Sun Gathering Debug Data for Sun Java System Calendar Server

Sun Java™ Communications Suite Technical Note

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

Part No: 820–0374–11
January 2007

CHAPTER 1

Sun Gathering Debug Data for Sun Java System Calendar Server

This technical note describes how to use Sun™ Gathering Debug Data (Sun GDD or GDD) to collect data that the Sun Support Center requires in order to debug problems with a Sun Java™ System Calendar Server system. By collecting this data before you open a Service Request, you can reduce substantially the time needed to analyze and resolve the problem. For more information on how this document and associated scripts can help you in better dealing with Calendar Server problems, see:

http://www.sun.com/service/gdd/index.xml

This document is intended for anyone who needs to open a Service Request about Calendar Server with the Sun Support Center.

This technical note contains the following sections:

1.1 Technical Note Revision History

Version	Date	Description of Changes
12	June 2007	Updated document to direct Solaris users to use the CS Capture script to collect Calendar Server data. Pointed to the CS Capture document for details.
11	January 2007	Updated "To Configure Solaris OS to Generate Core Files" on page 23.
10	December 2006	Initial release of this technical note.

1.2 About This Technical Note

This document covers the following versions of Sun Java System Calendar Server on the Solaris™ Operating System, HP-UX, Linux, and Microsoft Windows platforms:

- Sun Java System Calendar Server 6.3 (Communications Suite 5)
- Sun Java System Calendar Server 6.2 2005Q4
- Sun Java System Calendar Server 6.2 2005Q1
- Sun Java System Calendar Server 6.1 2004Q2
- Sun Java System Calendar Server 6.0 2003Q4

You can use this document in all types of environments, including test, pre-production, and production. Verbose debugging is not used (to reduce performance impact), except when it is deemed necessary. At the same time, it is possible that the problem could disappear when you configure logging for debug mode. However, this is the minimum to understand the problem. In the majority of cases, the debug data described in this document is sufficient to analyze the problem.

This document does not provide workarounds nor techniques or tools to analyze debug data. It provides some troubleshooting, but you should not use this guide as an approach to troubleshooting Calendar Server problems.

If your problem does not conveniently fit into any of the specific categories, supply the general information described in "1.5 What Calendar Server Debug Data Should You Collect?" on page 7 and clearly explain your problem.

If the information you initially provide is not sufficient to find the root cause of the problem, Sun will ask for more details, as needed.

1.2.1 Prerequisites for Collecting Calendar Server Debug Data

The prerequisites for collecting debug data for Calendar Server are as follows:

- Make sure you have `superuser` privileges.
- For the Solaris OS platform, obtain the `dbhang`, `pkg_app`, and `cscapture` scripts from the following location:

  `http://www.sun.com/bigadmin/scripts/indexSjs.html`

  For information about running the `pkg_app` script, see "To Run the pkg_app Script" on page 25.

  For information about running the `cscapture` script, see *Using Calendar Server Capture (cscapture) to Collect Debug Data for Sun Java System Calendar Server*.
- On the Windows platform, download the free Debugging Tools for Windows to help in analyzing process hang problems. The debugger Dr. Watson is not useful for process hang problems because it cannot generate a crash dump on a running process. Download the free Debugging Tools from the following location:

  `http://www.microsoft.com/whdc/devtools/debugging/default.mspx`

  Install the last version of Debugging Tools and the OS Symbols for your version of Windows. Also, you must add the environment variable `NT_SYMBOL_PATH`.

  Use the command `drwtsn32 -i` to select Dr. Watson as the default debugger. Use the command `drwtsn32`, check all options, and choose the path for crash dumps.

1.2.2 Variables Used in This Technical Note

The following describes the variables used in the procedures in this document. Gather the values of the variables if you don't already know them before you try to do the procedures.

- *calendar-service-port*: Port number used by the Calendar Server.
- *calendar-pid*: Process ID of a Calendar Server daemon.
- *calendar-process-name*: Process name of a Calendar Server process such as `csadmind` or `csnotifyd`.
- *identifier*: The Directory Server instance name used during installation. The installation program automatically adds the prefix `slapd-` to the name you specify. For example, if you name the *identifier* `tango`, the installation program creates the name `slapd-tango`.
- *cal-svr-base*: The directory on the Calendar Server machine dedicated to holding the Calendar Server program, configuration, maintenance, and information files. The default location for the Solaris OS version of Sun Java System Calendar Server 6 is `/opt/SUNWics5/cal/`. See "To Obtain the *cal-svr-base* Variable" on page 6 for more information on determining the value of *cal-svr-base*.

