Tivoli software

IBM

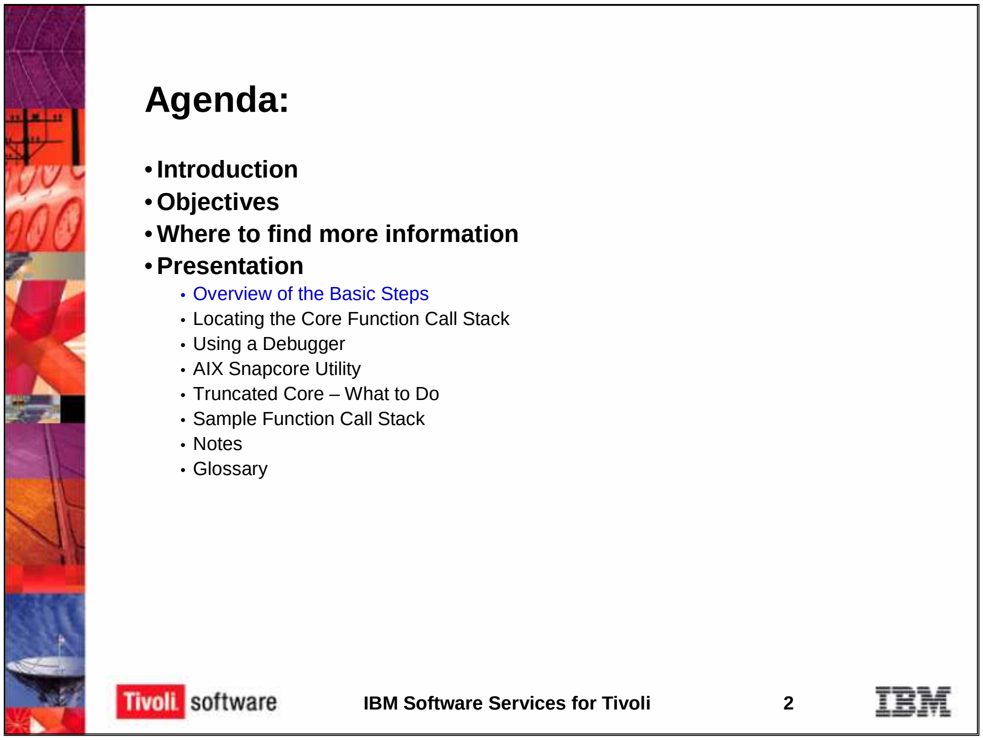
IBM Tivoli Support Technical Exchange Web Seminar:
Troubleshooting Tivoli Storage Manager Server Crashes

e business software

IBM Software Services for Tivoli

IBM Software Group

Welcome to the IBM Tivoli Support Technical Exchange Web Seminar on troubleshooting Tivoli Storage Manager server crashes.

