

# Sun™ Calendar Server 1.0 Installation and Administration Guide

---



THE NETWORK IS THE COMPUTER™

A Sun Microsystems, Inc. Business  
901 San Antonio Road  
Palo Alto, CA 94303 USA  
415 960-1300 fax 415 969-9131

Part No.: 805-7984-10  
Revision A, December 1998

Copyright 1998, 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Solaris, HotJava, Java, OpenWindows, and Sun Internet Mail Server are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**RESTRICTED RIGHTS:** Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

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

Copyright 1998, 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Solaris, HotJava, Java, OpenWindows, et Sun Internet Mail Server sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please  
Recycle



Adobe PostScript

# Contents

---

## **Preface**   vii

Who Should Use This Book   viii

How This Book Is Organized   viii

Related Information   ix

What Typographic Changes Mean   ix

Shell Prompts in Command Examples   x

## **1. Installing Sun Calendar Server 1.0**   1

System Requirements   1

Installing Sun Calendar Server 1.0   2

Installation Utility   2

Installing Sun Calendar Server 1.0   2

Troubleshooting Install   5

## **2. Administering Sun Calendar Server 1.0**   7

Server Administration   7

Starting Sun Calendar Server 1.0   7

Adding and Deleting Calendars   8

Migrating calendars from rpc.cmsd to Sun Calendar Server   9

Backing up and Restoring Calendars   10

Calendar Administration	11
Using Access Control Lists (ACL)	11

### **3. Calendaring Clients 13**

Accessing Sun Calendar Server 1.0 through Client Software	13
CDE Calendaring Client (dtdcm)	13
Web Access for Sun Calendar Server	14
Client Requirements for Running Web Access	14
Installing Web Access	15
Uninstalling Web Access	18
Starting and Stopping Web Access	18
Invoking Web Access	19
Troubleshooting	19
Post-installation Sun Web Access Setup	20
Changing the Port Number	21
Establishing Secure Service	22
Extending the Web Access Palette	23
Using An Alternate JVM for Web Access	24
Using Web Access with Netscape for Solaris	25
Using Web Access with HotJava and JavaStations	25
Changing the NIS Configuration	25
wasetup(1M) Man Page	26

**A. Man Pages for Sun Calendar Server 1.0 29**

Connecting to the Man Pages 29

Man Pages 30

iCald (1M) 30

iCal.conf 31

calsetup(1M) 33

calbackup(1M) 35

calrestor(1M) 37

calcp(1M) 40

calinplac(1M) 42

calimport (1M) 45

calexport (1M) 47

**Glossary 51**

**Index 55**



# Preface

---

The *Sun™ Calendar Server 1.0 Installation and Administration Guide* provides detailed information about installing and administering the Sun Calendar Server 1.0. This product is designed to manage large-scale calendaring systems. It offers features such as Calendar Store, Web Access, and administration with the goal of increasing both calendaring efficiency and performance. When used with the CDE Calendaring Client or Web Access for Sun Calendar Server, users can browse and edit their calendars anytime, anywhere.

Sun is committed to improving Sun Calendar Server 1.0 by helping to define a set of open calendaring and scheduling standards within the IETF (Internet Engineering Task Force). Sun Calendar Server 1.0 provides support for IETF calendar standards (iCalendar). This support for Internet standards-based features makes it easier for users to share calendaring information across the Internet.

Sun Calendar Server 1.0 lets you access your calendar from anywhere, regardless of location, platform, or browser.

## Key Features of the Product

- Real-time calendar and schedule management
- Access your calendar information from any browser, on any device
- Designated access control
- Scalable and flexible architecture
- Supporting Internet standards
- Calendar Store standards
- Maintain and administer calendars through command-line utilities

- Flexible client support – use either the Solaris™ CDE Calendaring Client or Web Access
- User self-administration
- Free-time search
- Migration tools to use with Solaris CDE calendar

---

**Note** – If you choose to work in the Open Windows environment, you must use the CDE Calendar Manager. See the *Solaris Common Desktop Environment User's Guide*.

---

## Who Should Use This Book

This book is intended for network and system administrators whose responsibilities include the set up and maintenance of the calendar servers. These servers can reside within medium to large networks of Solaris systems. The documentation can be used by both novice and experienced network and system administrators of desktop services.

## How This Book Is Organized

**Chapter 1, “Installing Sun Calendar Server 1.0,”** explains how to install and uninstall Sun Calendar Server 1.0.

**Chapter 2, “Administering Sun Calendar Server 1.0,”** explains how to manage user calendars. This includes creating, deleting, and modifying calendars; how to migrate users from the CDE Calendar manager to Sun Calendar; and how to backup users.

**Chapter 3, “Calendar Clients,”** details how to use the Solaris CDE Calendaring client and Web Access for Sun Calendar Server.

**Appendix A, “Man Pages,”** provides the complete manual pages associated with Sun Calendar Server 1.0.

**Glossary** is a list of words and phrases found in this book and their definitions.



## Related Information

For additional up-to-date product information, refer to the Sun Calendar Server web site. The URL is <http://www.sun.com/calendaring>. You will find:

- Press releases and data sheets
- Technical information
- Product documentation
- Product demos
- Product Frequently Asked Questions (FAQs)
- Updated Release Notes and Product Information

## What Typographic Changes Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> You have mail.
<b>AaBbCc123</b>	What you type, contrasted with on-screen computer output	<code>machine_name%</code> <b>su</b> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.

# Shell Prompts in Command Examples

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

**TABLE P-2** Shell Prompts

<b>Shell</b>	<b>Prompt</b>
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

# Installing Sun Calendar Server 1.0

---

This chapter describes how to install the Sun Calendar Server 1.0 onto the Solaris host machine. This chapter includes: installing and uninstalling.

---

## System Requirements

The minimum system requirements for installing Sun Calendar Server 1.0 are:

- **Platform:** Solaris™ 2.6 or 2.7 Operating Environment
- **Hardware:** SPARC™ or Intel Pentium platform
- **Disk Space for installation:** 70 Mbytes
- **Disk Space required for operation:** From 100 Mbytes to 2 Gbytes, depending on the number of users, plus user calendar storage space
- **Memory:** 64 Mbytes minimum; more for larger deployments

## Installation Packages

Sun Calendar Server 1.0 consists of two software packages:

- SUNWical – server and administration utilities
- SUNWwa – (Optional) Web Access for Calendar client

---

**Note** – If you choose to work in the Open Windows environment, you must use the CDE Calendar Manager. See the *Solaris Common Desktop Environment User's Guide*.

---

---

# Installing Sun Calendar Server 1.0

## Installation Utility

`setup` is a script that installs Sun Calendar Server 1.0 and its related files and packages onto the system.

The syntax for using this command is: `setup [-c install | remove] [-d]`  
The options for the `setup` command appear in TABLE 1-1.

**TABLE 1-1** Options for the `setup` Command

Option	Description
-c install	Specifies a standard install of Sun Calendar Server 1.0 and related files and packages. This is the default option.
-c remove	Specifies an uninstall of Sun Calendar Server 1.0 and related packages and files from the system.
-d	Specifies a non-interactive automated install using the <code>/tmp/scs_setup.dat</code> file, if it exists. If it does not exist, <code>setup</code> will default to the standard interactive install and prompt the user for necessary information.

## Installing Sun Calendar Server 1.0

The *Sun Calendar Server 1.0* CD includes all the packages you need (in uncompressed format) to install Sun Calendar Server 1.0 and Web Access for Sun Calendar.

### ▼ To install Sun Calendar Server 1.0

**1. Become superuser by typing:**

```
# su root
```

**2. Insert the CD into the CD-ROM drive.**

If your system is running Volume Manager, it will mount the CD in the following directory: `/cdrom/sun_calendar_server_1_0`

### 3. Change to this directory:

```
# cd /cdrom/sun_calendar_server_1_0
```

### 4. Run the setup to install all packages by typing:

```
# ./setup -c install
```

### 5. Select installation option.

```
# The following types of installations can be done on this machine:
```

1. Install SCS v1.0
2. Install SCS Optional Software

### 6. Type 1 to install Sun Calendar Server.

```
# Please select the type of installation (0=exit) [1|2|0]:
```

If you select 1, `setup` will offer you the option to install Web Access for Sun Calendar. If you select 2, the script will install only Web Access for Sun Calendar.

### 7. Type n (no) or y (yes) to install Web Access.

```
Web Access (n): [y, n,?]
```

If you type `y`, the install script invokes the appropriate Web Access install scripts. If do not install Web Access, you may install it later by running `setup` and choosing option 2.

`setup` begins installing the selected software packages.

---

**Note** – If you have Sun Calendar Server 1.0 already installed, `setup` will remove the existing packages and re-install the selected packages:

```
Removing existing SUNWical...
Removal of <SUNWical> was successful.
```

---

`setup` checks to see whether the required patches for a given platform are installed and prints a warning if the patches are not installed.

### 8. During reinstall only, configure the Calendar Store.

```
# Removing the calendar store on the Sun Calendar Server? (y):
[y, n,?]
```

---

**Caution** – If you answer `y`, `setup` will remove all the existing calendars in the calendar store. If `n`, the existing calendars are preserved.

---

### 9. During reinstall only, restore previously saved configuration files.

If you are reinstalling the Sun Calendar Server, you will be able to restore the configuration files from the previous install.

