



Sun Gathering Debug Data for Sun Java System Directory Server 5

Sun Java™ Enterprise System Technical Note



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

Part No: 820-0437
January 2007

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

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, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks 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.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux États-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-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 États-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.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des États-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

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

Sun Gathering Debug Data for Sun Java System Directory 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 Sun Java™ System Directory Server software.

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 Directory Server problems, see:

<http://www.sun.com/services/gdd/index.html>

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

This technical note contains the following sections:

- “1.1 Technical Note Revision History” on page 4
- “1.2 About This Technical Note” on page 4
- “1.3 Overview of Collecting Debug Data For Directory Server” on page 5
- “1.4 Creating a Service Request with the Sun Support Center” on page 5
- “1.5 What Directory Server Debug Data Should You Collect?” on page 6
- “1.6 Configuring the Operating System to Generate Core Files” on page 16
- “1.7 Reporting Problems” on page 20
- “1.9 Third-Party Web Site References” on page 20
- “1.11 Sun Welcomes Your Comments” on page 21

1.1 Technical Note Revision History

Version	Date	Description of Changes
1.0	December 2006	Initial release of this technical note.
1.1	January 2007	Updated “ To Configure Solaris OS to Generate Core Files ” on page 16.

1.2 About This Technical Note

This document covers Sun Java System Directory Server 5 on all supported platforms.

You can use this document in all types of environments, including test, pre-production, and production. Verbose debugging is not used so as to avoid performance impact, except when it is deemed necessary. In some cases, 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 Directory Server problems.

If your problem does not conveniently fit into any of the specific categories, supply the general information described in “[To Collect Required Debug Data For Any Directory Server Problem](#)” on page 6 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 Directory Server Debug Data

The prerequisites for collecting debug data for Directory Server are as follows.

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

<http://www.sun.com/bigadmin/scripts/indexSjs.html>

See “[To Run the pkg_app Script](#)” on page 19 for instructions on using `pkg_app`. See the documentation bundled with `dirtracer` for information on using that script.

- 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.aspx>

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.

server-root The base file system location for Directory Server or Directory Server.

win-dbg-root The base file system location for Windows Debugging Tools.

Many paths specified in this document use the forward slash format of UNIX. If you are running software on a Windows system, use the equivalent backslash format.

1.3 Overview of Collecting Debug Data For Directory Server

Collecting debug data for any Directory Server problem involves these basic operations:

1. Collecting basic problem and system information.
2. Collecting specific problem information.
3. Creating a tar .gz file of all the information and uploading the file to 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>

Note – When opening a Service Request by phone with the Sun Support Center, provide a summary of the problem. Also provide 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.

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

1.5 What Directory Server Debug Data Should You Collect?

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

This section contains the following tasks:

- “To Collect Required Debug Data For Any Directory Server Problem” on page 6
- “To Collect Required Debug Data For Directory Server Installation Problems” on page 8
- “To Collect Required Debug Data For an Unresponsive or Hung Directory Server Process” on page 9
- “To Collect Required Debug Data For a Crashed Directory Server Process” on page 12
- “To Collect Required Debug Data For Directory Server Replication Problems” on page 14
- “To Collect Required Debug Data For Directory Server Schema Problems” on page 15

▼ To Collect Required Debug Data For Any Directory 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 time or times the problem occurred.**2 If possible, collect explorer output from SUNWexpl0 software on the system where the problem occurred.****3 Provide graphical representation of your deployment.**

The graphical representation of your deployment is key to understanding the context of the problem. Show the following in the graphical representation.

- All computers involved, with their IP addresses, hostnames, roles in the deployment, operating systems, and versions used.
- All other relevant systems, including load balancers, firewalls, and so forth.

4 If possible, collect a copy of the database on spare disk space or backup media to be used later if necessary.

Database files and the transaction log are often indispensable for diagnosing deadlocks involving database locks.