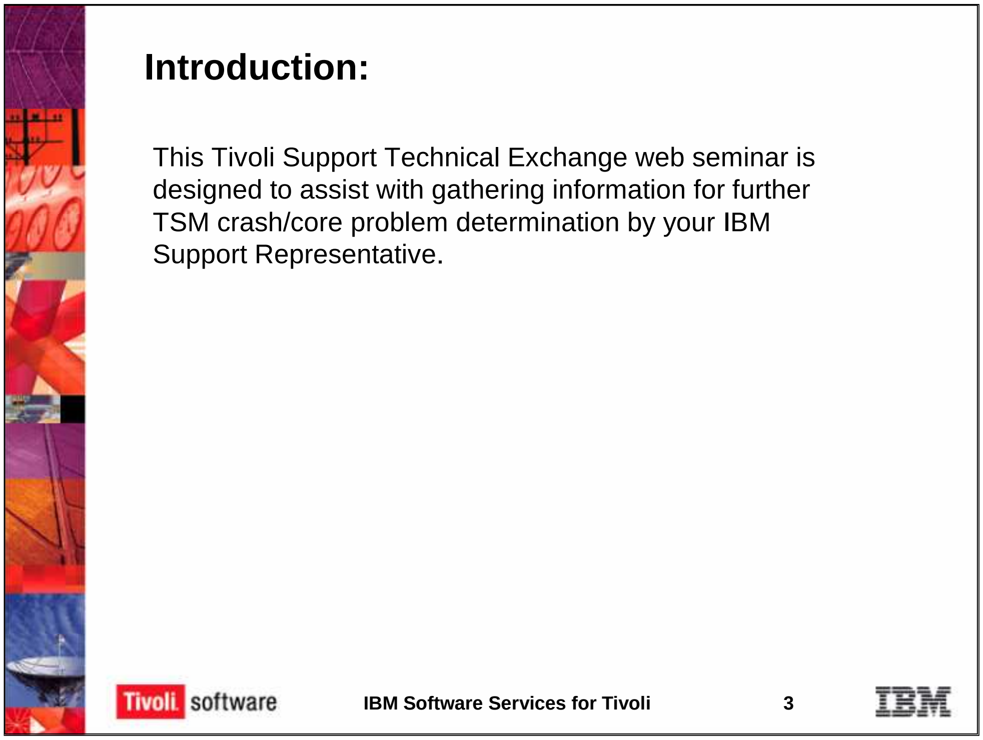


Agenda:

- **Introduction**
- **Objectives**
- **Where to find more information**
- **Presentation**
  - Overview of the Basic Steps
  - Locating the Core Function Call Stack
  - Using a Debugger
  - AIX Snapcore Utility
  - Truncated Core – What to Do
  - Sample Function Call Stack
  - Notes
  - Glossary

Tivoli software **IBM Software Services for Tivoli** **2** IBM

Our agenda will include an introduction, objectives, where to find more information and our basic presentation.

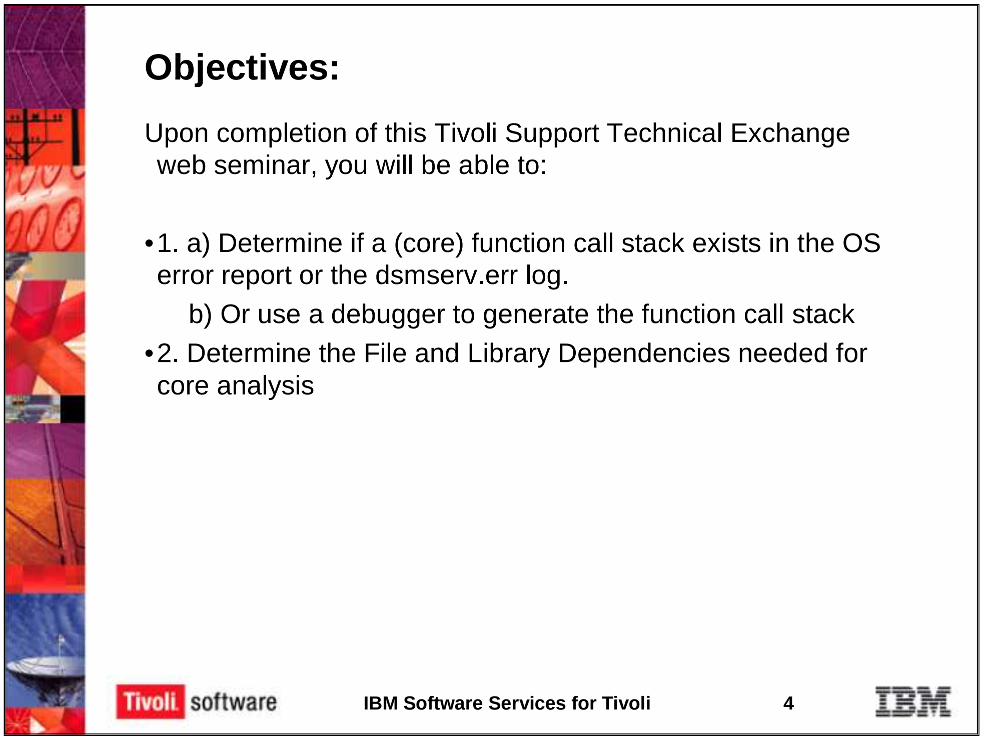


Introduction:

This Tivoli Support Technical Exchange web seminar is designed to assist with gathering information for further TSM crash/core problem determination by your IBM Support Representative.