- *server-root*: The placeholder for the directory where the Directory Server instances and data reside.
- *windbg-root*: The directory on the Windows Calendar Server machine dedicated to holding the Win Debugger program, and configuration, maintenance, and information files.

▼ To Obtain the *cal-svr-base* Variable

The *cal-svr-base* variable refers to the directory on the machine that holds the Calendar Server's programs, configuration, maintenance, and information files. This is the directory in which you install the server software.

● **Use the following commands to obtain the value of the Calendar Server *cal-svr-base* variable on Solaris and Linux systems:**

Solaris	`pkgparam -v SUNWics5`
Linux	`rpm -q –qf ’%{INSTALLPREFIX}’ sun-calendar-server`

1.3 Overview of Collecting Debug Data for Calendar Server

Collecting debug data for a Calendar Server problem involves these basic operations:

1. Collecting basic problem and system information.
2. Collecting specific problem information (installation problem, process hang, process crash, and so on).
3. Creating a `tar.gz` file of all the information and uploading it for the Sun Support Center.
4. Creating a Service Request with the Sun Support Center.

1.4 Creating a Service Request with the Sun Support Center

When you create a Service Request with the Sun Support Center, either online or by phone, provide the following information:

- A clear problem description
- Details of the state of the system, both before and after the problem started
- Impact on end users
- All recent software and hardware changes
- Any actions already attempted
- Whether the problem is reproducible; when reproducible, provide the detailed test case

- Whether a pre-production or test environment is available
- Name and location of the archive file containing the debug data

Upload your debug data archive file to one of the following locations:

- `http://supportfiles.sun.com/upload`
- `https://supportfiles.sun.com/upload`

For more information on how to upload files to this site, see: `http://supportfiles.sun.com/show?target=faq`

Note – When opening a Service Request by phone with the Sun Support Center, provide a summary of the problem, then give the details in a text file named `Description.txt`. Be sure to include `Description.txt` in the archive along with the rest of your debug data.

1.5 What Calendar Server Debug Data Should You Collect?

This section describes the kinds of debug data that you need to provide based on the kind of problem you are experiencing.

This section contains the following tasks:

- "To Collect Required Debug Data for Any Calendar Server Problem" on page 7
- "To Collect Debug Data on Calendar Server Installation Problems" on page 9
- "To Collect Debug Data on a Hung or Unresponsive Calendar Server Process" on page 11
- "To Collect Debug Data on a Calendar Server Crashed Process" on page 15
- "To Collect Debug Data on a Calendar Server Database Problem" on page 18
- "To Collect Debug Data on a Calendar Server CLD/DWP Problem" on page 20
- "To Collect Debug Data on a Calendar Express or Communications Express Interface Problem" on page 23

▼ To Collect Required Debug Data for Any Calendar Server Problem

All problems described in this technical note need basic information collected about when the problem occurred and about the system having the problem. Use this task to collect that basic information.

1 **Note the day(s) and time(s) the problem occurred.**

2 Provide a graphical representation of your deployment. Include all hosts and IP addresses, host names, operating system versions, role they perform, and other important systems such as load balancers, firewalls, and so forth. Also include the following information:

- **Calendar Server topology.** For example, describe front-end and back-end Calendar servers, if you have them.
- **LDAP Directory topology.** Describe your Directory Server deployment features such as replicated directories. Are any caches enabled?
- **Calendar Server features.** Describe other Calendar Server features you have configured. These include, at a minimum, virtual domains, SSL, SSO, and LDAP cache..

3 Note the operating system.

Solaris OS	`uname -a`
HP-UX	`uname -r`
Linux	`more /etc/redhat-release`
Windows	`C:\Program Files\Common files\Microsoft Shared\MSInfo\msinfo32.exe /report C:\report.txt`

4 Note the patch level.

Solaris OS	`showrev -p | grep SUNWics5`
HP-UX	`swlist`
Linux	`rpm -qa`
Windows	Already provided in the `C:\report.txt` file above.

Note – If possible, also provide explorer information (`SUNWexplo`) of the machine having the problem. Edit the `ics.conf` file and increase the log level by using the `logfile.loglevel = "Debug"` parameter.

---

5 Note the version of Calendar Server.

Be sure to send the entire screen output of the command.

UNIX and Linux	`cd` *cal-svr-base* `./cal/bin/csstart version`

Windows	`cd` *cal-svr-base*`\cal\bin\`
	`csstart.exe version`

6 Create a tar file of the Calendar Server configuration directory.

UNIX and Linux	Create a tar file of the *cal-svr-base*`/cal/bin/config` directory.

Windows	Create a tar file of the *cal-svr-base*`\cal\bin\config` `directory`.

Note – Always be sure to include the `ics.conf` file.

