



Sun Gathering Debug Data for Sun Java System Portal Server

Sun Java™ Enterprise System Technical Note



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

Part No: 819-5489-11
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 CONTREFAÇON.

Sun Gathering Debug Data for Sun Java System Portal 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 Portal 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 Portal 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 Portal Server with the Sun Support Center.

This technical note contain 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 Portal Server” on page 5
- “1.4 Creating a Service Request with the Sun Support Center” on page 6
- “1.5 What Portal Server Debug Data Should You Collect?” on page 6
- “1.6 Configuring Solaris OS to Generate Core Files” on page 17
- “1.7 Running the Portal Server Debugging Scripts” on page 19
- “1.8 Reporting Problems” on page 20
- “1.9 Accessing Sun Resources Online” on page 21

1.1 Technical Note Revision History

Version	Date	Description of Changes
11	January 2007	Updated “To Configure Solaris OS to Generate Core Files” on page 18.
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 Portal Server on the Solaris™ Operating System, HP-UX, Linux, and Microsoft Windows platforms:

- Sun Java System Portal Server 6 2005Q4
- Sun Java System Portal Server 6 2005Q1
- Sun Java System Portal Server 6 2004Q2
- Sun Java System Portal Server 6 2003Q4
- Sun ONE Portal Server 6.0

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 Portal Server problems.

If your problem does not conveniently fit into any of the specific categories, supply the general information described in [“1.5 What Portal Server Debug Data Should You Collect?”](#) 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 Portal Server Debug Data

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

- Make sure you have superuser privileges.
- For the Solaris OS platform, obtain the `ps6info.sh` and `pkg_app` scripts from the following location:

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

- 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.

- *instance-dir*: The directory on the Portal Server machine dedicated to holding configuration, maintenance, and information files for a specific instance. This directory is located under *server-root*.
- *server-root*: The directory on the Portal Server machine dedicated to holding the server program, configuration, maintenance, and information files.
- *web-pid*: Process ID of the Web Server daemon.

1.3 Overview of Collecting Debug Data for Portal Server

Collecting debug data for a Portal 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 Portal 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 Portal Server Problem” on page 7
- “To Collect Debug Data on Portal Server Installation Problems” on page 8
- “To Collect Debug Data on a Hung or Unresponsive Portal Server Process” on page 9
- “To Collect Debug Data on a Portal Server Crashed Process” on page 15

▼ To Collect Required Debug Data for Any Portal 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.

For problems with Portal Server Secure Remote Access (gateway), you need to collect data from both the portal and gateway hosts if they are separate (the usual configuration in a production environment). If possible, provide the output from Sun Explorer Data Collector (SUNWexplo) of the machine where the problem occurred.

- 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.**
- 3 **For Solaris OS systems, use the `ps6info.sh` script to gather all the necessary information. For HP-UX, Linux, and Windows platforms, or if you do not have the `ps6info.sh` script, continue with the remaining steps.**

- 4 **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`

- 5 **Note the patch level.**

Solaris OS `patchadd -p`

HP-UX `swlist`

Linux `rpm -qa`

Windows Already provided in the `C:\report.txt` file above.

- 6 **Get the `/etc/opt/SUNWps/.version` file (or `.version-sra` for the Portal Server Secure Remote Access).**
- 7 **Note the web container (Sun Java System Web Server, Sun Java System Application Server, BEA WebLogic, or IBM WebSphere).**

8 Get the log files.

UNIX and Linux `/var/opt/SUNWps/*`

Windows `server-root\instance-dir\portal\logs*`

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.