IBM Software Services for Tivoli 3

This Tivoli Support Technical Exchange Web Seminar is designed to assist with gathering information for further problem determination by your IBM Support Representative regarding TSM crashes or when a TSM core is available.



Objectives:

Upon completion of this Tivoli Support Technical Exchange web seminar, you will be able to:

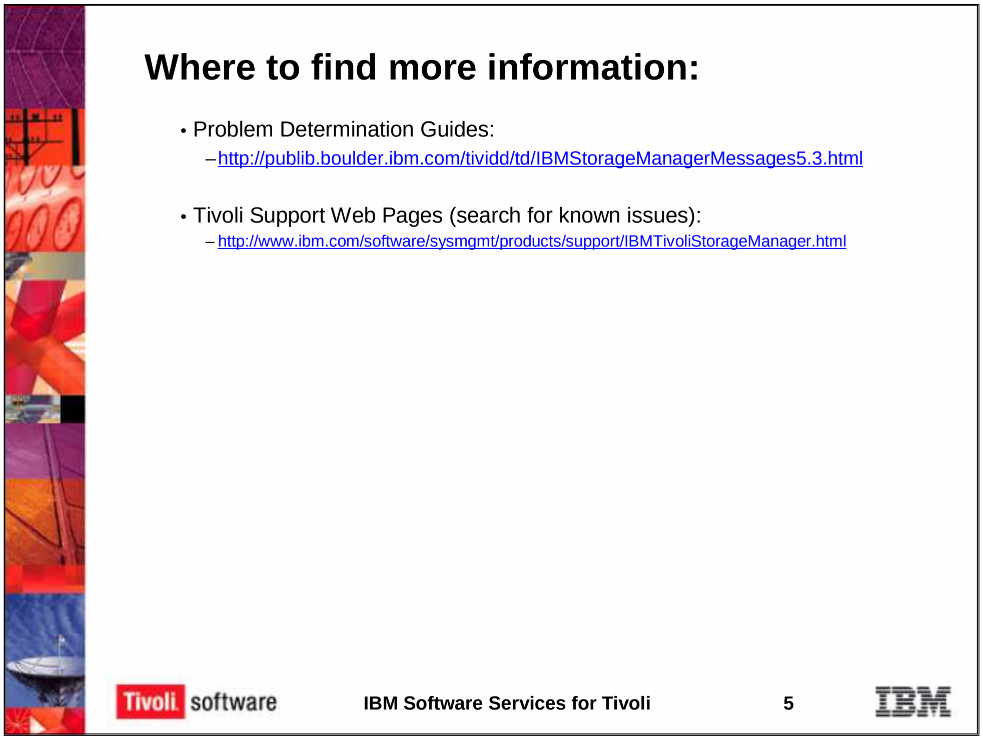
- 1. a) Determine if a (core) function call stack exists in the OS error report or the dsmserv.err log.

  b) Or use a debugger to generate the function call stack
- 2. Determine the File and Library Dependencies needed for core analysis

Tivoli. software IBM Software Services for Tivoli 4 IBM

There are two main objectives in this Tivoli Suport Technical Exchange. The first being to make a determination if a function call stack already exists in the OS error report and if not to determine if a call stack exists in the dsmserv.err log. If we don't have a stack in either of these two places then what we need to do is use a debugger to generate the function call stack. That's our first objective.

The second objective would be, if we need further core analysis then at this point we need to determine all the dependent files and libraries needed for core analysis. So they can be zipped and sent along with the core.