7 **Get the `start`, `stop`, `http`, `admin`, and `notifyd` processes log files.**

The `logfile.logdir` parameter in the `ics.conf` file specifies the paths of these log files.

Note – If possible, provide just the relevant extracts of log files for the same time period that show the problem, with sufficient context to see what else was occurring during the error occurrence and shortly before. Thus for relatively short log files, send the entire log file, whereas for long-running hence large log files, an extract might be more appropriate, though be sure to include all the material from the time of the error as well as at least some lead-in logging from before the error apparently occurred.

8 **Get the Access, Errors, and Audit logs from the Directory Server used by Calendar Server.**

The paths of these files are specified by the following parameters in the Directory Server `dse.ldif` file:

- `nssldap-accesslog`
- `nssldap-errorlog`
- `nssldap-auditlog`

UNIX and Linux	*server-root*`/slapd-`*identifier*`/logs/access` *server-root*`/slapd-`*identifier*`/logs/errors` *server-root*`/slapd-`*identifier*`/logs/audit` (if enabled)
Windows	*server-root*`\slapd-`*identifier*`\logs\access` *server-root*`\slapd-`*identifier*`\logs\errors` *server-root*`\slapd-`*identifier*`\logs\audit` (if enabled)

▼ To Collect Debug Data on Calendar Server Installation Problems

Follow these steps if you are unable to complete the installation or if you get a “failed” installation status for Calendar Server.

1. **Consult the following troubleshooting information.**

   Read the “Troubleshooting” chapter in the *Sun Java Communications Suite Installation Guide* or, for older versions of Calendar Server, *Sun Java Enterprise System Installation Guide*.

   - Calendar Server 6.3: *Sun Java Communications Suite 5 Installation Guide*

     http://docs.sun.com/app/docs/doc//819–7560/6n9gev7mh?a=view
   - Calendar Server 6.2 2005Q4: Chapter 9, “Troubleshooting,” in *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX*
   - Calendar Server 6.2 2005Q1: *Sun Java Enterprise System 2005Q1 Installation Guide*

     http://docs.sun.com/source/819–0056/troubleshooting.html
   - Calendar Server 6.1 2004Q2: *Sun Java Enterprise System 2004Q2 Installation Guide*

     http://docs.sun.com/source/817–5760/troubleshooting.html
   - Calendar Server 6.0 2003Q4: *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX*

     http://docs.sun.com/source/816–6874/std-troubleshooting.html

   If the problem persists after using this troubleshooting information, then continue with this procedure to collect the necessary data for the Sun Support Center.

2. **Collect the general system information as explained in “To Collect Required Debug Data for Any Calendar Server Problem” on page 7.**

3. **Specify if this is a first-time installation or a Hot Fix installation on a pre-existing Calendar Server instance.**

4. **Get the installation logs.**

   Solaris OS
   : `/var/sadm/install/logs`
   The log file names start with `Java_Enterprise_System*_install.B`*datetime*, where *date* and *time* correspond to the failing installing (for example, `B12161532`).

   HP-UX and Linux
   : `/var/opt/sun/install/logs`
   The log file names start with `Java_Enterprise_System*_install.B`*datetime*, where *date* and *time* correspond to the failing installing (for example, `B12161532`).

   Windows
   : `C:\DocumentsandSettings\`*current-user*`\LocalSettings\Temp`
   The log file names start with `CSI*.log` (usually a text file). The asterisk (*) represents a random number in the `Temp` directory for each CSI based setup.

▼ To Collect Debug Data on a Hung or Unresponsive Calendar Server Process

A process hang is defined as one of the Calendar Server processes not responding to requests anymore while the process is still running locally. Calendar Server's six specific processes are:

- `enpd`—Collects and dispatches events that occur to properties of resources (calendars).
- `csadmind`—Provides a single point of authentication for administering Calendar Server. This service also manages alarm notifications and group scheduling requests.
- `csnotifyd`—Sends notifications of events and tasks and subscribes to alarm events. When an alarm event occurs, this service sends SMTP message reminders to recipients.
- `cshttpd`—Listens for and receives HTTP commands from Calendar Server end users and returns calendar data.
- `csdwpd`—Distributes calendar databases across multiple back-end servers.
- `csstored`—When configured properly, creates automatic backups of the calendar database.

Caution – Calendar Server processes usually hang because of an orphan lock left in one of the databases. Stopping the server (especially the `csstored` process), and cleaning the temporary shared database files helps to resolve the problem. This task is described in Step 13, at the end of the following procedure.