5 Note the operating system version.

Solaris OS `uname -a`

HP-UX `uname -r`

Red Hat `cat /etc/redhat-release`

Windows `C:\Program Files\Common files\Microsoft Shared\MSInfo\msinfo32.exe /report C:\report.txt`

6 Note the patch level.

Solaris OS `showrev -p`

HP-UX `swlist`

Red Hat `rpm -qa`

Windows Already provided in `C:\report.txt`.

7 Collect Directory Server version information.

`server-root/bin/slapd/server/ns-slapd -D instance-dir -V`

The server executable is `slapd.exe` on Windows systems.

8 Collect the Directory Server configuration file, `server-root/slapd-serverID/config/dse.ldif`.**9 Collect Directory Server access, errors, and audit logs.**

By default, you find these logs in the following locations:

```
server-root/slapd-serverID/logs/access
```

```
server-root/slapd-serverID/logs/errors
```

```
server-root/slapd-serverID/logs/audit (if enabled)
```

If these log files are not in the default locations, examine the Directory Server configuration file, `server-root/slapd-/serverID/config/dse.ldif`, to find the paths to the logs. The paths are specified as the values of attributes `nsslapd-accesslog`, `nsslapd-errorlog`, and `nsslapd-auditlog`.

▼ To Collect Required Debug Data For Directory Server Installation Problems

This section describes what data to collect when you cannot complete Directory Server installation.

Before You Begin If the problem concerns a general installation failure for Java Enterprise System, first check the installation troubleshooting chapter in the *Installation Guide* for your version of Java Enterprise System.

1 For compressed archive installations, collect installation output showing system calls.

Reinstall while using the appropriate command for your system from the following list. Collect the output of the command that is displayed during installation.

```
Solaris OS    truss -ealf -rall -wall -vall -o /tmp/install-directory-truss.out
              ./setup
```

```
HP-UX        tusc -v -fealT -rall -wall -o /tmp/install-directory-tusc.out
              ./setup
```

```
Red Hat      strace -fv -o /tmp/install-directory-strace.out ./setup
```

```
Windows     Use DebugView.
```

DebugView is available at
<http://www.sysinternals.com/Utilities/DebugView.html>.

2 For Java Enterprise System installations, collect installation error logs.

The log file is named after the date and time that the installation failed. For example, a log file for an installation that failed on December 16 at 3:32 p.m. would have a name like `Java_Enterprise_System*_install.B12161532`.

On Solaris systems, installation logs are located under `/var/sadm/install/logs`.

On Red Hat and HP-UX systems, installation logs are located under `/var/opt/sun/install/logs`.

On Windows systems, installation logs are located under `C:\Documents and Settings\current-user\Local Settings\Temp`.

▼ To Collect Required Debug Data For an Unresponsive or Hung Directory Server Process

This procedure describes what data to collect when Directory Server is still running, but is no longer responding to client application requests.

Collect the data describe in this procedure *while the server is hanging*.

- 1 **Note the time during which the hang is seen to occur.**
- 2 **Collect information about the port used during the hang.**
- 3 **Collect statistics about the system running Directory Server.**

Solaris OS `ps -aux | grep server-root`

`vmstat 5 5`

`iostat -x`

`top`

`uptime`

HP-UX `ps -aux | grep server-root`

`vmstat 5 5`

`iostat -x`

`top`

`sar`

Red Hat `ps -aux | grep server-root`

`vmstat 5 5`

top

uptime

sar

Windows Get the process ID using the `tlst.exe` command, then get process details using the same command.

win-dbg-root\tlst.exe pid

4 Collect swap information.

Solaris OS `swap -l`

HP-UX `swapinfo`

Red Hat `free`

Windows Already provided in `C:\report.txt`.

5 On Solaris systems, collect output from `pstack` and `pmap` five times, once every ten seconds.

`pstack pid`

`pmap -x pid`

6 Get output showing system calls during the hang, by letting each of the commands listed here run for about a minute, then stopping them by typing Control-C.