9 Get the configuration file.

UNIX and Linux `more /etc/opt/SUNWps/.version` (or `.version-sra` for Secure Remote Access)

Windows `more server-root\.version` (or `.version-sra` for Secure Remote Access)

▼ To Collect Debug Data on Portal Server Installation Problems

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

1 Consult the following troubleshooting information:

- Sun Java System Portal Server 2005Q4 :
Chapter 9, “Troubleshooting,” in *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX*
- Sun Java System Portal Server 2005Q1:
Chapter 13, “Troubleshooting,” in *Sun Java Enterprise System 2005Q1 Installation Guide*
- Sun Java System Portal Server 2004Q2:
Chapter 11, “Troubleshooting,” in *Sun Java Enterprise System 2004Q2 Installation Guide*
- Sun ONE Portal Server:
Chapter 9, “Troubleshooting Installation Problems,” in *Sun Java Enterprise System 2003Q4 Installation Guide*

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 Portal Server Problem”](#) on page 7.
- 3 Specify if this is a first-time installation or a Hot Fix installation on a pre-existing Sun ONE Portal Server instance.
- 4 Get the installation logs.
 - On Solaris OS systems and Sun ONE Portal Server (Portal Server 6.0 and 6.1) systems, get the following logs:


```
/var/sadm/install/logs/pssetup.install.pid/*
```
 - On Windows systems, if you chose the Config Later option during the installation, provide the following log files:


```
server-root\instance_dir\portal\config\logs\*
```

5 Get the install error messages.

Solaris OS	<pre style="margin-left: 20px;">/var/sadm/install/logs</pre> <p>The log file names start with <code>Java_Enterprise_System*_install.Bdate</code><i>time</i>, where <i>date</i> and <i>time</i> correspond to the failing installing (for example, B12161532).</p>
Linux and HP-UX	<pre style="margin-left: 20px;">/var/opt/sun/install/logs</pre> <p>The log file names start with <code>Java_Enterprise_System*_install.Bdate</code><i>time</i>, where <i>date</i> and <i>time</i> correspond to the failing installing (for example, B12161532).</p>
Windows	<pre style="margin-left: 20px;">C:\DocumentsandSettings\current-user\LocalSettings\Temp</pre> <p>The log file names start with <code>MSI*.log</code> (usually a text file). The asterisk (*) represents a random number in the Temp directory for each MSI based setup.</p>

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

A process hang is defined as one of the Portal Server processes not responding to requests anymore while the process is still running locally. The Portal Server processes are:

- `appservd`: When Portal Server is hosted on Sun Java System Application Server
- `webservd`: When Portal Server is hosted on Sun Java System Web Server
- `java process`: Secure Remote Access (gateway)

- 1 Collect the general system information as explained in [“To Collect Debug Data on Portal Server Installation Problems”](#) on page 8.

2 (Secure Remote Access only) Consult the following information.

<http://sunsolve.sun.com/search/document.do?assetkey=1-25-75583-1&searchclause=75583>

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

3 (Secure Remote Access only) Can you connect to the Portal Server host when you bypass the gateway host?

If yes, the gateway java process is hung. Collect the debug data that follows on this process and not on the Portal Server container process.

4 Get the pid of the Web Server process.

Solaris OS and HP-UX `ps -ef | grep uxwdog`

The result will give you the PID of the uxwdog daemon, for example, 11449:

Linux `ps -ef | grep ns-hhttpd`

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

For example, on Solaris OS:

```
# ptree 11449
11449 ./uxwdog -d /prods/crypto/60SP6/https-sun/config
    11450 ns-httpd -d /prods/crypto/60SP6/https-sun/config
        11451 ns-httpd -d /prods/crypto/60SP6/https-sun/config
```

You want to gather data on the highest PID process, which in this example is 11451. The Web processes is either ns-hhttpd or webservd depending on the Web Server version.

5 Note the day and time that the process hang occurred.**6 Get the output of the following command.**

UNIX and Linux `netstat -an | grep web-port (or gateway-port)`

Windows `netstat -an | web-port (or gateway-port)`

7 For Solaris OS systems, the iwshang script gathers all the following debug data for you, except the output of the pkg_app script.