If you are *sure* that the hung process is a fleeting and insignificant issue, and you do *not* need help from the Sun Support Center, you can go to Step 13 now. **Otherwise, do not stop and restart Calendar Server until you have gathered the data requested in the following procedure. Stopping and restarting the server destroys all the debug data related to the hung process.**

Before You Begin Make sure that you collect all the data over the same time frame in which the problem occurs. See "1.6 Configuring Solaris OS to Generate Core Files" on page 23 if a core file is not generated.

On Solaris systems, you can easily gather the required data by running the `cscapture` command in Invasive mode.

```
./cscapture -i
```

For more information about running `cscapture` and its Invasive operation, see *Using Calendar Server Capture (cscapture) to Collect Debug Data for Sun Java System Calendar Server*.

The `cscapture` command gathers `netstat`, `ps`, `swap`, `gcore`, `pkg_app`, and other data. After you run `cscapture` on a hung or unresponsive process, proceed to Step 10, below, which describes how to restart the calendar services.

For all other platforms, collect the following information for process hang problems. Run the commands in order when the problem occurs. Be sure to specify the time when the process hang happened and affected processes, if possible.

1 **Collect the general system information as explained in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.**

2 **Specify the time the hang occurred and, if possible, the process that was hung.**

3 **Run the `netstat` command and save the output.**

HP-UX and Linux `netstat -an | grep` *calendar-service-port*

Windows `netstat -an`

4 **Run the following commands and save the output.**

HP-UX

```
ps -ef | grep cal-svr-base
vmstat 5 5
iostat -x
top
sar
```

Linux

```
ps -aux | grep cal-svr-base
vmstat 5 5
top
uptime
sar
```

Windows

Obtain the `CALENDAR` process PID: `C:\`*windbg-root*`>tlist.exe`

Obtain process details of the `CALENDAR` running process PID: `C:\`*windbg-root*`>tlist.exe` *calendar-pid*

Note – To use the preceding commands on Windows systems, you need to install the debugging tools, available from the following url:

`http://www.microsoft.com/whdc/devtools/debugging/default.mspx`

Install the latest version of the debugging tools and OS symbols for your version of Windows.

You also must add the environment variable `"_NT_SYMBOL_PATH"`.

---

5 **Get the swap information.**

HP-UX `swapinfo`

Linux	`free`
Windows	Already provided in `C:\report.txt` as described in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.

6. **Get the Calendar Server process log files.**

The `logfile.logdir` parameter in the `ics.conf` file specifies the locations of these log files.

On Solaris systems, the default value of the path is `/var/opt/SUNWics5/logs`.

Each process uses its own log file, as shown in the following list:

`csadmind`
: `ics.conf` setting: `logfile.admin.logname`

  Default value: `admin.log`

`cshttpd`
: `ics.conf` setting: `logfile.http.logname`

  Default value: `http.log`

`csdwpd`
: `ics.conf` setting: `logfile.dwp.logname`

  Default value: `dwp.log`

`csnotifyd`
: `ics.conf` setting: `logifle.notify.logname`

  Default value: `notify.log`

`csstored`
: `ics.conf` setting: `logifle.store.logname`

  Default value: `store.log`

`watcher` (Communications Suite 5 release only)
: `ics.conf`: No setting; file is always called `watcher.log`.

  Default value: `watcher.log`

The enpd process does not have a log.

7. **Look for any core file that could have been dumped by one of the Calendar Server processes. If you find one, see "To Collect Debug Data on a Calendar Server Crashed Process" on page 15.**

8. **Get the output of the following command.**

HP-UX	`tusc -v -fealT -rall -wall -o /tmp/`*calendar-process-name*`-`*calendar-pid*`.tusc.out -p` *calendar-pid*

Linux strace -fv -o /tmp/*calendar-process-name*-*calendar-pid*.strace.out -p *calendar-pid*

Windows Use DebugView: http://www.sysinternals.com/Utilities/DebugView.html

---

Note – Wait one minute after launching the appropriate command (truss, strace, tusc, or DebugView) then stop it by pressing **Control-C** in the terminal where you launched the command.

---

9 **Get core files and the output of the following commands.**

In a process hang situation, it is helpful to compare several core files to review the state of the threads over time. To not overwrite a core file, copy that core file to a new name, wait approximately one minute, then rerun the following commands. Do this three times to obtain three core files.

Note – For HP-UX, you need the following two patches to use the gcore command: PHKL_31876 and PHCO_32173. If you cannot install these patch, use the HP-UX /opt/langtools/bin/gdb command from version 3.2 and later, or the dumpcore command.