Solaris OS `truss -ealf -rall -wall -vall -o /tmp/truss.out -p pid`

HP-UX `tusc -v -fealT -rall -wall -o /tmp/truss.out -p pid`

Red Hat `strace -fv -o /tmp/strace.out -p pid`

Windows Use Debug View.

DebugView is available at

<http://www.sysinternals.com/Utilities/DebugView.html>.

7 Collect core files or crash dumps, and related command output.

When the server is hanging, attempt to get several core files that show the state of the server threads over time. You can do this by generating a core file, changing the name of the core file, waiting 30 seconds to a minute, and generating another core file. Repeat the process at least once to get a minimum of three sets of core files and related data.

Solaris OS `cd server-root/bin/slapd/server`

`gcore -o /tmp/directory-core pid`

`pstack /tmp/directory-core`

For each core file on Solaris OS, collect output from the following commands.

```
cd server-root/bin/slapd/server
```

```
file core-file
```

```
pstack core-file
```

```
pmap core-file
```

```
pflags core-file
```

For at least one of the core files on Solaris OS, collect output from the `pkg_app` script.

```
./pkg_app.ksh pid /tmp/directory-core
```

Here, *pid* is the server process ID number. See [“To Run the pkg_app Script” on page 19](#) for instructions on using `pkg_app`.

HP-UX

```
cd server-root/bin/slapd/server
```

If you have the patches PHKL_31876 and PHCO_32173 patches installed, generate the core file using the `gcore` command.

```
gcore -p pid
```

Otherwise, use the `gdb` command to generate the core file.

```
gdb
```

```
(gdb) attach pid
```

```
Attaching to process pid
```

```
No executable file name was specified
```

```
(gdb) dumpcore
```

```
Dumping core to the core file core.pid
```

```
(gdb) quit
```

```
The program is running. Quit anyway (and detach it)? (y or n) y
```

```
Detaching from program: , process pid
```

Red Hat

```
cd server-root/bin/slapd/server
```

```
gdb
```

```
(gdb) attach pid
```

```
Attaching to process pid
```

```
No executable file was specified
```

```
(gdb) gcore
```

```
Saved corefile core.pid
```

```
(gdb) backtrace
```

```
(gdb) quit
```

```
Windows win-dbg-root\tlist.exe
```

```
win-dbg-root\adplus.vbs -hang -p pid -o C:\dump-dir
```

```
Collect everything in the folder under C:\dump-dir.
```

8 Collect output from `idsktune`.

The `idsktune` command provides information on system parameters, patch level, tuning recommendations, and so forth. The command is described in the product documentation.

```
server-root/bin/slapd/server/idsktune
```

▼ To Collect Required Debug Data For a Crashed Directory Server Process

This procedure describes what data to collect when a Directory Server process unexpectedly dies.

1 Try to restart Directory Server and record the results.

2 Collect statistics about the system running Directory Server.

```
Solaris OS ps -aux | grep server-root
```

```
vmstat 5 5
```

```
iostat -x
```

```
top
```

```
uptime
```

```
HP-UX ps -aux | grep server-root
```

	<code>vmstat 5 5</code>
	<code>iostat -x</code>
	<code>top</code>
	<code>sar</code>
Red Hat	<code>ps -aux grep server-root</code>
	<code>vmstat 5 5</code>
	<code>top</code>
	<code>uptime</code>
	<code>sar</code>
Windows	Get the process ID using the <code>tlst.exe</code> command, then get process details using the same command.
	<code>win-dbg-root\tlst.exe pid</code>

3 Collect swap information.

Solaris OS	<code>swap -l</code>
HP-UX	<code>swapinfo</code>
Red Hat	<code>free</code>
Windows	Already provided in <code>C:\report.txt</code> .

4 Collect system logs.

Solaris OS	<code>/var/adm/messages</code>
	<code>/var/log/syslog</code>
HP-UX	<code>/var/adm/syslog/syslog.log</code>
Red Hat	<code>/var/log/messages</code>
	<code>/var/log/syslog</code>
Windows	Retrieve the Event log files.
	From the Control Panel, open the Event Viewer. Select the event log. Then select Action > Save log file as to save the log.

