` `name`	" "	If `csapi.plugin.loadall` is "no" and `csapi.plugin.userprefs` is "yes", this parameter is used. If not specified or blank (" "), load all user preferences class plug-ins. Otherwise, only load this specific plug-in.

E.3 Counters Configuration (counter.conf) File

Calendar Server counters (statistics) configuration parameters are in the following file:

/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 Appendix D, "Calendar Server Command-Line Utilities Reference."

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:

- "E.3.1 Alarm Counters" on page 497
- "E.3.8 Server Response Counters" on page 500
- "E.3.4 Group Scheduling Counters" on page 499
- "E.3.3 HTTP Counters" on page 498
- "E.3.7 Database Counters" on page 500
- "E.3.6 WCAP Counters" on page 500
- "E.3.2 Disk Usage Counters" on page 498
- "E.3.9 Session Status Counters" on page 501
- "E.3.5 Authentication Counters" on page 499

E.3.1 Alarm Counters

The following table shows each alarm counter's name, type, size, and description.

TABLE E–24 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.

E.3.2 Disk Usage Counters

The following table shows each disk usage counter's name, type, size, and description.

TABLE E–25 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.

E.3.3 HTTP Counters

The following table shows each HTTP counter's name, type, size, and description.

TABLE E–26 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.

E.3.4 Group Scheduling Counters

The following table shows each Group Scheduling Engine (GSE) counter's name, type, size, and description.

TABLE E–27 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.

E.3.5 Authentication Counters

The following table shows each Authentication counter's name, type, size, and description.

TABLE E–28 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.

E.3.6 WCAP Counters

The following table shows each WCAP counter's name, type, size, and description.

TABLE E–29 WCAP (wcapstat) Counters in the counter.conf File

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

E.3.7 Database Counters

The following table shows each Database counter's name, type, size, and description.

TABLE E–30 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.

E.3.8 Server Response Counters

The following table shows each Server Response counter's name, type, size, and description.

TABLE E–31 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.

E.3.9 Session Status Counters

The following table shows each Session Status counter's name, type, size, and description.

TABLE E–32 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.

E.4 Calendar Server Email Notifications

Calendar Server sends the types of email messages described in "E.4.1 Calendar Server Email Notifications Configuration Parameters and Format Files" on page 502. 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 locale (such as `/en` for English and `/fr` for French) in the following directory:

/etc`/opt/SUNWics5/config`

For example, the English version of the task (todo) alarm message format is specified in the file:

/etc`/opt/SUNWics5/config/en/mail_todoalarm.fmt`

This section describes:

- "E.4.1 Calendar Server Email Notifications Configuration Parameters and Format Files" on page 502
- "E.4.2 Calendar Server Special Character Sequences for Event Notifications" on page 504
- "E.4.3 Calendar Server Notifications Date Sub-Format Strings" on page 505
- "E.4.4 Calendar Server Notifications Conditional Printing Format" on page 506
- "E.4.5 Special Character Sequences for Task Notifications" on page 507
- "E.4.6 Special Character Sequences for Dates" on page 508
- "E.4.7 Simple Event Reminder Example" on page 509
- "E.4.8 Complex Event Reminder Example" on page 511

E.4.1 Calendar Server Email Notifications Configuration Parameters and Format Files

The following table shows the message type, `ics.conf` parameter name, default format file description, and recipient for each Calendar Server Mail parameter.

TABLE E–33 Calendar Server Email Formats in the ics.conf File

Message Type	Parameter	Format File (default)	Description	Recipients
Event Publish	`calmail.imipeventpublish.` `fname`	`"mail_eventpublish.fmt"`	Announces an event or a change to an existing event	Those listed in Notification
Event Cancel	`calmail.imipeventcancel.` `fname`	`"mail_eventcancel.fmt"`	Announces an event cancellation	Those listed in Notification
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
Recurring Event Notification	`calmail.imipevent` `notificationrecur.fname`	`"mail_event` `notificationrecur.fmt"`	Notifies of a recurring event	Those listed in Notification
Event Cancel Notification	`calmail.imipeventcancel` `notification.fname`	`"mail_eventcancel` `notification.fmt"`	Notifies of a cancelled event	Those listed in Notification
Recurring Event Cancel Notification	`calmail.imipeventcancel` `notificationrecur.fname`	`"mail_eventcancel` `notificationrecur.fmt"`	Notifies of a cancelled recurring event	Those listed in Notification
Attendee Reply: Accept Notification	`calmail.imipeventaccept` `notification.fname`	`"mail_eventaccept` `notification.fmt"`	Notifies the event organizer that an attendee accepted the invitation.	Event Organizer
Attendee Reply: Decline Notification	`calmail.imipeventdecline` `notification.fname`	`"mail_eventdecline` `notification.fmt"`	Notifies the event organizer that an attendee declined the invitation.	Event Organizer