HP-UX

```
# cd cal-svr-base/cal/bin
# gcore -p calendar-pid
(gdb) attach calendar-pid
Attaching to process calendar-pid
No executable file name was specified
(gdb) dumpcore
Dumping core to the core file core.calendar-pid
(gdb) quit
The program is running. Quit anyway (and detach it)? (y or n) y
Detaching from program: , process calendar-pid
```

Linux

```
# cd cal-svr-base/cal/bin
# gdb
(gdb) attach calendar-pid
Attaching to process calendar-pid
No executable file name was specified
(gdb) gcore
Saved corefile core.calendar-pid

(gdb)backtrace
(gdb)quit
```

Windows Get the CALENDAR process PID:

```
C:\windbg-root>tlist.exe
```

Generate a crash dump on the CALENDAR running process PID:

```
C:\windbg-root>adplus.vbs -hang -p calendar-pid -o C:\crashdump_dir
```

Note – For Windows, provide the complete generated folder under `C:\crashdump_dir`.

10 **When you have collected all debug data, perform the following steps to restore the service.**

a. **Stop Calendar Server.**

```
cd cal-svr-base/cal/sbin
```

```
./stop-cal
```

b. **Make sure that all Calendar Server processes stopped.**

Wait one minute, then kill any remaining processes.

c. **Clean the temporary shared database files.**

```
cd caldb.berkeleydb.homedir.path;
rm __db.00*
```

where *caldb.berkeleydb.homedir.path* is the path of the database, which is specified in the *caldb.berkeleydb.homedir.path* parameter in the `ics.conf` file.

d. **Restart Calendar Server.**

```
./start-cal
```

e. **After restarting the services, check the logs for any unexpected behavior.**

▼ To Collect Debug Data on a Calendar Server Crashed Process

Use this task to collect data when a Calendar Server process has stopped (crashed) unexpectedly.

On Solaris systems, you can easily gather the required data by running the `cscapture` command.

```
./cscapture
```

For more information about running `cscapture`, see *Using Calendar Server Capture (cscapture) to Collect Debug Data for Sun Java System Calendar Server*.

The `cscapture` command gathers `ps`, `swap`, `gcore`, `pkg_app`, and other data.

On all other platforms, manually run all the commands on the actual machine where the core file(s) were generated, as described in the following procedure.

1. **Collect the general system information as explained in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.**

2. **Note whether you can you restart Calendar Server. If the problem is reproducible, provide a test case that can be reproduced in the Sun Support Center labs.**

3. **Get the output of the following commands.**

HP-UX	`ps -ef | grep` *cal-svr-base*
	`vmstat 5 5`
	`iostat -x`
	`top`
	`sar`

Linux	`ps -aux | grep` *cal-svr-base*
	`vmstat 5 5`
	`top`
	`uptime`
	`sar`

Windows	Obtain the `CALENDAR` process PID: `C:\`*windbg-root*`>tlist.exe` Obtain process details of the `CALENDAR` running process PID: `C:\`*windbg-root*`>tlist.exe` *calendar-pid*

Note – To use the preceding commands on Windows systems, you need to install the debugging tools, available from the following url:

`http://www.microsoft.com/whdc/devtools/debugging/default.mspx`

Install the latest version of the debugging tools and OS symbols for your version of Windows.

You also must add the environment variable `"_NT_SYMBOL_PATH"`.

4. **Get the swap information.**

HP-UX	`swapinfo`

Linux	`free`
Windows	Already provided in `C:\report.txt` as described in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.

5 Get the system logs.

Linux	`/var/adm/messages` `/var/log/syslog`
HP-UX	`/var/adm/syslog/syslog.log`
Windows	Event log files: Start-> Settings-> Control Panel —> Event Viewer-> Select Log Then click Action-> Save log file as

6 Get core files (called "Crash Dumps" by Windows).

Solaris OS

See "1.6 Configuring Solaris OS to Generate Core Files" on page 23 if a core file was not generated.

Linux

Core dumps are turned off by default in the `/etc/profile` file. You can make per user changes by editing your `~/.bash_profile` file. Look for the following line:

```
ulimit -S -c 0 > /dev/null 2>&1
```

You can either comment out the entire line to set no limit on the size of the core files or set your own maximum size.

Windows

Generate a crash dump during a crash of Calendar Server by using the following commands:

Get the CALENDAR process PID :
`C:\`*windbg-root*`>tlist.exe`
Generate a crash dump when the CALENDAR process crashes:
`C:\`*windbg-root*`>adplus.vbs -crash -FullOnFirst -p` *calendar-pid* `-o C:\crashdump_dir`

The `adplus.vbs` command watches *calendar-pid* until it crashes and will generate the `dmp` file. Provide the complete generated folder under `C:\`*crashdump_dir*.