5 Collect core files.

For instructions on preparing your system to produce core files or crash dumps in the event of a crash, see [“1.6 Configuring the Operating System to Generate Core Files”](#) on page 16.

a. For each core file on Solaris OS, collect output from the following commands.

```
cd server-root/bin/slapd/server
```

```
file core-file
```

```
pstack core-file
```

```
pmap core-file
```

```
pflags core-file
```

b. For at least one of the core files on Solaris OS, collect output from the pkg_app script.

```
./pkg_app.ksh pid core-file
```

Here, *pid* is the server process ID number. See [“To Run the pkg_app Script”](#) on page 19 for instructions on using pkg_app.

▼ To Collect Required Debug Data For Directory Server Replication Problems

This procedure describes what data to collect when experiencing inconsistencies and other issues involving Directory Server replication.

Directory Server commands used here are described in the product documentation.

- 1 Collect the hostname, IP address, and *serverID* for each server instance in the replication topology.**
- 2 Collect a copy of the schema folder, *server-root/slapd-serverID/config/schema*, and all the files in the folder.**
- 3 Collect access logs configured to show replication information.**

Before you collect access logs as described in [“To Collect Required Debug Data For Any Directory Server Problem”](#) on page 6, adjust the log level to keep replication information.

You can use the Console to adjust the access log level.

Alternatively, you can use the `ldapmodify` command as follows.

```
$ ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
```

```
replace: nsslapd-infolog-area # nsslapd-errorlog-level in 5.1
nsslapd-infolog-area: 8192
```

To return to the default log level, use the following command.

```
$ ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-infolog-area # nsslapd-errorlog-level in 5.1
nsslapd-infolog-area: 0
```

You must restart Directory Server for the change to take effect.

4 Provide a listing of the changelog directory.

UNIX and Linux `ls -la changelog-dir`

Windows `dir changelog-dir`

If you cannot find the change log, examine the Directory Server configuration file, `server-root/slapd-serverID/config/dse.ldif`, to find the path. The path is specified as the values of attribute `nsslapd-changelogdir`.

5 Collect output from the `insync` command.

The `insync` command indicates the state of synchronization between a master replica and one or more consumer replicas. The following command shows the state over a period of 30 seconds.

```
server-root/shared/bin/insync -s "cn=Directory
Manager:password@hostname1:ldap-port" -c "cn=Directory
Manager:password@hostname2:ldap-port" 30
```

6 Collect output from the `repldisc` command.

The `repldisc` command displays the replication topology, building a graph of all known replica, then showing the result as a matrix.

```
server-root/shared/bin/repldisc -D "cn=Directory Manager" -w password -b base-dn -s
host:ldap-port
```

Here, `base-dn` is the DN of the replicated suffix.

▼ To Collect Required Debug Data For Directory Server Schema Problems

This procedure describes what data to collect when experiencing Directory Server schema violation errors.

- 1 **Collect a copy of the schema folder**, `server-root/slapd-serverID/config/schema`, **and all the files in the folder.**
- 2 **Indicate whether you have the same problem both in the Console and on the command line.**
- 3 **Provide a list of the last changes made to the schema files.**
- 4 **If the schema violation occurred during an add, modify, delete, or replace operation, provide an LDIF representation of the changes and a list of the commands used.**

1.6 Configuring the Operating System to Generate Core Files

Core files and crash dumps are generated when a process or application terminates abnormally. You can also generate core files and crash dumps to help diagnose why a process does not respond to client application requests.

This section includes the following procedures:

- [“To Configure Solaris OS to Generate Core Files” on page 16](#)
- [“To Configure Red Hat Linux to Generate Core Files” on page 18](#)
- [“To Configure Windows to Generate Crash Dumps” on page 18](#)
- [“To Run the pkg_app Script” on page 19](#)

▼ To Configure Solaris OS to Generate Core Files

This procedure shows you how to use the `coreadm` command to configure the system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining core files in a specific directory whenever a Solaris OS process or daemon terminates abnormally.