## 10. Configure the domain information:

---

**Note** – Sun Calendar Server (iCald) must know the primary e-mail domain name of its users.

---

If `setup` is unable to determine the email domain of the host, it will prompt you for the information.

## 11. Exit the installation process:

```
# Press the <enter> key to exit
```

Sun Calendar Server 1.0 installation log files are located in `/var/opt/SUNWical/log/SCS_installation_log` after installation is completed, and are in `/tmp` during installation

## Coexisting with the CDE Calendar Server

When the Sun Calendar Server 1.0 is installed, the install process automatically disables the CDE Calendar Server (`rpc.cmsd`). Sun Calendar Server 1.0 and `rpc.cmsd` cannot be executed simultaneously.

## ▼ To uninstall Sun Calendar Server 1.0

### 1. Invoke `setup` (as superuser).

```
# ./setup -c remove
```

### 2. Confirm uninstall of iCald.

```
# Are you sure you want to remove Sun Calendar Server(n): [y, n, ?,q]
```

---

**Note** – Uninstalling does not remove the calendar store; data is protected.

---

### 3. Confirm uninstall of Web Access for Sun Calendar.

```
# Are you sure you want to remove Sun Calendar Server(n): [y, n, ?,q]
```

# Troubleshooting Install

- If install fails:
  1. Check available disk space in */var/opt* and */opt*.
  2. Review install logs for errors (*/var/opt/SUNWical/log*).
- If iCald fails to respond after install, restart *inetd*.





# Administering Sun Calendar Server 1.0

---

There are two aspects of administration for the Sun Calendar server:

- Server administration, typically performed by the system administrator
- Calendar administration, performed by the owner of the calendar.

---

## Server Administration

Sun Calendar Server 1.0 is administered with command-line utilities. Administrative functionality includes creating, deleting, backup, restore, and migration of calendars.

- `iCald` – server executable
- `sdtcm_admin` – Solaris Calendar Manager (dtcm) administration utilities
- `calimport`, `calexport`, `calinplac` – conversion utilities
- `calbackup`, `calrestor`, and `calcp` – backup/restore/copy utilities

## Starting Sun Calendar Server 1.0

By default, the calendar server, `iCald`, is started by the Internet Services daemon (`inetd`). During install, the Internet servers database (`/etc/inet/inetd.conf`) is edited to include `iCald`.

`iCald` can also be run in stand-alone mode by first removing its entry in the Internet Services database. (See *Solaris Common Desktop Environment Administrator's Guide* for more information on configuring `inetd`.) You can then start `iCald` from the following directory: `/opt/SUNWiCal/lib/iCald`.

# Adding and Deleting Calendars

Use the `sdtcm_admin` command to create and administer calendars, and to add or remove a calendar on a local system or remotely on a host on a network. Also use `sdtcm_admin` to list calendars that exist on a host or to grant and revoke access to users to a specific calendar.

For a full description of the command line arguments, refer to the man pages in Appendix A.

---

**Note** – All calendar users must have a valid UNIX account on both the calendar server and all machines from which they access their calendars from. This includes any Web Access servers.

---

## ▼ To create a calendar

1. **The owner of the calendar must have a valid Solaris user account, for example: `jdoe`.**

Use `admintool` to create a Solaris user account, if the user does not have one. See the *Solaris Common Desktop Environment User's Guide* for information.

2. **As superuser, switch user id to the owner of the calendar.**

```
# su jdoe
```

3. **Create a calendar name (for example, `jdoe`), using the command:**

```
% /usr/dt/bin/sdtcm_admin -a -c jdoe@hostname
```

## ▼ To delete a calendar

1. **As superuser, switch user id to the owner of the calendar.**

```
# su jdoe
```

2. **A user can delete a calendar that he or she owns using the command:**

```
% sdtcm_admin -d -c calendar@host
```

For complete documentation on calendar management using `sdtcm_admin`, see the *Solaris Common Desktop Environment (CDE): Administrator's Guide*.

# Migrating calendars from `rpc.cmsd` to Sun Calendar Server

The migration utilities `calimport` and `calexport` are client programs that allow a user or system administrator to migrate calendar data from server to server. This allows a calendar to be moved without loss of information from `rpc.cmsd` to `iCald`.

For a full description of the command line arguments for the following utilities, refer to the man pages in Appendix A.

- `calexport` - export calendars from source server (Sun Calendar Server 1.0) to destination server.
- `calimport` - import calendars from source server to destination server (Sun Calendar Server 1.0).
- `calinplac` - setup environments to prepare for in-place conversion (migration).

## ▼ Migrating users with `calimport`

- **To migrate an individual user, as superuser run:**

```
# calimport jdoe@old_server jdoe@new_server
```

## ▼ Migrating users with `calinplac`

To upgrade an existing `rpc.cmsd` server to run `iCald`, `Cald` may be configured to run simultaneously with `rpc.cmsd` to facilitate migration of the existing calendars. This is referred to as in place conversion. A script, `calinplac`, is provided to assist with the in place conversion.

1. **As superuser, configure the system for in place conversion.**

```
# calinplac -p 100069
```

2. **Migrate a user, as shown.**

```
# calimport -p 100069 jdoe@server jdoe@server
```

3. **Return the system to its original configuration.**

```
# calinplac -r
```

For complete details on calendar migration utilities, refer to the man pages in Appendix A.

## ▼ Migrating calendars from Solaris 2.5.1

1. Upgrade the `rpc.cmsd` server's operating system to Solaris 2.6, or copy the `callog` file to a Solaris 2.6 system.
2. Upgrade the `callog` file to Version 4 on the 2.6 system.

```
# sdtcm_convert -v 4 cal
```

`sdtcm_convert` is automatically invoked during the Solaris 2.6 upgrade. Refer to the *Solaris Common Desktop Environment User's Guide* for complete usage details of `sdtcm_convert`.

3. Migrate the calendar to Sun Calendar Server.

```
# calimport cal@old_server cal@new_server
```

## Backing up and Restoring Calendars

All backup and restore functions are performed on-line. The backup utility produces a text file representation of the user's calendar. These files are saved on a local or remote file system (via NFS). Make arrangements to back up these files in accordance with your site policy.

- `calbackup` - back up a calendar to a text file.
- `calrestor` - restore a calendar from a text file.

---

**Note** – The format of the backup files is undocumented and subject to change. End users should not view or edit these files.

---

### ▼ Backing up a single calendar (as superuser)

```
# calbackup jdoe@host /export/BACKUPS
```

This will create a file called `/export/BACKUPS/jdoe.ics`.

### ▼ Backing up all calendars on a server (as superuser)

```
# calbackup host /export/BACKUPS
```

This will create a file called `/export/BACKUPS/<calendar_name>.ics` for each calendar on the host.

### ▼ Restoring a single calendar from a file (as superuser)

```
# calrestor /export/BACKUPS/jdoe.ics jdoe@host
```

This will restore the snapshot of the calendar stored in `/export/BACKUPS/jdoe.ics`.

## ▼ Restoring multiple calendars (as superuser)

```
# calrestor /export/BACKUPS host
```

This will restore all calendars saved in */export/BACKUPS*.

---

# Calendar Administration

The Sun Calendar Server allows the owner of a calendar to administer access rights to the calendar using Access Control Lists (ACL). ACLs can be viewed and updated with the `sdtcm_admin` command line utility or CDE calendaring client, `dtcm`.

## Using Access Control Lists (ACL)

Calendar entries are classified as PUBLIC, CONFIDENTIAL or PRIVATE. An ACL is stored within each calendar and maintained by the calendar owner using `sdtcm_admin`, or a GUI client such as `dtcm`. An ACL entry grants or denies an user individual permission to view, insert, and change entries of each classification:

**TABLE 2-1** Calendar Entry Permissions Matrix

	View	Insert	Change
Public	on/off	on/off	on/off
Confidential	on/off	on/off	on/off
Private	on/off	on/off	on/off

Connected users must be listed in the ACL with sufficient permissions for the operation they are attempting. For instance, if a connected user has only been granted permission to view PUBLIC entries, the server will filter out PRIVATE entries, and replace textual descriptions of entries marked confidential with the string `CLASSIFIED`.

## ▼ To view ACL information

### 1. List calendar information (as the owner of the calendar).

```
% sdtcm_admin -i -c cal_name@cal_server
calendar:          cal_name@cal_server
owner:             user
date created:      Tue Dec 1 21:28:13 PST 1998
data version:      unknown
server version:    6
size (bytes):      unknown
entries:           73
character set:     en_US.ISO-8859-1
product id:        -//DT//NONSGML Calendar Product Version 1//EN
access rights:
--> jdoe@Eng
Public: View Insert Change
Semiprivate: --
Private: --
Calendar Attributes: --
Roles: --
--> world
Public: View
Semiprivate: --
Private: --
Calendar Attributes: --
Roles: --
```

In this example, user jdoe has permission to view, insert, and change only Public entries. All other users have permission to only View public entries. Only the owner has permission to view, insert, and change Private and Semi-Private entries.

## ▼ To change ACL information

### ● Allow user “bob” to view, insert, and change public entries.

```
% sdtcm_admin -p pu+vic -n bob -c cal_name@cal_server
```

For complete details on `sdtcm_admin`, refer to *Solaris Common Desktop Environment User's Guide*.