Note – If you didn't install the Debugging Tools for Windows, you can use the `drwtsn32.exe -i` command to select Dr. Watson as the default debugger. Use the `drwtsn32.exe` command, check all options, and choose the path for crash dumps. Then provide the dump and the `drwtsn32.log` files.

▼ To Collect Debug Data on a Calendar Server Database Problem

Use this task to collect data when Calendar Server is experiencing a database problem in a standard database deployment.

A Calendar Server database problem can occur when there are errors in the Calendar Server backup database (obtained with the `csbackup` command) or when error messages in the logs identify problems in the integrity of the database.

1. **Collect the general system information as explained in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.**

2. **(Solaris only) Run the `cscapture` script to gather a copy of the Calendar Server's database and additional information:**

   ```
   ./cscapture -db
   ```

3. **(Other platforms) Run the `capture_environment.pl` script.**

   First go to the directory where the script is located. For example:

   ```
   cd /export/home/myscripts
   perl capture_environment.pl
   ```

   This command creates a file named `archive.tar.gz`, which contains Calendar Server database logs, system statistics, and configuration information, and database information.

4. **Get the output from the `csdb check` command.**

   The `csdb check` command reports errors on the Calendar Server backup database and on database corruption issues that turn up in the error logs.

   Note – This command can return false errors when it is run on a production server database.

UNIX and Linux	`cd` *cal-svr-base*`/cal/bin`
	`./csdb check`

Windows	`cd` *cal-svr-base*`\cal\bin`
	`csdb.exe check`

5. **Try to recover your database from a backup.**

   Read the section, "Checking and Rebuilding a Calendar Database" or "To Rebuild the Calendar Databases (caldb)" in the *Calendar Server 6 Administration Guide*:

   - Calendar Server 6.3: *Sun Java System Calendar Server 6.3 Administration Guide* http://docs.sun.com/app/docs/doc/819–4654/6n6prj59n?a=view

- Calendar Server 6.2 2005Q4: Chapter 16, "Administering Calendar Server Databases with csdb," in *Sun Java System Calendar Server 6 2005Q4 Administration Guide*

  http://docs.sun.com/app/docs/doc/819–2433/6n4nlfjs1?a=view
- Calendar Server 6.2 2005Q1: *Sun Java System Calendar Server 6 2005Q1 Administration Guide*

  http://docs.sun.com/source/819–0024/csagDatabases.html
- Calendar Server 6.1 2004Q2: *Sun Java Sytem Calendar Server 6 2004Q2 Administration Guide*

  http://docs.sun.com/source/817–5697/csagDatabases.html
- Calendar Server 6.0 2003Q4:

  http://docs.sun.com/source/816-6708–10/csag5.html

6 **If the database is corrupted, and you have tried to recover it without success, provide a tar file of the entire database directory.**

The path of the database is specified in the *caldb.berkeleydb.homedir.path* parameter in the `ics.conf` file.

For a Windows system, the following example shows the physical path of the database event, task, and alarm files:

```
caldb.berkeleydb.homedir.path =
"C:/www/Cal6-1-1/CalendarServer6/var/csdb"
```

7 **Obtain user information for at least 2 users impacted by the problem and 2 users unaffected by the problem.**

If no users can connect to the database, please include this information.

Use the `csattribute` command to get the LDIF entry of each user. The `csattribute` command displays the user's DN and the attributes of the specified user. following is a sample DN:

```
uid=user1,ou=people,o=siroe.com,o=isp
```

Run the command as follows:

UNIX and Linux	*cal-svr-base*`/cal/sbin/csattribute -v list` *uid*
Windows	`csattribute -v list` *uid*

where:

server-root The directory on the Calendar Server machine dedicated to holding the Calendar Server program, and configuration, maintenance, and information files. The default location for UNIX and Linux versions of Calendar Server is `/var/opt/mps/serverroot/`.

uid User ID of the user you are searching for.

8 **Get the LDIF entry of the domain where the impacted user resides.**

Use the domain portion of the DN shown by the `csattribute` command in the preceding step.

UNIX and Linux

```
dir-root/shared/bin/ldapsearch -h hostname -p port -D
"cn=Directory Manager" -w password -s base -b "baseDN"
"(objectclass=*)" > /tmp/domain.ldif
```

Windows

```
dir-root\shared\bin\ldapsearch.exe -h hostname -p port -D
"cn=Directory Manager" -w password -s base -b "baseDN"
"(objectclass=*)" > C:\domain.ldif
```

where:

dir-root The directory on the Directory Server machine dedicated to holding the server program, and configuration, maintenance, and information files. The default location for UNIX and Linux versions of Calendar Server is `/var/opt/mps/serverroot/`.

hostname Name of the host running Directory Server. You can omit `-h` *hostname* if the Directory Server is running locally.

port Port number on which Directory Server is listening. The default is 389. You can omit *port* if the Directory Server is running on port 389.