Before You Begin Make sure that the `/var` file system where the core files are generated has sufficient space. Once you configure Solaris OS to generate core files as shown here, all processes that crash write a core file to the `/var/cores` directory.

- 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.

```
# coreadm global 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 file size as large as possible, using the `ulimit` command.

```
# ulimit -c unlimited
# ulimit -a
   coredump(blocks) unlimited
```

See the `ulimit(1)` man page for details.

4 Verify that applications can in fact generate core files.

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

You should see a core file for the `sleep` process you killed.

▼ To Configure Red Hat Linux to Generate Core Files

On Red Hat systems, you can enable core files to be generated on a per user basis.

- 1 **Open the `~/.bash_profile` file for the server user in a text editor.**
- 2 **Search for a line using the `ulimit` command as follows.**

```
ulimit -S -c 0 > /dev/null 2>&1
```
- 3 **Either comment out the line, or set your own limit for core file size.**

▼ To Configure Windows to Generate Crash Dumps

The native debug tool on Windows systems, Dr. Watson, allows you to generate crash dumps.

However, Dr. Watson does not allow generation of crash dumps on a running process. To generate crash dumps from a running process, install the Debugging Tools. The Debugging Tools are freely available from the Windows web site at

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

- 1 **You can use Dr. Watson for crash dumps generated when a process dies.**
 - a. **Use the `drwtsn32 -i` command to make Dr. Watson the default debugger.**
 - b. **Open Dr. Watson with the `drwtsn32 -i` command.**
 - c. **Check all options.**
 - d. **Choose the path where crash dumps are generated.**
When providing crash dumps, collect both the `dmp` and `drwtsn32.log` files.
- 2 **Use the Window Debugging Tools to generate crash dumps of a running process.**
 - a. **Make sure you install the latest version of the Debugging Tools and OS Symbols for your version of Windows.**
 - b. **Set the `_NT_SYMBOL_PATH` for your environment.**
- 3 **Enable generation of a crash dump for your application.**
Get the process ID of the application using the `tlst.exe` command, then enable the crash dump.

```
win-dbg-root\tlist.exe
```

```
win-dbg-root\adplus.vbs -crash -FullOnFirst -p pid -o C:\dump-dir
```

The `adplus.vbs` command tracks the application with process ID *pid*. The `adplus.vbs` command generates a `dmp` file in the event of a crash.

- 4 **When collecting crash dump information, take the complete folder generated under `C:\dump-dir`.**

▼ To Run the `pkg_app` Script

The `pkg_app` script packages an executable and all of its shared libraries into one compressed tar file given the process ID 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 names only, allowing them to be unpacked in one directory.

On Solaris 9 and Solaris 10, the list of files is derived from the core file rather than the process image if it is specified. You must still provide the process ID 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/` directory, 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 the server is installed.**
- 2 **Become superuser.**
- 3 **Run the `pkg_app` script in one of the following ways.**

- `./pkg_app server-pid core-file`
- `./pkg_app server-pid`
The `pkg_app` script prompts for the `core-file` name.
- `./pkg_app core-file`

1.7 Reporting Problems

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

- gdd-feedback@sun.com For feedback on this document
gdd-issue-tracker@sun.com To report problems in gathering debug data

1.8 Documentation, Support, and Training

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

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

1.9 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.

1.10 Searching Sun Product Documentation

Besides searching for Sun product documentation from the docs.sun.com web site, you can use a search engine of your choice by typing the following syntax in the search field:

```
search-term site:docs.sun.com
```

For example, to search for Directory Server, type the following:

```
"Directory Server" site:docs.sun.com
```

To include other Sun web sites in your search, such as java.sun.com, www.sun.com, and developers.sun.com, use sun . com in place of docs . sun . com in the search field.

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 820-0437-10.