## Calendaring Clients

---

---

### Accessing Sun Calendar Server 1.0 through Client Software

Sun Calendar Server 1.0 is based on native, Internet standards, and works with:

- CDE Calendaring Client (dctm)
- WebAccess for Sun Calendar Server

---

### CDE Calendaring Client (dctm)

CDE Calendaring Client (dctm) is a desktop application that lets you schedule appointments and To Do items, set reminders, make and print appointment and To Do lists, browse other calendars, and schedule group appointments.

---

**Note** – CDE is the acronym for Common Desktop Environment (in Solaris), a graphical user interface running on UNIX.

---

Type `dctm` to start Calendar from the command line. Use `sdctm_admin` from the command line to create and administer calendars.

Refer to *Solaris™ Common Desktop Environment (CDE): User's Guide* for detailed information.

---

# Web Access for Sun Calendar Server

Web Access for Sun Calendar Server 1.0 is a browser-based client environment that provides simplified Web access to information services such as email, calendar and directory. This approach:

- allows more global access to these key services via the web,
- leverages the ubiquity of the browser to deliver “client” functionality,
- and retains almost all functionality on the server where it can be more effectively and economically managed.

Web Access allows HTML based access to Sun Calendar Server. Using Web Access and virtually any Web browser, users can read, send, receive, store, and manage email; create and access calendar information; and view directory information.

End users can use virtually any desktop system in the organization without having to install the right client software. Web Access provides true multiplatform support through HTML. Using any Web browser capable of displaying HTML 3.2 compliant documents, the Web Access for Sun Calendar Server user will be able to view the contents of his/her calendar, create, edit, and delete appointments, and view the contents of other users' calendars stored on the Sun Calendar Server.

This section shows you how to install and invoke Web Access.

## Client Requirements for Running Web Access

Before invoking Web Access on a Sun Calendar Server 1.0 client desktop, make sure the Sun Calendar Server 1.0 package with the Web Access option is installed.

End users can invoke and run Web Access from any Internet browser capable of displaying HTML 3.2 compliant documents with frames running on any platform, including Windows, Mac, OS/2, and DOS desktops.

---

**Note** – Web Access has been tested extensively with Netscape Navigator 3.0, Microsoft Internet Explorer 3.0, and HotJava 1.1.

---

Web Access uses Java Web Server (JWS) 1.1 servlet technology and does not require a Java-enabled browser. Web Access does require a browser that supports cookies; cookie functionality must be enabled on any browser that uses Web Access.



# Installing Web Access

You can install Web Access on the same server as the Sun Calendar Server or on a separate server as a standalone product. Installing Web Access as part of the Sun Calendar Server process is described in Chapter 1. Installing Web Access on a separate server is detailed below.

## Installing Stand-alone Web Access

For optimal performance, it is recommended that Web Access be installed as a stand-alone product on a server separate from the Sun Calendar Server.

### *Web Access Installation Utility*

`wasetup` is a script that installs Web Access and its related files and packages onto the system.

The syntax for using this command is: `wasetup [-c install | remove] [-d][-v sims | scs] [-?]`.

The options for the `wasetup` command appear in TABLE 3-1.

**TABLE 3-1** Options for the `wasetup` Command

Option	Description
-c install	Specifies a standard install of Web Access and related files and packages.
-c remove	Specifies an uninstall of Web Access and related packages and files from the system.
-d	Specifies a non-interactive automated install using the <code>/tmp/wasetup.dat</code> file, if it exists. If <code>/tmp/wasetup.dat</code> does not exist, <code>wasetup</code> will default to the standard interactive install and prompt the user for necessary information.
-v sims	Specifies that Web Access is being installed for a SIMS environment.
-v scs	Specifies that Web Access is being installed for a SCS environment.
-?	Displays the usage and options for the <code>wasetup</code> command.

The following command performs a standard interactive installation of Web Access in a Sun Calendar Server environment:

```
# wasetup -c install -v scs
```

If you do not specify any arguments, wasetup will perform a `-c install -v sims` operation.

The following command uninstalls Web Access and related packages and files from the system:

```
# wasetup -c remove
```

## ▼ To Run wasetup Install using the CD-ROM

The Sun Calendar Server 1.0 CD includes all the Web Access packages you need in uncompressed format.

### 1. Insert the CD into the CD-ROM drive.

Because your system is running Volume manager, it will mount the CD in `/cdrom/sun_calendar_server_1_0`.

### 2. Change the current directory to the Sun Calendar Server product directory:

```
% cd /cdrom/sun_calendar_server_1_0/products/scs
```

### 3. Run wasetup to install all packages by typing:

```
% wasetup -c install -v scs
```

Installs the Web Access packages. wasetup begins installing software packages.

```
Installing Sun Web Access(tm) v1.0.3...
```

```
Checking system for minimum requirements.
```

```
System check indicates ultrek met minimum requirements.
```

```
Starting installation of software packages...
```

You see the following message:

```
-----  
Adding Package: SUNWjeev...  
-----  
=====> Adding Package: SUNWjeev
```

and then a series of copyright and file installation information for the Java Web Server. Finally you will see the message:

```
Installation of <SUNWjeev> was successful.
```

```
=====> Successfully added SUNWjeev
```

which indicates that the Java Web Server package was installed successfully.

Next, the installation of the Web Access package begins and you see the following message:

```
-----  
Adding Package: SUNWwa...
```

```
-----  
=====> Adding Package: SUNWwa
```

and then a series of copyright and file installation information for the Web Access package. Finally, you will see the message:

```
Installation of <SUNWwa> was successful.
```

```
=====> Successfully added SUNWwa
```

which indicates that the Web Access package was installed successfully.

#### 4. Select the naming service to use with Web Access:

```
Configuring Web Access for SCS
```

```
Please enter NIS Type[NIS|NIS+]:
```

```
Enter NIS or NIS+
```

#### 5. If NIS was chosen, enter the NIS domain:

```
Please enter NIS Domain(eng.sun.com):
```

Enter your NIS domain or press Enter to accept the default value displayed.

#### 6. Press Enter to complete the installation process:

```
Sun Web Access(tm) v1.0.3 installed and configured.
```

```
Please refer to your installation manual for further post-  
installation tasks.
```

```
Press the <enter> key to exit
```

## ▼ To install over the network

If you choose to install on the network, you must copy the entire CD to the disk storage on your network server. All packages are stored in the corporate internal network server, in an uncompressed format.

#### 1. The System Administrator accesses the network server.

#### 2. The System Administrator runs wasetup to install the Web Access packages.

The installation log is in `/var/opt/SUNWwa/log/WA_installation_log.n`, where `n` is the date and time stamp of the form `YYMMDD.HR:MM`. This number is generated automatically. For example, `/var/opt/SUNWwa/log/WA_installation_log.981214.18:38`.

# Uninstalling Web Access

## ▼ To uninstall Web Access

- **As root, type:**

```
# ./wasetup -c remove
```

This specifies a standard uninstall and removal of related files and packages.

The following message appears:

```
Are you sure you want to remove Sun Web Access(tm) (n):  
[y,n,?,q]
```

Answer `y` to remove Web Access from the system.

The following messages appear:

```
Removing SUNWjeev...  
Removal of <SUNWjeev> was successful.  
Removing SUNWwa...  
Performing Web Access shutdown...  
Web Access shutdown complete.  
Removal of <SUNWwa> was successful.
```

---

**Note** – Uninstalling does not remove the Web Access WebStore; its data is protected.

---

## Starting and Stopping Web Access

To start the Web Access server, type the following command:

```
# /etc/rc3.d/K89webaccess start
```

To shutdown the Web Access server, type the following command:

```
# /etc/rc3.d/K89webaccess stop
```

## Invoking Web Access

After installing and configuring Web Access for Sun Calendar Server 1.0, obtain the Web Access URL. Typically, the Web Access URL is as follows:

*http://<hostname>:8080/Web Access*

where hostname is the name of the Sun Calendar Server 1.0 providing Web Access service. The default port number of the Web Access server is 8080, and may be changed by the site administrator using the Java Web Server Administration Console.

### ▼ To invoke Web Access from a Sun Calendar Server 1.0 networked desktop

1. **Open an Internet browser.**
2. **Enter the Web Access URL and press Return.**

This invokes Web Access. See the on-line help that comes with Web Access.

3. **Login to Web Access.**

A dialog prompts you for permission to set a cookie.

4. **Click OK.**

## Troubleshooting

Section titles here try to anticipate questions and suggest system administrator responses.

### ▼ I can't login to Sun Web Access

Check to see whether Web Access is up and running.

1. **Go to `http://<hostname>:8080/WebAccess` and try to log into your own account.**

If you are unable to log in go to the next step.

2. **Go to `http://<hostname>:9090` and check whether the Java Web Server is running in the Java Web Server Administration Console.**

If the Java Web Server is not running the Administration Console, check to see if the Java Web Server process is running by typing the following in a shell:

```
% ps -ef | grep webaccess
```

### 3. Restart Web Access.

```
# /etc/rc3.d/K89webaccess stop
# /etc/rc3.d/K89webaccess start
```

If the login problem persists, the user may be trying to access Web Access through an incorrect URL.