9 **If you have not already done so, set the Calendar Server debug log level to `Debug` and provide all errors printed to the Calendar Server logs.**

Provide all log files with pointers to timestamps or error messages.

10 **If you have not already done so, provide specific steps showing how to reproduce the problem.**

Include any observed error messages from the Web UI, WCAP commands, and the Calendar Server log files and timestamps showing when the error occurred.

▼ To Collect Debug Data on a Calendar Server CLD/DWP Problem

Use this task to collect data when Calendar Server is experiencing a database problem in a Calendar Lookup Database/Database Wire Protocol (CLD/DWP) deployment.

A Calendar Server CLD/DWP installation comprises a multi-tiered deployment with multiple servers connecting to a distributed back-end Calendar Server database. If a database problem occurs in this environment, you must provide detailed information about your network topology and front-end and back-end servers to facilitate the identification of the issue.

1 **Collect the general system information as explained in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.**

2 **Provide a detailed explanation of how you have set up your Calendar Server deployment. For example, you could have**

- Two front-end servers and two back-end servers
- Two front-end servers and one back-end server
- Three front-end servers and three back-end servers

and so on.

3 **Provide the specific name of each machine in your deployment. Identify whether it is a front-end server or back-end server.**

4 **(Solaris only) Run the `cscapture` script to gather Calendar Server information:**

```
./cscapture
```

5 **(Other platforms) Run the `capture_environment.pl` script.**

First go to the directory where the script is located. For example:

```
cd /export/home/myscripts
perl capture_environment.pl
```

This command creates a file named `archive.tar.gz`, which contains Calendar Server database logs, system statistics, and configuration information, and database information.

6 **Obtain user information for at least 2 users impacted by the problem and 2 users unaffected by the problem.**

If no users can connect to the database, please include this information.

Use the `csattribute` command to get the LDIF entry of each user. The `csattribute` command displays the user's DN and the attributes of the specified user. following is a sample DN:

```
uid=user1,ou=people,o=siroe.com,o=isp
```

Run the command as follows:

UNIX and Linux	*cal-svr-base*`/cal/sbin/csattribute -v list` *uid*
Windows	`csattribute -v list` *uid*

where:

server-root The directory on the Calendar Server machine dedicated to holding the Calendar Server program, and configuration, maintenance, and information files. The default location for UNIX and Linux versions of Calendar Server is `/var/opt/mps/serverroot/`.

uid User ID of the user you are searching for.

7 **Get the LDIF entry of the domain where the impacted user resides.**

Use the domain portion of the DN shown by the `csattribute` command in the preceding step.

UNIX and Linux

```
dir-root/shared/bin/ldapsearch -h hostname -p port -D
"cn=Directory Manager" -w password -s base -b "baseDN"
"(objectclass=*)" > /tmp/domain.ldif
```

Windows

```
dir-root\shared\bin\ldapsearch.exe -h hostname -p port -D
"cn=Directory Manager" -w password -s base -b "baseDN"
"(objectclass=*)" > C:\domain.ldif
```

where:

dir-root The directory on the Directory Server machine dedicated to holding the server program, and configuration, maintenance, and information files. The default location for UNIX and Linux versions of Calendar Server is `/var/opt/mps/serverroot/`.

hostname Name of the host running Directory Server. You can omit `-h` *hostname* if the Directory Server is running locally.

port Port number on which Directory Server is listening. The default is 389. You can omit *port* if the Directory Server is running on port 389.

8 **If you have not already done so, set the Calendar Server debug log level to `Debug` and provide all errors printed to the Calendar Server logs.**

Provide all log files with pointers to timestamps or error messages. For example:

```
[04/Jul/2006:19:49:52 +0200] mouline cshttpd[25278]:
General Error: chttp_ctx_async_connect: ASock_Connect
failed: pctx->s 924500

[04/Jul/2006:19:49:52 +0200] mouline cshttpd[25278]:
General Notice: CldCacheInit: attemptint to open cache
database for 25278
```

9 **If you have not already done so, provide specific steps showing how to reproduce the problem.**

Include any observed error messages from the Web UI, WCAP commands, and the Calendar Server log files and timestamps showing when the error occurred.

▼ To Collect Debug Data on a Calendar Express or Communications Express Interface Problem

Use this task to collect data for a problem with the either the Calendar Express or Communications Express interfaces. The most common problems are related to incorrect translation of fields when using a localized Calendar Server interface.