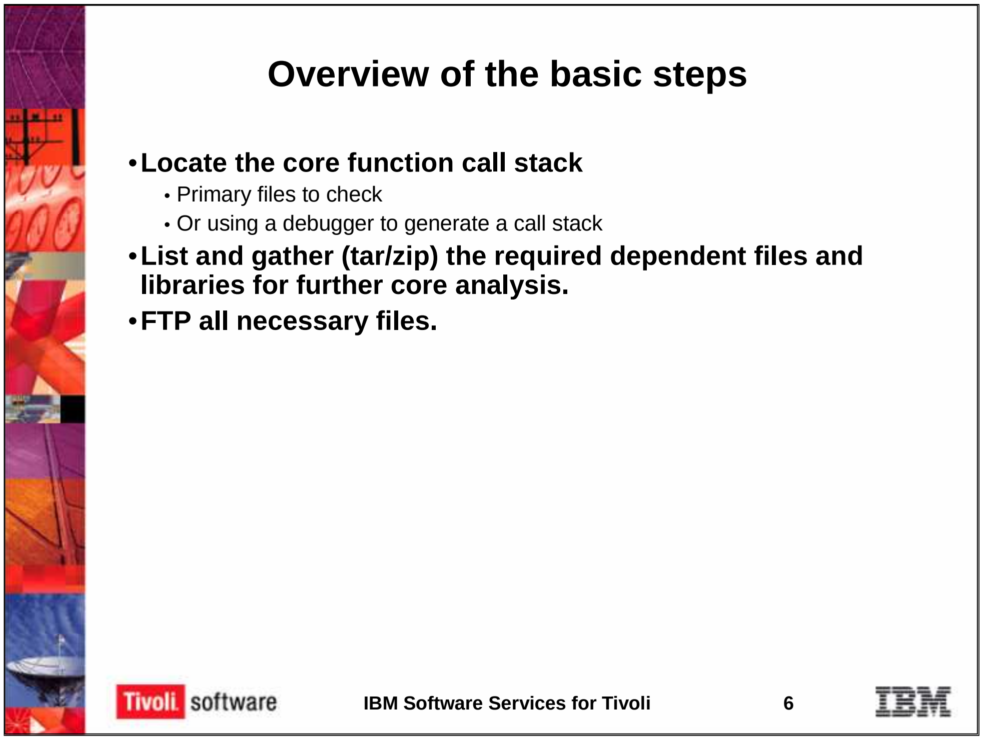


Where to find more information:

- Problem Determination Guides:
  - http://publib.boulder.ibm.com/tividd/td/IBMStorageManagerMessages5.3.html
- Tivoli Support Web Pages (search for known issues):
  - http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html

Tivoli software IBM Software Services for Tivoli 5 IBM

More information can be found at the two following links. The first one is for the Problem Determination Guide. The second one is the Tivoli Support Web pages.