### 4. Point the user to `http://<hostname>:8080/WebAccess`.

If the problem remains, check to confirm the user has a proper entry in NIS.

Here are other items to check:

- Is the user running a supported browser? Supported browsers include: Netscape Navigator 3.0, Microsoft Internet Explorer 3.0, and HotJava 1.1 or compatible versions.
- Does the browser have cookies enabled? Web Access requires that cookies be enabled. See the appropriate documentation for the user's browser and enable the cookies feature if it is not already.
- Is the user entering the correct login name and password? Check the user's NIS/passwd entry.

## Post-installation Sun Web Access Setup

After you have installed the Sun Web Access server either as an option along with Sun Calendar Server 1.0 or as a stand-alone product on its own server, the next step is to configure Web Access.

### Starting Sun Web Access

You can invoke and run Web Access from any Internet browser capable of displaying HTML 3.2 compliant documents with frames running on any platform, including Windows, Mac, OS/2, and DOS desktops. Web Access uses Java Web Server (JWS) 1.1 servlet technology from Java Software, and does not require a Java-enabled browser. Web Access does require a browser that supports cookies. Cookie functionality must be enabled on any browser that uses Web Access.

## Configuring Sun Web Access

You can perform any of the configuration options discussed here from an administrator-secure computer, the Java Web Server Administration Console, or from any networked computer. For more information about Java Web Server administration, go to <http://<hostname>:9090/system/doc/administration/main.html> or <http://jserv.javasoft.com/products/java-server/documentation/index.html>. This requires that the Java Web Server already be running on port 9090.

Configuring Web Access involves several post-installation options and considerations, including:

- Restarting the Web Access Server
- Changing the port number
- Establishing secure service
- Extending Web Access capabilities
- Using an alternate JVM
- Using Web Access with a Netscape for Solaris Browser
- Using Web Access with a HotJava Browser or a JavaStation

### ▼ To restart the Web Access server

- **Type the following commands:**

```
#!/etc/rc3.d/K89webaccess stop
#!/etc/rc3.d/S89webaccess start
```

## Changing the Port Number

The port for Web Access is managed through the Java Web Server Administration Console.

### ▼ To change the port number

1. **Open a Java Web Server Administration Console.**
2. **Enter the URL to access the Java Web Server.**

```
http://<hostname>:9090
```

where hostname is the Java Web Server server host name.

3. **Login to the Administration Console.**

---

**Note** – This procedure is documented in the Java Web Server help pages. To see them, click the Help button on the Java Web Server Administration Console Login screen.

---

**4. Select Web Service in the scrolling list of services and click on the Manage button.**

A dialog appears.

**5. Click on Network from the list at the left of the dialog.**

The frame to the right should show the Java Web Server port number.

**6. Select the port number and change it.**

**7. Restart the Web Access server.**

```
#!/etc/rc3.d/K89webaccess stop
#!/etc/rc3.d/S89webaccess start
```

## Establishing Secure Service

You can quickly and easily provide secure service with Web Access installations by enabling the Secure Socket Layer (SSL) feature of the Java Web Server. For requirements and a procedure, see “Using Secure Sockets Layer (SSL) with Java Web Server,” at the following URL (assuming Sun Web Access and the Java Web Server have been installed and are running):

*<http://<hostname>:8080/system/doc/security/ssl.html>*

This procedure shows you how to store server credentials in the key repository. Once server credentials are stored in the key repository, the Java Web Server will prompt the site administrator for a passphrase to gain access to the repository. By default, the server attempts to open a dialogue box using an X11 display at “:0”. If the server is unable to open this dialog, server start-up will fail.

### ▼ For sites that require hands-off restart ability

**1. Edit /etc/rc3.d/S89webaccess to include the line.**

```
WA_ARGS=$WA_ARGS "-passfile"
```

**2. Create a file in /opt/SUNWjeev called keys.passphrase, that contains the appropriate passphrase.**



### 3. Perform the following steps.

```
#chown root /opt/SUNWjeev/keys.passphrase
#chmod 600 /opt/SUNWjeev/keys.passphrase
```

---

**Note** – You must perform step 3 to protect the passphrase.

---

## ▼ For sites that require secure login

You can also configure Web Access services to require that users log in using secure connections only. Web Access provides a simple mechanism for enforcing this policy:

### 1. Open the following file `/opt/SUNWwa/properties/realm.SCS.properties`.

### 2. Answer yes to the following line.

```
SCSAuth.sslRequired=<yes|no>
```

### 3. Restart the Web Access server.

```
#/etc/rc3.d/K89webaccess stop
#/etc/rc3.d/S89webaccess start
```

You also can use the Java Web Server Administration Console to configure the “Document Root Directory” so that the Secure Web Service has exclusive access to the Web Access directory: `/opt/SUNWwa/public_html/WebAccess`. To do so, see the Java Web Server documentation.

## Extending the Web Access Palette

You can extend Web Access functionality by adding site-wide applications to the Web Access application palette. You can add an application by editing `selector.button` statements in the Web Access properties file.

## ▼ To add an application to Web Access

### 1. Open the file: `/opt/SUNWwa/properties/realm.SCS.properties`

**2. Add the following lines of code.**

```
selector.button<button-num>.method=<button-class>
selector.button<button-num>.href=<href-definition>
selector.button<button-num>.img=<button-icon-image>
selector.button<button-num>.alt=<button-alt-text-value>
selector.button<button-num>.targ=<button-targ>
```

where `button-class` specifies the Java servlet object class which will be invoked to fetch the appropriate HTML for the button. The remaining configuration items should be self-explanatory.

Buttons may be removed from the selector by deleting the appropriate `selector.button` statements. In any case, buttons must be numbered sequentially, 0 through n, and will be displayed in a corresponding top to bottom fashion.

## Using An Alternate JVM for Web Access

Web Access for Sun Calendar Server 1.0 ships with a JVM (Java Virtual Machine) that implements the JDK 1.1.6 (Java Development Kit) level of functionality. To use another JVM, for example, for performance and scalability, the JVM you specify must implement functionality equivalent to that of JDK 1.1.6.

### ▼ To configure Web Access to use another JVM

**1. Open the file:** `/etc/rc3.d/S89webaccess`

**2. Edit the file to include the following line.**

```
WA_ARGS=$WA_ARGS -nojre -javahome <java-home>
```

where `java-home` is the location of the new JVM.

**3. Restart the Web Access server.**

```
#/etc/rc3.d/K89webaccess stop
#/etc/rc3.d/S89webaccess start
```

It is your responsibility to insure that the resulting configuration meets your functional objectives.

## Using Web Access with Netscape for Solaris

System administrators should be aware of a potential security risk for authenticated users who access Web Access using a Netscape for Solaris browser. If that user logs out of Web Access and leaves the desktop unattended, it is possible for unauthorized persons to use that instance of the Netscape browser to gain access to the user's Web Access account without presenting a valid user name and password. Such users should:

- always exit the browser after logging out of Web Access
- use existing Solaris security mechanisms (screen lock, for example) when leaving their desktops unattended (even after logging out of Web Access)

---

**Note** – In extensive testing of Web Access, the Netscape for Solaris browser was the only browser that presented the potential security risk described above.

---

## Using Web Access with HotJava and JavaStations

System administrators that deploy Web Access for use on JavaStations or with the HotJava browser must first edit a Web Access configuration file to alter the Web Access authentication system.

Add the following line to the `realm.SCS.properties` file:

```
SCSAuth.checkId=no
```

This allows HotJava browsers and JavaStations to operate with Web Access. Administrators should be aware that such a Web Access configuration faces greater exposure to the limitations of BASIC HTTP authentication.

## Changing the NIS Configuration

After installing Web Access, you can specify an alternate NIS configuration and also configure the NIS domain used by Web Access.

## ▼ To change the NIS configuration

### 1. Open the following file:

```
/opt/SUNWical/properties/realm.SCS.properties
```

This file contains entries that determine the NIS configuration for Web Access.

### 2. Edit one or more of the following lines:

```
access.NISAccess.nisType=[NIS|NIS+]
```

```
access.NISAccess.nisDomain=<nis-domain-name>
```

Where, *nis-domain-name* refers to the name of the NIS domain to use and is needed only if *nisType* is set to NIS.

## wasetup(1M) Man Page

### NAME

wasetup - Set up the Web Access Server 1.0 and related files and packages.

### SYNOPSIS

```
wasetup[ -c install | remove ] [ -d ] [ -v sims | scs ]  
[ -? ]
```

### DESCRIPTION

wasetup is a script that installs the Web Access Server and related files and packages onto the system.

### OPTIONS

- c install Specifies a standard install of Web Access and related files and packages.
- c remove Specifies an uninstall of Web Access and related packages and files from the system.
- d Specifies a non-interactive automated install using the /tmp/wasetup.dat file, if it exists. If /tmp/wasetup.dat does not exist, wasetup will default to the standard interactive install and prompt the user for necessary information.
- v sims Specifies that Web Access is being installed for a SIMS environment.
- v scs Specifies that Web Access is being installed for a Sun Calendar Server environment.
- ? Displays the usage and options for this command.