TABLE E–33 Calendar Server Email Formats in the ics.conf File *(Continued)*

Message Type	Parameter	Format File (default)	Description	Recipients
Attendee Reply: Tentative Accept Notification	`calmail.imipeventtentative acceptnotification.fname`	`"mail_eventtentative acceptnotification.fmt"`	Notifies the event organizer that an attendee has tentatively accepted the invitation.	Event Organizer
Attendee Reply: Accept Notification for Recurring Event	`calmail.imipeventaccept notificationrecur.fname`	`"mail_eventaccept notificationrecur.fmt"`	Notifies the event organizer that an attendee has accepted an invitation to a recurring event.	Event Organizer
Attendee Reply: Decline Notification for Recurring Event	`calmail.imipeventdecline notificationrecur.fname`	`"mail_eventdecline notificationrecur.fmt"`	Notifies the event organizer that an attendee declined an invitation to a recurring event.	Event Organizer
Attendee Reply: Tentative Accept Notification for a Recurring Event	`calmail.imipevent tentativeaccept notificationrecur.fname`	`"mail_eventtentative acceptnotificationrecur.fmt"`	Notifies the event organizer that an attendee tentatively accepted an invitation to a recurring event.	Event Organizer
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:

- "E.4.2 Calendar Server Special Character Sequences for Event Notifications" on page 504
- "E.4.5 Special Character Sequences for Task Notifications" on page 507
- "E.4.6 Special Character Sequences for Dates" on page 508

E.4.2 Calendar Server Special Character Sequences for Event Notifications

The following table shows the format code and meaning for special character sequences used in event notifications.

TABLE E–34 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 ISO 8601 date strings listing dates to exclude)
%a	rdates in iCalendar format (semicolon-separated list of ISO 8601 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 parameter has a value equal to the date, only the month/day/year portion of the date is output. If the end time has the same month/day/year value as the start time, only the start time is generated.
%C	Create time
%c	Event class
%d	Event description. (also see %F)
%E	End time (also see %Z)

TABLE E–34 Special Character Sequences for Event Notifications *(Continued)*

Format Code	Meaning
%e	Exception rules in iCalendar format
%F	Event description - folded line, iCalendar format (also see %d)
%G	The event's geographic location (latitude and longitude)
%g	Organizer's email address. (There is no guarantee as to the authenticity of this value.)
%K	Organizer email in the form of a `mailto:url`
%k	Alarm count
%L	Location
%l	Recurrence rules in iCalendar format
%M	Modify time
%N	New line
%n	The current time stamp used with `DTSTAMP`
%P	Priority
%r	Recurrence id (blank if this event does not recur)
%S	Event sequence number
%s	Summary
%t	Event status
%U	Unique Event Identifier
%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
%	Specifies a sub-format for the data identified by code. (For details, see "E.4.3 Calendar Server Notifications Date Sub-Format Strings" on page 505.)

E.4.3 Calendar Server Notifications Date Sub-Format Strings

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 "E.4.6 Special Character Sequences for Dates" on page 508). If you were only interested in the month and year of the start time, instead of %B, you would use: %(%m %Y)B.

E.4.3.1 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
```

E.4.4 Calendar Server Notifications Conditional Printing Format

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
```

E.4.5 Special Character Sequences for Task Notifications

The following table shows the format code and meaning for Special Character Sequences for Task Notifications.

TABLE E–35 Special Character Sequences for Task Notifications

Format Code	Meaning
%A	exdates in iCalendar format (semicolon-separated list of ISO 8601 date strings listing dates to exclude)
%a	rdates in iCalendar format (semicolon-separated list of ISO 8601 date strings listing recurrence dates)
%B	start time (also see %Z)
%C	create time
%c	task class
%D	due date and time.
%d	task description. (also see %F)
%E	due date and 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.

TABLE E–35 Special Character Sequences for Task Notifications *(Continued)*

Format Code	Meaning
`%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
`%t`	the status
`%U`	the `UID`
`%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 "E.4.3 Calendar Server Notifications Date Sub-Format Strings" on page 505)

E.4.6 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 rewrite any of the codes, but simply uses the operating system implementation.

TABLE E–36 Special Character Sequences for Dates

Format Code	Meaning
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for locale
%d	Day of month as decimal number (01 - 31)
%H	Hour in 24 hour format (00 - 23)
%I	Hour in 12 hour format (01 - 12)
%j	Day of year as decimal number (001 - 366)
%m	Month as decimal number (01 - 12)
%M	Minute as decimal number (00 - 59)
%p	Current locale's A.M./P.M. indicator for 12 hour clock
%S	Second as decimal number (00 - 59)
%U	Week of year as decimal number, with Sunday as first day of week (00 - 53)
%w	Weekday as decimal number (0 - 6; Sunday is 0)
%W	Week of year as decimal number, with Monday as first day of week (00 - 53)
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without century, as decimal number (00 - 99)
%Y	Year with century, as decimal number
%Z	Time-zone name or abbreviation; no characters if time zone is unknown
%%	Percent sign

E.4.7 Simple Event Reminder Example

The following example shows the default event reminder message format:

```
EVENT REMINDER
~~MIME-Version: 1.0%N
~~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 "E.4.3 Calendar Server Notifications Date Sub-Format Strings" on page 505.
- 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.
```

E.4.8 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--
```

Index