You must run the pkg_app script as indicated on one of core files generated by the iwshang script. Be sure to launch the iwshang script on the valid PID. For HP-UX, Linux, and Windows platforms, or if you do not have the iwshang script, continue with the remaining steps. See [“To Run the iwshang Script” on page 19](#) for more information.

8 Run the following commands and save the output.

Solaris OS	<pre>ps -ef grep server-root vmstat 5 5 iostat -x top uptime</pre>
HP-UX	<pre>ps -aux grep server-root vmstat 5 5 iostat -x top sar</pre>
Linux	<pre>ps -aux grep server-root vmstat 5 5 top uptime sar</pre>
Windows	<p>Obtain the WEB process PID: <code>C:\windbg-root>tlist.exe</code></p> <p>Obtain process details of the WEB running process PID: <code>C:\windbg-root>tlist.exe web-pid</code></p>

9 Get the swap information.

Solaris OS	<code>swap -l</code>
HP-UX	<code>swapinfo</code>
Linux	<code>free</code>
Windows	Already provided in <code>C:\report.txt</code> as described in “To Collect Debug Data on Portal Server Installation Problems” on page 8.

10 For Unix-Linux systems, if you are able to isolate the hanging process, get the following debug data for that process. Otherwise, get the following data for each of the Web Server processes. For Windows systems, get the following data for the `webservd.exe` or `ns-httpd.exe` process.

a. For Solaris OS only, using the PID obtained in Step 4, get a series of five of the following commands (one every 10 seconds):

```
pstack web-pid
pmap -x web-pid
```

b. For Solaris OS only, get the output of the following commands:

```
prstat -L -p web-pid
pmap web-pid
```

pfiles *web-pid*

11 Get the output of the following command.

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

HP-UX `tusc -v -fealT -rall -wall -o /tmpweb-pid.tusc -p web-pid`

Linux `strace -fv -o /tmp/web-pid.strace -p web-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.

12 Get the the Directory Server Access, Errors, and Audit logs used by Portal Server.

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)

13 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.

Solaris OS `cd server-root/bin/https/bin`
`gcore -o /tmp/web_process-core`
 Archive the result of the pkg_app script:
`./pkg_app.ksh PID-of-application corefile`

The output of the pkg_app script is required to analyze the core files.

Note – Make sure that you have set the size of the core dumps to unlimited by running the `ulimit` command, and that the user is not nobody. Also, check the `coreadm` command for additional control. See “[1.6 Configuring Solaris OS to Generate Core Files](#)” on page 17 if a core file isn't generated.

HP-UX

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

The file `core.web-pid` should be generated in the `https-instance/config` directory.

Linux

```
cd server-root/bin/https/bin
# gdb
(gdb) attach web-pid
Attaching to process web-pid
No executable file name was specified
(gdb) gcore
Saved corefile core.web-pid

(gdb) backtrace
(gdb) quit
```

Windows

Get the WEB process PID:

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

Generate a crash dump on the WEB running process PID:

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

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

14 Get the Access Manager configuration file.

UNIX and Linux /opt/SUNWam/lib/AMConfig.properties
Windows access-manager-server-root\lib\AMConfig.properties

15 Get the Access Manager log files.

UNIX and Linux /var/opt/SUNWam/*
Windows access-manager-server-root\debug*

16 Get network trace files between the gateway and the portal hosts, and between the client and the portal host.

Make sure that all the data collection is done over the same time frame in which you had the problem. Try to indicate the hung process if possible.

Note – Indicate clearly all IP addresses and host names for each component to correctly read these network traces.

Solaris OS snoop -V -vvv -d *interface* -o /tmp/gw-snoop-portal *ip-portal-server*

HP-UX tcpdump -i *interface* -w /tmp/gw-snoop-portal *ip-portal-server*

The tcpdump command is available here:

<http://hpux.connect.org.uk>

You can use the native nettl command too.