## EXAMPLES

- The following command performs a standard interactive installation in a Sun Calendar Server environment:

```
example% wasetup -c install -v scs
```

- Execute the following command to uninstall Web Access and related packages and files from the system:

```
example% wasetup -c remove
```

- The following command:

```
example% wasetup -d
```

performs a non-interactive install which uses the file `/tmp/wasetup.dat` for all necessary configuration data if it exists. If the file does not exist, `wasetup` reverts to the interactive install, which prompts the user for necessary information.

## FILES

```
/cdrom/sun_calendar_server_1_0/products/scs/wasetup
```

## EXIT STATUS

- 0 Successful completion.
- 1 Utility failed.
- 2 Utility did not run because of usage error at command line prompt.

## NOTE

Because `wasetup` is not installed on the target system, you must retrieve the `wasetup` program from the distribution image and not from the system.

## ATTRIBUTES

See `attributes(5)` for a description of the following attributes:

**TABLE 3-2** Attributes for `wasetup`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

## SEE ALSO

`attributes(5)`



# Man Pages for Sun Calendar Server 1.0

---

Sun provides man pages for its operating system and associated software. Each man page discusses one subject, which could be a user command or library function. This appendix includes the man pages (short for manual pages) that you might need to refer to when using the Sun Calendar Server 1.0.

---

## Connecting to the Man Pages

You can access the Sun Calendar Server 1.0 man pages using the `man` command.

The man pages reside in the `/opt/SUNWical/man` directory. You can add this directory to your `$MANPATH` variable in your `.login` or `.cshrc` file by entering the following command:

```
% setenv MANPATH $MANPATH\:/opt/SUNWical/man
```

---

# Man Pages

## iCald (1M)

### NAME

iCald - Sun Calendar Server 1.0

### SYNOPSIS

/opt/SUNWical/lib/iCald

### DESCRIPTION

iCald is a high performance scalable replacement for the CDE calendar server `rpc.cmsd(1M)`. iCald manages the calendar storage for the client applications written to the X/Open Calendaring and Scheduling API. The primary interface for communicating with iCald is provided by `/usr/dt/lib/libcsa.so` which is described in `csa(5)`. iCald is normally invoked by `inetd(1M)` when an rpc request is received. At startup it reads the configuration files `iCald.conf(4)` and `cmsd.conf(4)`. It exits when it receives a SIGTERM signal. Messages are printed using `syslog`.

Calendar client applications currently supported include `dtcm(1)`, Web Access, and Hot Java views. The CDE calendar command line utilities `dtcm_lookup(1)`, `dtcm_insert(1)`, `dtcm_delete(1)`, and `dtcm_editor`, may also communicate with iCald.

Users may create and maintain calendars with `sdtcm_admin(1)`. Calendars may be migrated from `rpc.cmsd` using `calimport(1M)`. Calendars may be saved using `calbackup(1M)` and re-created with `calrestor(1M)`. iCald is based on rfc 2445 iCalendar. Future releases will support additional Internet standards.

iCald is started on demand by `inetd`.

### EXIT STATUS

The following exit values are returned:

- 0 Normal termination. A zero exit status means iCald received SIGTERM signal.
- >0 Abnormal termination. An error occurred.



## ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

**TABLE A-1** Attributes for iCald

Attribute Type	Attribute Value
Availability	SUNWical
Interface Stability	Stable

## SEE ALSO

calbackup(1M), calimport(1M), calrestor(1M), cmsd.conf(4), csa(1), dtcm(1), iCald.conf(4), rpc.cmsd(1M), sdtcm\_admin(1M)

## NOTES

The OpenWindows calendar client cm is not supported.

# iCal.conf

## NAME

`iCal.conf` - iCalendar (iCald) server configuration file

## SYNOPSIS

```
/etc/opt/SUNWical/iCald.conf
```

## DESCRIPTION

The `/etc/opt/SUNWical/iCald.conf` file contains the configuration information for operation of iCal, the iCalendar server. It is a text file containing keyword-value pairs and comments. The keyword-value pairs specify the value to use for parameters used by the iCal server. Comments can be entered by starting the line using the `#` character. The user can add comments to the file for customized configurations. Blank lines are ignored.

## KEYWORDS

All keywords are case-sensitive. Not all keywords must be present. If a keyword is not present, internal defaults, which are the default values described here, will be used. Keyword-value pairs are specified by:

```
keyword = value
```

#### MAX\_CLIENTS

A numeric value specifying the maximum number of simultaneous clients to be in service. A value of -1 means unlimited except where system resources is the limiting factor. Any positive value will set a limit on the number of clients to be in service at the same time unless system resource constraints come in before the limit. The default is (-1).

#### LISTEN\_Q\_SIZE

This is the value passed in as the backlog argument to the listen() network function. For a description of this value, see the listen() network function man page.

#### RPC\_PORT\_NUMBER

The port number for the calendar rpc protocol. If present, iCald respond to RPC calendar requests at the specified RPC port. All known existing RPC clients (CDE-dtcm, and Java-Views) will be looking for the RPC-calendar-server on the default value, RPC port 100068. The calendar migration utilities `calimport(1)` and `cal_export(1)` may be configured to send requests to a server on a port other than 100068. Thus iCald can be configured to run simultaneously with the CDE calendar server `rpc.cmsd`. Such operation facilitates so-called "in-place conversion". See `calinplac(1)` for more information.

#### TOP\_DIRECTORY

This keyword specifies the location for the top of the tree that stores calendar files. This directory must be on a local file system - i.e. the top directory may not be remote mounted. The default value is `/var/opt/SUNWical`.

#### EMAIL\_DOMAIN

iCald needs the hosts email domain name. If the host has a configured `/etc/resolv.conf` file, then this keyword is optional and iCald will use the domain value from the `/etc/resolv.conf` file. If the host does not have the `/etc/resolv.conf` file or the domain value is not in that file, then the EMAIL\_DOMAIN keyword must be supplied. The default is the domain value supplied in `/etc/resolv.conf`. If nether are supplied, iCald will exit and print an error via `syslogd` (see `syslogd(1M)`).

#### FILES

`/etc/opt/SUNWical/iCald.conf`

`/opt/SUNWical/lib/iCald`

## ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

**TABLE A-2** Attributes for iCald.conf

Attribute Type	Attribute Value
Availability	SUNWical
Interface Stability	Stable

## SEE ALSO

calexport(1m), calimport(1m), calinplac(1m), cmsd.conf(4), iCald(1)  
listen(3XN)

# calsetup(1M)

## NAME

`calsetup` - sets up the Sun Calendar Server (SCS) 1.0 and related files and packages.

## SYNOPSIS

```
calsetup [-c install | remove] [-d]
```

## DESCRIPTION

`calsetup` is a script that installs the Sun Calendar Server (SCS) and related files and packages onto the system.

## OPTIONS

- c `install` Specifies a standard install of SCS and related files and packages.
- c `remove` Specifies an uninstall of SCS and related packages and files from the system.
- d Specifies a non-interactive automated install using the `/tmp/scs_setup.dat` file, if it exists. If `/tmp/scs_setup.dat` does not exist, `calsetup` will default to the standard interactive install and prompt the user for necessary information. If `/tmp/scs_setup.dat` exists and `calsetup` is executed without the `-d` option specified, the `/tmp/scs_setup.dat` file is removed and the standard interactive install continues.

## EXAMPLES

The following command performs a standard interactive installation:

```
example% calsetup -c install
```

Execute the following to uninstall SCS and related packages and files from the system:

```
example% calsetup -c remove
```

The following command performs a non-interactive install:

```
example% calsetup -d
```

which uses the file `/tmp/scs_setup.dat` if it exists. It will gather all necessary configuration data from the `/tmp/scs_setup.dat` file.

## FILES

`/cdrom/sun_calendar_server_1_0/products/scs/calsetup`

## EXIT STATUS

- 0 Successful completion.
- 1 Utility failed.
- 2 Utility did not run because of usage error at command line prompt.

## NOTE

Because `calsetup` is not installed on the target system, you must retrieve the `calsetup` program from the distribution image and not from the system.

## ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

**TABLE A-3** Attributes for `calsetup`

Attribute Type	Attribute Value
Availability	SUNWical
Interface Stability	unstable

## SEE ALSO

`attributes(5)`

# calbackup(1M)

## NAME

calbackup - Backup one or multiple calendar(s)

## SYNOPSIS

```
calbackup [-f] [-v] src@src_host [file]
```

```
calbackup [-f] [-k] [-v] src_host [dst_dir]
```

## DESCRIPTION

The `calbackup` gets a snapshot of a calendar and writes the data to external storage. (The data is stored in iCalendar object format as a text file in UTF8 charset). Please see the NOTES for important information.

This utility should generally be run by users that have `read` and `write` permission to all entries in the calendar file(s). That is generally, the owner of the calendar or super user, `root`. If a user with less access permission of a target calendar run this program, information may be lost.

## OPTIONS

The following options are supported:

- f If a calendar file with the same name exists in the destination server, overwrite it without prompting the user for confirmation.
- k Keep processing subsequent calendars even when an error occurs. This option is only applied to the processing of multiple calendars.
- v Toggle verbose mode. If this option is enabled, for each calendar, `calbackup` will display total number of entries received on the screen and insert a comment saying the calendar was backup at a timestamp.

## OPERANDS

The following operands are supported:

`src@src_host` Source calendar name.