1. **Collect the general system information as explained in "To Collect Required Debug Data for Any Calendar Server Problem" on page 7.**

2. **Take a snapshot of the problematic screen(s).**

3. **Note the step-by-step procedure to reproduce the problem. Include a test case.**

Note – The Sun Support Center does not support Webmail customizations. Contact your sales representative for those problems.

1.6 Configuring Solaris OS to Generate Core Files

Core files are generated when a process or application terminates abnormally. Core files are managed with the `coreadm` command. This section describes how to use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a Solaris OS process or daemon terminates abnormally.

Before configuring your system for core files, make sure that the `/var` file system has sufficient space. Once you configure Solaris OS to generate core files, from now on all processes that crash will write a core file to the `/var/cores` directory.

▼ To Configure Solaris OS to Generate Core Files

1. **Run the following commands as `root`.**

```
mkdir -p /var/cores
coreadm -g /var/cores/%f.%n.%p.%t.core -e global -e global-setid \
-e log -d process -d proc-setid
```

In this command:

`-g` Specifies the global core file name pattern. Unless a per-process pattern or setting overrides it, core files are stored in the specified directory with a name such as *program*.*node*.*pid*.*time*.`core`, for example: `mytest.myhost.1234.1102010309.core`.

`-e` Specifies options to enable. The preceding command enables:

- Use of the global (that is, system-wide) core file name pattern (and thereby location)
- Capability of `setuid` programs to also dump core as per the same pattern
- Generation of a syslog message by any attempt to dump core (successful or not)

`-d` Specifies options to disable. The preceding command disables:

- Core dumps per the per-process core file pattern
- Per-process core dumps of `setuid` programs

The preceding command stores all core dumps in a central location with names identifying what process dumped core and when. These changes only impact processes started after you run the `coreadm` command. Use the `coreadm -u` command after the preceding command to apply the settings to all existing processes.

2 Display the core configuration.

```
# coreadmglobal core file pattern: /var/cores/%f.%n.%p.%t.core
      init core file pattern: core
           global core dumps: enabled
      per-process core dumps: disabled
     global setid core dumps: enabled
per-process setid core dumps: disabled
    global core dump logging: enabled
```

See the `coreadm` man page for further information.

3 Set the size of the core dumps to unlimited.

```
# ulimit -c unlimited
# ulimit -a

        coredump(blocks) unlimited
```

See the `ulimit` man page for further information.

4 Verify core file creation.

```
# cd /var/cores
# sleep 100000 &
[1] PID
# kill -8 PID
# ls
```

1.7 Running the Calendar Server Debugging Scripts

This section describes how to run the `cscapture` and `pkg_app` scripts.

▼ To Run the CS Capture Script

● **On Solaris platforms, run `./cscapture`.**

The CS Capture script, `cscapture`, gathers information about your Calendar Server environment to help debug crashes, hangs, performance issues, and database problems.

The gathered data is compressed into a tar file ready to be uploaded to Sun. CS Capture also incorporates the `pkg_app` script, described in the following procedure, and gathers `pkg_app` data. Currently, the CS Capture script is only available on Solaris platforms and replaces the `capture_environment` script. For details about running `cscapture`, see *Using Calendar Server Capture (cscapture) to Collect Debug Data for Sun Java System Calendar Server*.

▼ To Run the pkg_app Script

This script packages an executable and all of its shared libraries into one compressed tar file given the PID of the application and optionally the name of the core file to be opened. The files are stripped of their directory paths and are stored under a relative directory named `app/` with their name only, allowing them to be unpacked in one directory.

On Solaris 9 OS or greater, the list of files is derived from the core file rather than the process image if it is specified. You still must provide the PID of the running application to assist in path resolution.

Two scripts are created to facilitate opening the core file when the tar file is unpacked:

- `opencore`. This is the script to be executed once unpacked. It sets the name of the core file and the linker path to use the `app/` subdirectory and then invokes `dbx` with the `dbxrc` file as the argument.
- `dbxrc`. This script contains the `dbx` initialization commands to open the core file.

1. **Copy the script to a temporary directory on the system where Calendar Server is installed.**

2. **Become superuser.**

3. **Execute the `pkg_app` script in one of the following three ways:**

   - `./pkg_app` *pid-of-running-application corefile*

- `./pkg_app` *pid-of-the-running-application*
  (The *pkg_app* scripts prompts for the *corefile* name.)
- `./pkg_app` *core file*

1.8 Reporting Problems

Use the following email aliases to report problems with this document and its associated scripts:

- To provide feedback: gdd-feedback@sun.com
- To report problems: gdd-issue-tracker@sun.com

1.9 Accessing Sun Resources Online

The docs.sun.com web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

1.10 Third-Party Web Site References

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

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused 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.

1.11 Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 805-6005-10.