Linux tethereal -V -F snoop -i *interface* -w /tmp/gw-snoop-portal
ip-portal-server

Note – The tethereal command already should be installed. If not, get it from the following location: <http://www.ethereal.com>. You can also use the ethereal GUI or the tcpdump command.

Windows tethereal -vvv -i *interface* -w /tmp/gw-snoop-portal *host ip-portal-server*

Note – The tethereal command is available at the following location: <http://www.ethereal.com>. You can also use the ethereal GUI.

▼ To Collect Debug Data on a Portal Server Crashed Process

Use this task to collect data when a Portal Server process has stopped (crashed) unexpectedly. Run all the commands on the actual machine where the core file(s) were generated.

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

- 2 **Get the output of the following commands.**

Solaris OS `ps -ef | grep server-root`
 `vmstat 5 5`
 `iostat -x`
 `top`
 `uptime`

HP-UX `ps -aux | grep server-root`
 `vmstat 5 5`
 `iostat -x`
 `top`
 `sar`

Linux `ps -aux | grep server-root`
 `vmstat 5 5`
 `top`
 `uptime`
 `sar`

Windows Obtain the PROXY process PID: `C:\windbg-root>tlist.exe`

 Obtain process details of the PROXY running process PID:
 `C:\windbg-root>tlist.exe proxy-pid`

- 3 **Get the swap information.**

Solaris OS `swap -l`

HP-UX `swapinfo`

Linux `free`

Windows Already provided in `C:\report.txt` as described in “To Collect Required Debug Data for Any Portal Server Problem” on page 7.

- 4 **Get the system logs.**

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

5 Get the the Directory Server Access, Errors, and Audit logs used by Portal Server.

UNIX and Linux	<code>server-root/slapd-identifier/logs/access</code> <code>server-root/slapd-identifier/logs/errors</code> <code>server-root/slapd-identifier/logs/audit</code> (if enabled)
Windows	<code>server-root\slapd-identifier\logs\access</code> <code>server-root\slapd-identifier\logs\errors</code> <code>server-root\slapd-identifier\logs\audit</code> (if enabled)

6 Get the Access Manager configuration file.

UNIX and Linux	<code>/opt/SUNWam/lib/AMConfig.properties</code>
Windows	<code>access-manager-server-root\lib\AMConfig.properties</code>

7 Get the Access Manager log files.

UNIX and Linux	<code>/var/opt/SUNWam/*</code>
Windows	<code>access-manager-server-root\debug*</code>

8 Get core files (called “Crash Dumps” by Windows).

Solaris OS See “1.6 Configuring Solaris OS to Generate Core Files” on page 17 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 Portal Server by using the following commands:

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

The `adplus.vbs` command watches *portal-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.

9 (Solaris OS only) For each core file, provide the output of the following commands.

```
file corefile
pstack corefile
pmap corefile
pflags corefile
```

10 (Solaris OS only) Archive the result of the script `pkg_app` (one core file is sufficient).

```
./pkg_app.ksh Pid-of-application corefile
```

Note – The Sun Support Center must have the output from the `pkg_app` script to properly analyze the core file(s).

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

Before You Begin If you use the Web Server as the web container, make sure that the Web Server user is not nobody, as it may not provide the necessary core file in this case.

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 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 Portal Server Debugging Scripts

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

▼ To Run the `psinfo.sh` Script

- 1 Copy the script to a temporary directory on the system where Portal Server is installed.
- 2 Become superuser.
- 3 Make sure that you have executable permission on the script.
- 4 Run the script.
- 5 Collect the result.

▼ To Run the `iwshang` Script

- 1 The `iwshang` script collects three snapshots of the following information at 15 seconds interval against the hung instance:

```
pstack
pfiles
prstat -L -a
pflags
pmap -x
pldd
```

Note – You can modify the time interval by editing the script and changing the variable `DURATION`.

- 2 Run the `iwshang` script. It shows a list of Web Server process.
- 3 Choose the process that has the problem.

▼ 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 Portal 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 819-5489-10.