`file` Output iCalendar format data into a file. The file could be one of the followings:

- `regular file` Output data will be put into this file.
- `none` Output data will be put into a default file, `cal.ics.cal` is the calendar name.
- `'-'` Output data will be sent to the `stdout`.

- `directory` If file specified is an existing directory, the output data will be put into a file named after the calendar name with `.ics` as the file extension in that directory. If file specified doesn't exist, it will be created as a regular file instead of a directory.

`src_host` Source calendar host.

`dst_dir` Destination directory to store iCalendar format files. See “Files and Directories” in the next section for more information.

## USAGE

### Files and Directories

PLEASE SEE THE NOTES SECTION REGARDING THE FILE AND ITS CONTENTS.

If the user specifies the output data file name, `calbackup` will put the output data into it. Otherwise `calbackup` will automatically generate a default file name.

The default file name is taken from the calendar name and the extension is `.ics`. For instance, if the calendar name is `foo@scoots`, the output file will be `foo.ics`. If there is a subcalendar name, such as, `cal/subcal@src_host`, the `'/'` will be replaced by `'.'`. For instance, if the calendar name is `foo/foobar@scoots`, the output file name will be `foo.foobar.ics`.

If the directory is used, the creation and names of the files follows the specification described above. All file names will have the `.ics` file extensions appended to them. If the directory is specified, the files will be put into the specified directory, `dst_dir`. However, if the directory is not specified, the files will be put into the current working directory.

### EXAMPLES

- To backup one calendar to a file in the current directory:

```
example% calbackup foo@scoots old_cal.icsc
```

- To backup one calendar to `stdout`:

```
example% calbackup foo@scoots -
```

- To backup one calendar using the default file name:

```
example% calbackup foo@scoots
```

The output data will be put into a file `foo.ics` in the current working directory. `foo` is the calendar name.

- To backup all calendars on a server:

```
example% calbackup scoots tmp
```

- To backup all calendars on a server and put them into the current working directory:

```
example% calbackup scoots
```

## EXIT STATUS

The following exit values are returned:

- 0 All operands were traversed successfully.
- >0 An error occurred.

## ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

**TABLE A-4** Attributes for calbackup

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWical
CSI	SUNWical
Interface Stability	Stable

## SEE ALSO

calrestor(1m), calcp(1m)

## NOTES

Although the data is put into a file in iCalendar format, it is the private data to be written and used by the calbackup and calrestor utilities only. Please DO NOT modify the contents of the file or use it for other purpose.

# calrestor(1M)

## NAME

calrestor - Restore one or multiple calendar(s)

## SYNOPSIS

```
calrestor [-f] [-u username] [-v] file [dst@dst_host]
```

```
calrestor [-f] [-k] [-u username] [-v] file [dst_host]
```

```
calrestor [-f] [-k] [-u username] [-v] src_Dir [dst_host]
```

## DESCRIPTION

The `calrestor` takes the data generated by `calbackup` and recreates the original calendar. Note that in this release of the product incremental backup and restore is not supported.

This utility should generally be run by users that have `read` and `write` permission to all entries in the calendar file(s). That is generally, the owner of the calendar or super user, `root`.

## OPTIONS

The following options are supported:

- f If a calendar with the same name exists in the destination server, overwrite it without prompting the user for confirmation.
- k Keep processing subsequent calendars even when an error occurs. This option is applied to only the processing of multiple calendars.
- u Allow the owner of the calendar to be set or transferred to the specified user name.
- v Toggle verbose mode. If this option is enabled, for each calendar, `calrestor` will display the calendar name currently in process on the screen.

## OPERANDS

The following operands are supported:

- `file` Restore `iCalendar` format data from a file. The file can be:
  - `regular file` Restore `iCalendar` format data from this file.
  - `'-'` Restore `iCalendar` format data from the `stdin`. This operand is applied only to the single calendar address (`dst@dst_host`) case.
- `dst@dst_host` Destination calendar address. See “Calendar Name” and “Calendar Host Name” in the next section for more information.
- `dst_host` Destination calendar address. See “Calendar Host Name” in the next section for more information.
- `src_dir` Source directory contains `iCalendar` format input files. See “Files and Directories” in the next section for more information.
- `username` New calendar owner name. See “Calendar Owner Name” in the next section for more information.



## USAGE

### Files and Directories

PLEASE SEE THE NOTES SECTION REGARDING THE FILE AND ITS CONTENTS.

`calrestor` will restore `iCalendar` format files which were originally created by the `calbackup` utility.

If an `iCalendar` format input file name is specified, `calrestor` will restore data from it.

If the directory is used, `calrestor` will restore all valid `iCalendar` format files in this directory.

### Calendar Name

Normally the calendar name will be taken from the input file. However, if the calendar name is specified on the command line, `foo@scoots`, the calendar name specified, `cal`, will override the calendar name in the input file. In this way, `calrestor` provides the flexibility of allowing the user to restore a calendar under different name.

### Calendar Host Name

Normally the calendar host name will be taken from the input file. However, if the calendar host name is specified on the command line, `foo@scoots` or `scoots`. The calendar host name specified, `scoots`, will override the calendar host name in the input file. This way, `calrestor` provides the flexibility of allowing the user to restore a calendar onto different host name.

### Calendar Owner Name

Normally, the calendar owner name will be taken from the input file. However, if the calendar owner name is specified on the command line, `[-u username]`, the calendar owner name specified, `username`, will override the calendar owner name in the input file. This way, `calrestor` provides the flexibility of allowing the user to transfer a calendar from one owner to another. Note that only the root user has permission to create calendar with owner set to other user.

## EXAMPLES

- To restore one calendar from a file:  
example% `calrestor foo.ics`
- To restore one calendar from stdin:  
example% `calrestor - f`
- To restore all calendars on a server:  
example% `calrestor tmp`

## EXIT STATUS

The following exit values are returned:

0 All operands were traversed successfully.

>0 An error occurred.

## ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

**TABLE A-5** Attributes for `calrestor`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWical
CSI	Enabled
Interface Stability	Stable

## SEE ALSO

`calbackup(1m)`, `calcp(1m)`

## NOTES

Although the data is put into a file in iCalendar format, it is the private data to be written and used by the `calbackup` and `calrestor` utilities only. Please **DO NOT** modify the contents of the file or use it for other purpose.

# calcp(1M)

## NAME

`calcp` - Copy one or multiple calendar(s)

## SYNOPSIS

```
calcp [-f] [-v] src@srchost dst@dst_host
```

```
calcp [-f] [-k] [-v] srchost dst_host
```

## DESCRIPTION

`calcp` copies the information from one calendar to a new calendar with a given name.

This utility generally should be run by users who have `read` and `write` permission to all entries in the calendar file(s). That is generally, the owner of the calendar or super user, `root`.

## OPTIONS

The following options are supported:

- f If a calendar file with the same name exists in the destination server, overwrite it without prompting the user for confirmation.
- k Keep processing subsequent calendars even when an error occurs. This option is applied only to the processing of multiple calendars.
- v Toggle verbose mode. If this option is enabled, for each calendar, `calbackup` will display total number of entries received on the screen.

## OPERANDS

The following operands are supported:

<code>src@src_host</code>	Source calendar name
<code>dst@dst_host</code>	Destination calendar name
<code>src_host</code>	Source calendar host
<code>dst_host</code>	Destination calendar host

## EXAMPLES

- To copy one calendar file:  

```
example% calcp foo@scoots foo@polo3
```
- To copy all calendar(s) from the source server to the destination server:  

```
example% calcp scoots polo3
```

## EXIT STATUS

The following exit values are returned:

- 0 All operands were traversed successfully.
- >0 An error occurred.

## ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

**TABLE A-6** Attributes for calcp

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWical
CSI	Enabled
Interface Stability	Stable

## SEE ALSO

calbackup(1m), calrestor(1m)

# calinplac(1M)

## NAME

calinplac - Sets up or resets the calendar in-place conversion environments

## SYNOPSIS

```
calinplac [-p RPC PortNum] [-r] [-h]
```

## DESCRIPTION

For the purpose of migrating calendar, Sun Calendar Server 1.0 may be configured to run simultaneously with `rpc.cmsd` on the same machine. (`iCald` listens on a different port, while `rpc.cmsd` listens on the standard `rpc.cmsd` port, 100068.) This configuration is referred to as in place conversion. That is, conversion from `rpc.cmsd` style calendars to `iCald` style calendars with a single machine.

This `calinplac` script is provided to move the configuration of the system back and forth between in place conversion mode and the standard mode of operation for `iCald`. This utility must be run by the root user.

To do in-place conversion, the root user first runs `calinplac` to set the in place conversion mode of operation for `iCald`. After the in-place mode was set, the root user will run the migration tools, `calimport` or `calexport`, to migrate calendars. After the migration completed, the root user must run `calinplac` to exit the in place conversion mode and reset to the standard mode of operation for `iCald`.

To set the in place conversion mode of operation for *iCald*:

1. The root user runs `calinplac` to set the in place conversion mode of operation for *iCald*. While invoking `calinplac`, the root user can select any available RPC port number or run `calinplac` without any argument. If no argument is specified, `calinplac` will use the default RPC port number 100069.

```
calinplac
```

or

```
calinplac -p 100070
```

2. The running processes, `rpc.cmsd` and Sun Calendar Server, will be stopped.
3. The `RPC_PORT_NUMBER` in the `/etc/opt/SUNWical/iCald.conf` will be set to the new port num entered.
4. The `/etc/inet/inetd.conf` will be modified to enable port 100068 for `rpc.cmsd` and disable it for Sun Calendar Server.
5. The `rpc.cmsd` and Sun Calendar Server processes will be started again.
6. The root user is ready to do the calendar in place conversion. The root user will run the migration tools, `calimport` or `calexport` to migrate calendars.

To exit the in place conversion mode and reset to the standard mode of operation for *iCald*, the following tasks will be performed:

1. The root user runs `calinplac` to exit the in place conversion mode and reset to the standard mode of operation for *iCald*.

```
calinplac -r
```

2. The running processes, `rpc.cmsd` and Sun Calendar Server, will be stopped.
3. The `RPC_PORT_NUMBER` in the `/etc/opt/SUNWical/iCald.conf` will be reset to the original port num.
4. The `/etc/inet/inetd.conf` will be modified to enable port 100068 for Sun Calendar Server and disable it for `rpc.cmsd`.
5. Only the Sun Calendar Server process, not `rpc.cmsd`, will be started again.

## OPTIONS

The following options are supported:

- p The `port num` specifies the RPC Port ID of the *iCald* server.
- r Exit the in place conversion mode and reset to the standard mode of operation for *iCald*.
- h display this help.

## EXAMPLES

- Use the default RPC port number 100069 for the in-place conversion mode.  
example% calinplac
- Sets up the calendar in-place conversion mode to use RPC port on 100070.  
example% calinplac -p 100070
- Exit the in-place conversion mode and reset to the standard mode of operation for iCald.  
example% calinplac -rEXIT STATUS

## EXIT STATUS

The following exit values are returned:

- 0 All operands were traversed successfully.
- >0 An error occurred.

## ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

**TABLE A-7** Attributes for calinplac

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWical
Interface Stability	Stable

## SEE ALSO

calimport(1m), calexport(1m)

## NOTES

If there is no argument entered, calinplac will use the default RPC port number 100069 for the in-place conversion mode.

# calimport (1M)

## NAME

calimport - Convert calendar file

## SYNOPSIS

[`-d` *yyyymmdd*] Conversion start date <year month day>  
[`-f`] Overwrite destination without prompting  
[`-k`] Keep processing subsequent entries on error  
[`-p` *port num*] Port Number for server of destination calendar  
[`-v`] Toggle verbose mode  
<*src@src\_host*> Source calendar  
<*dst@dst\_host*> Destination calendar

## DESCRIPTION

The `calimport` command converts calendar file from `rpc` format to a `iCald` format. This utility should be run by users who have read and write permission to all entries in the calendar file. That is, generally, the owner of the calendar or super user, root.

The conversion utility is capable of converting calendars on a single machine or on two different machines. The option `-p` followed by port number should be specified when conversion is performed on a single host. This is known as in-place conversion.

## OPTIONS

The following options are supported:

- `-d` (*yyyymmdd*) This option indicates the starting date for conversion utility to convert calendar entries.
- `-f` If a calendar file with the same name exists in the destination host and it is in the `iCald` format, overwrite it without prompting the user for confirmation.
- `-k` Keep processing subsequent calendar entries even when an error occurs.

- p (port number) This option is used for any in-place conversions. The port number refers to the `RPC_PORT_NUMBER` configuration parameter of the iCald server. In order for this to work, the iCald server has to be configured with the `RPC_PORT_NUMBER` parameter set a NON-DEFAULT value, and the port number mentioned in this option should be the same value. See the `iCald.conf` man page for details on configuring the iCald server.
- v Toggle the verbose mode to indicate its progress.

## EXAMPLES

- To convert a calendar file where source host is different from the destination host:

```
example% calimport calendar@src_host calendar@dst_host
```

- To convert a calendar file to a new calendar name:

```
example% calimport calendar@src_host new_cal@dst_host
```

- To overwrite an existing converted calendar file without using `-f` option:

```
example% calimport calendar@src_host calendar@dst_host
```

```
Calendar "calendar@dst_host" exists. Do you want to replace it
(yes/no)?
```

- To overwrite an existing converted calendar file without prompting the user:

```
example% calimport -f calendar@src_host calendar@dst_host
```

- To convert a calendar file starting from a specific date:

```
example% calimport -d 19990125 calendar@src_host
calendar@dst_host
```

- To convert a calendar file on a single host (in place):

```
example% calimport -p 100070 calendar@src_host calendar@src
```

## EXIT STATUS

0 Migration utility ran successfully.

<0 An error occurred.

## ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

**TABLE A-8** Attributes for `calimport`

Attribute Type	Attribute Value
Availability	SUNWical
CSI	Enabled



# calexport (1M)

## NAME

calexport - Convert calendar file

## SYNOPSIS

`[-d yyyyymmdd]` Conversion start date <year month day>  
`[-f]` Overwrite destination without prompting  
`[-k]` Keep processing subsequent entries on error  
`[-p port num]` Port Num for server of source calendar  
`[-v]` Toggle verbose mode  
`<src@src_host>` Source calendar  
`<dst@dst_host>` Destination calendar

## DESCRIPTION

The `calexport` command converts calendar file from iCald format to a `callog rpc` format. This utility should be run by users that have read and write permission to all entries in the calendar file. That is generally, the owner of the calendar or super user, root.

The conversion utility is capable of converting calendars on a single machine or on two different machines. The option, `-p` followed by port number should be specified when conversion is performed on a single host. This type of conversion is known as in place migration. The host name for both source and destination host should be the same for in place migrations.

## OPTIONS

The following options are supported:

- `-d` (yyyyymmdd) Indicates the starting date for conversion utility to convert calendar entries.
- `-f` If a calendar file with the same name exists in the destination host and it is in the iCald format, overwrite it without prompting the user for confirmation.
- `-k` Keep processing subsequent calendar entries even when an error occurs.

- p (port number) Used to perform in place migrations. The port number refers to the `RPC_PORT_NUMBER` configuration parameter of the iCald server. The `RPC_PORT_NUMBER` can be configured using `calinplac`. See `calinplac` and `iCald.conf` man pages for details on configuring the iCald server.
- v Toggle the verbose mode to indicate its progress.

## EXAMPLES

- To convert a calendar file where source host is different from the destination host:

```
example% callexport calendar@src_host calendar@dst_host
```

- To convert a calendar file to a new calendar name:

```
example% callexport calendar@src_host new_cal@dst_host
```

- To overwrite an existing converted calendar file without using `-f` option:

```
example% callexport calendar@src_host calendar@dst_host
```

```
Calendar "calendar@dst_host" exists. Do you want to replace it
(yes/no)?
```

- To overwrite an existing converted calendar file without prompting the user:

```
example% callexport -f calendar@src_host calendar@dst_host
```

- To convert a calendar file starting from a specific date:

```
example% callexport -d 19990125 calendar@src_host
calendar@dst_host
```

- To convert a calendar file on a single host (in place):

```
example% callexport -p 100080 calendar@src_host
calendar@src_host
```

## EXIT STATUS

0 Reverse migration ran successfully.

<0 An error occurred.

## ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

**TABLE A-9** Attributes for `callexport`

Attribute Type	Attribute Value
Availability	SUNWical
CSI	Enabled
Interface Stability	Stable

## SEE ALSO

`calimport(1m)`, `calinplac(1m)`, `iCald(1)`, `attributes(5)`



# Glossary

---

- access control list** (ACL) An ACL is stored within each calendar and maintained by the calendar owner using `sdtcm_admin`, or a GUI client such as `dtcm`. An ACL entry grants or denies an user individual permission to view, insert, and change entries of each classification.
- API** Application Program Interface. A set of calling conventions defining how a service is invoked through a software package.
- backup** Gets a snapshot of a calendar in a single atomic transaction and writes the data to a file system. (The data is stored in iCalendar object format as a text file in UTF8 charset). (`calbackup`)
- Calendar** A desktop application that enables you to schedule appointments and To Do items, set reminders, make and print appointment and To Do lists, browse other calendars, and schedule group appointments.
- Calendar server** (daemon)
- Calendar view** The day, week, month, or year displayed in the main Calendar window.
- CAP** Client access protocol.
- CDE** (Common Desktop Environment) A graphical user interface running on UNIX®.
- CDE Calendaring Client (dtcm)** A desktop application that lets you schedule appointments and To Do items, set reminders, make and print appointment and To Do lists, browse other calendars, and schedule group appointments.
- client** In the client-server model for communications, the client is a process that remotely accesses resources of a compute server, such as compute power and large memory capacity.

- client-server model** A common way to describe network services and the model user processes (programs) of those services. Examples include the name-server/name-resolver paradigm of the DNS and fileserver/file-client relationships such as NFS and diskless hosts.
- command line** A string of characters beginning with a command, often followed by arguments, including options, file names, and other expressions, and terminated by the end-of-line character.
- cookie** A small piece of information sent by a web server to the user's browser. The user's browser stores the information and returns it to the server whenever the browser sends a request to the server.
- copy** Copying the information from one calendar to a new calendar with a given name (`calcp`). (This differs from the migration utility because it uses extensions to the `cmsd` protocol that allow an iCalendar representation to be transferred. While, `calimport` uses standard CSA.)
- DNS** Domain Name Service. The naming facilities of the Internet.
- domain** In the Internet, a part of a naming hierarchy. Syntactically, an Internet domain name consists of a sequence of names (labels) separated by periods (dots), for example, `tundra.mpk.ca.us`. In OSI, domain is generally used as an administrative partition of a complex distributed system, as in MHS Private Management Domain (PRMD), and Directory Management Domain (DMD).
- domain name** The name assigned to a group of systems on a local network that share administrative files. It is required for the network information service database to work properly. See also *domain*.
- dtdcm** See *CDE Calendaring client*.
- HotJava browser** A World Wide Web (WWW) browser, developed by Sun Microsystems, Inc., that uses the Java(TM) programming language. The HotJava browser enables the import and execution of code fragments, called applets, across the Internet.
- HTML** Hypertext Markup Language.
- HTTP** Hypertext Transfer Protocol.
- IAB** Internet Activities Board. The technical body that oversees the development of the Internet suite of protocols (commonly referred to as TCP/IP). It has two task forces (the IRTF and the IETF) charged with investigating a particular area.
- iCalendar** Core object specification for Internet calendaring and scheduling. (IETF calendaring standards)
- IESG** Internet Engineering Steering Group. The executive committee of the IETF.

<b>IETF</b>	Internet Engineering Task Force. One of the task forces of the IAB. The IETF is responsible for solving short-term engineering needs of the Internet. It has over 40 Working Groups.
<b>in place conversion</b>	Conversion from <code>rpc.cmsd</code> calendars to Sun Calendar Server 1.0 calendars with a single machine. The Sun Calendar Server may be configured to run simultaneously with <code>rpc.cmsd</code> . The <code>calinplac</code> command moves the configuration of the system back and forth between “in place” mode and the standard mode of operation for iCald.
<b>IP Internet Protocol</b>	The network layer protocol for the Internet protocol suite.
<b>iMIP</b>	Message-based interoperability protocol.
<b>iRIP</b>	Real-time interoperability protocol.
<b>iTIP</b>	Transport-independent interoperability protocol.
<b>Java</b>	A programming language developed by Sun Microsystems.
<b>Java Web server</b>	Based on the Java programming language, the Java Web Server 1.product uses Java technology servlets -- server side programs -- that enable you to easily extend the power of your web server. Servlets take the place of CGI scripts.
<b>JDK</b>	Java Development Kit. The JDK 1.1 platform contains the software and tools that developers need to compile, debug, and run applets and applications written using the Java programming language.
<b>JMAPI</b>	Java Management Application Programming Interface. JMAPI is a collection of programming language classes that enable a diverse set of autonomous applications to be brought together under a common look, feel, and behavior everywhere they run.
<b>JVM</b>	Java Virtual Machine. The cornerstone of Sun Java programming language. It is the component of the Java technology responsible for Java’s cross-platform delivery, the small size of its compiled code, and Java’s ability to protect users from malicious programs.
<b>man page</b>	UNIX Reference manual pages.
<b>migration utility</b>	Moves calendar data from one server to another. <code>calimport</code> transfers data from one calendar to a new calendar that it creates. <code>calexport</code> exports data from iCald. (The only difference between <code>calimport</code> and <code>calexport</code> is whether the input or output port is configurable). See also <i>in place conversion</i> .
<b>multithreaded</b>	The ability to handle multiple, simultaneous sessions in a single process.
<b>name resolution</b>	The process of mapping a name into the corresponding address. See also DNS.

- NFS®** Network File System. A distributed file system developed by Sun Microsystems which allows a set of computers to cooperatively access each other's files in a transparent manner.
- NIS** The SunOS™ 4.x network information service. A distributed network database containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.
- protocol** A formal description of messages to be exchanged and rules to be followed for two or more systems to exchange information.
- restore** Takes the data generated by `calbackup` and recreates the original calendar. (Incremental backup and restore is not supported.) (`calrestor`)
- RFC** Request For Comments. The document series, begun in 1969, describes the Internet suite of protocols and related experiments. Not all (in fact very few) RFCs describe Internet standards, but all Internet standards are published as RFCs. See <http://www.imc.org/rfc.html>.
- RPC** Remote Procedure Call. An easy and popular paradigm for implementing the client-server model of distributed computing. A request is sent to a remote system to execute a designated procedure, using arguments supplied, and the result is returned to the caller. The many variations and subtleties result in a variety of different RPC protocols.
- servlet** A server-side program that gives Java technology-enabled servers additional functionality.
- SIMS** See *Sun Internet Mail Server*.
- SSL** Secure Socket Layer feature of the Java Web Server
- Sun Internet Mail Client** The client end of the SIMS solution that supports online, offline, and disconnected states.
- Sun Internet Mail Server** An enterprise-wide, open-standards based, scalable electronic message-handling system.
- URL** (Uniform Resource Locator) A standard for writing a textual reference to an arbitrary piece of data in the World Wide Web (WWW). A URL looks like "protocol://host/localinfo" where "protocol" specifies a protocol for fetching the object (like HTTP or FTP), "host" specifies the Internet name of the targeted host, and "localinfo" is a string (often a file name) passed to the protocol handler on the remote host.
- Web Access for Sun Calendar Server** A browser-based client environment that provides simplified Web access to your calendar.



# Index

---

## A

- Access Control Lists (ACL) 11
- adding applications to Web Access 23
- adding calendars 8
- administering Sun Calendar Server 1.0 7

## B

- backing up a single calendar 10
- backing up all calendars on a server 10

## C

- c install option 2
- c remove option 2
- calbackup(1M) man page 35
- calbackup, backup calendars 10
- calcp(1M) man page 40
- calendar administration 11
- calendar administration, Sun Calendar 7
- calendar entry permissions 11
- calendar store 4
- calendar users, UNIX account 8
- calendarizing clients 13

## calendars

- Access Control Lists (ACL) 11
- adding and deleting 8
- backing up a single 10
- backing up all on a server 10
- creating 8
- deleting 8
- migrating from Solaris 2.5.1 10
- restoring from a file 10
- restoring multiple 11

- calexport (1M) man page 47
- calimport (1M) man page 45
- calimport, migrating users 9
- calinplac(1M) man page 42
- calinplac, migrating users 9
- callog file, upgrading 10
- calrestor(1M) man page 37
- calrestor, restoring calendars 10
- calsetup(1M) man page 33
- CDE Calendar Server, and Sun Calendar Server 4
- CDE Calendarizing Client (dtcm) 13

- changing
  - NIS configuration 26
  - port number 21
- client requirements for Web Access 14
- client software 13
- configuring Web Access 21
- cookies 14
- creating calendars 8

## D

- d option 2
- deleting calendars 8
- dtcm, CDE Calendaring Client 13

## E

- exiting installation, Sun Calendar Server 4

## I

- iCal.conf man page 31
- iCald
  - calender server 7
  - stand-alone mode 7
- iCald (1M) man page 30
- inetd daemon 7
- install, troubleshooting 5
- installation packages 1
- installation utility, setup command 2
- installing
  - stand-alone Web Access 15
  - Sun Calendar Server 1.0 1-5
  - Web Access 3, 15-17
  - Web Access over the network 17
- invoking Web Access 19

## J

- JVM (Java Virtual Machine) 24

## M

- man pages 29-49
- migrating
  - calendars
    - from rpc.cmsd to Sun Calendar Server 9
    - from Solaris 2.5.1 10
  - users with calimport 9
  - users with calinplac 9

## N

- network install 17
- NIS configuration, changing 26

## O

- Open Windows, using with Sun Calendar Server 1

## P

- port number, changing 21

## R

- reinstall, Sun Calendar Server 3
- restarting Web Access server 21
- restoring a single calendar from a file 10
- restoring multiple calendars 11

## S

- sdtcm\_admin command 8
- secure service, Web Access 22
- Secure Socket Layer (SSL) 22
- server administration, Sun Calendar 7
- setup command, Sun Calendar Server 2
- shutting down Web Access 18
- Solaris user account, required 8
- starting
  - Sun Calendar Server 1.0 7
  - Web Access 18
- Sun Calendar Server 1.0
  - administration 7
  - and CDE Calendar Server 4
  - installation packages 1
  - installing 1-5
  - migrating from rpc.cmsd 9
  - reinstall 3
  - starting 7
  - system requirements 1
  - uninstalling 4

- SUNWical 1
- SUNWwa 1
- system requirements 1

## T

- troubleshooting
  - install 5
  - Web Access 19

## U

- uninstalling
  - Sun Calendar Server 1.0 4
  - Web Access 18
- UNIX account, required 8
- upgrading callog file 10
- users
  - migrating with calimport 9
  - migrating with calinplac 9
  - UNIX account 8

## W

- wasetup script 15
  - install 16
- wasetup(1M) man page 26

### Web Access

- adding applications 23
- alternate JVM for 24
- client requirements 14
- configuring 21
- installation utility 15
- installing 3, 15–17
- installing stand-alone 15
- invoking 19
- network install 17
- shutting down 18
- starting 18
- troubleshooting 19
- uninstalling 18
- wasetup script 15

Web Access for Sun Calendar Server 14–27