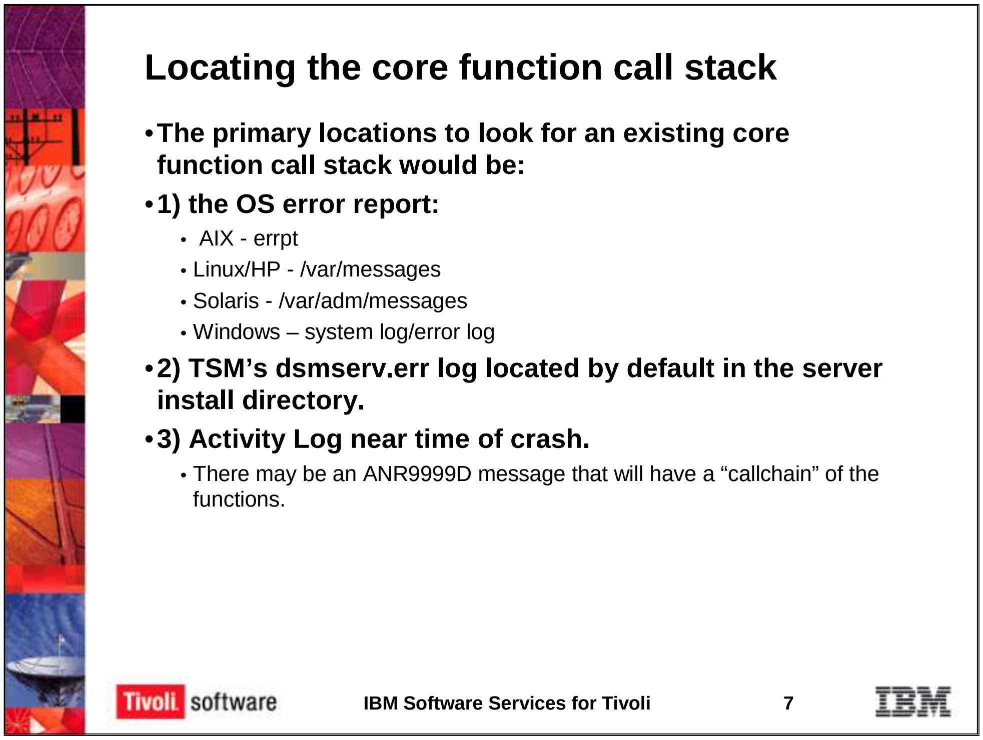


Overview of the basic steps

- **Locate the core function call stack**
  - Primary files to check
  - Or using a debugger to generate a call stack
- **List and gather (tar/zip) the required dependent files and libraries for further core analysis.**
- **FTP all necessary files.**

Tivoli software **IBM Software Services for Tivoli** **6**

A high level overview of the basic steps are to locate the function call stack. We can look at primary files - to check. If we don't find the function call stack in any of these files we can use a debugger to generate the call stack from the core. Once we have the call stack we can also using a debugger acquire the list of dependent files and libraries that would need to be gathered for further core analysis. All these files and the call stack, activity log and the core would need to be ftp'd to IBM.

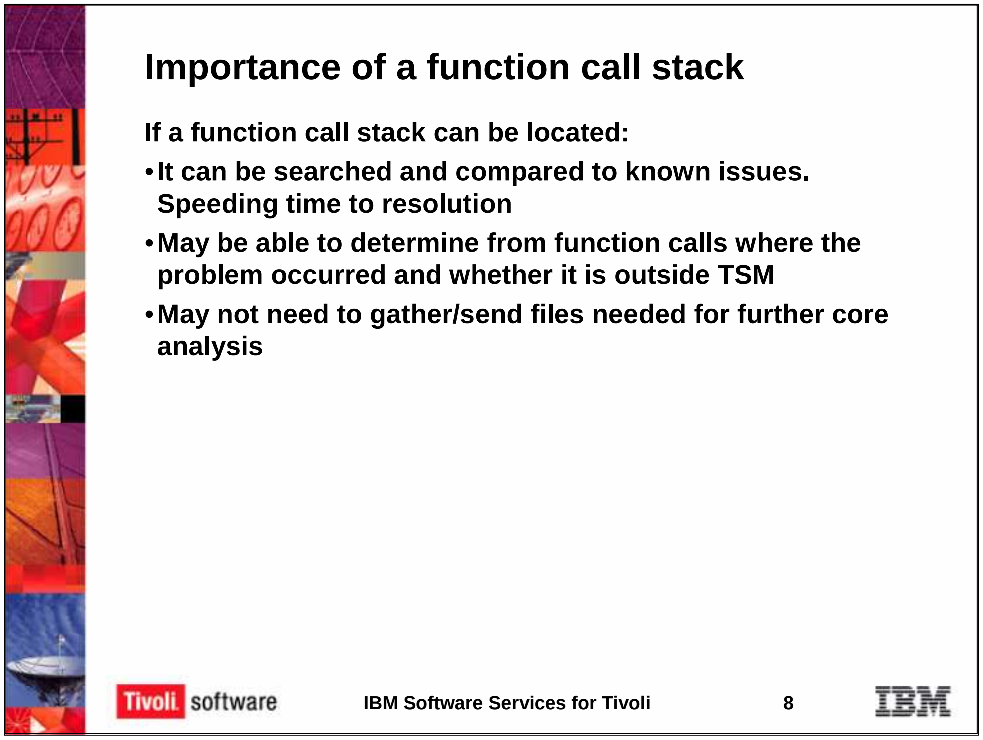


Locating the core function call stack

- **The primary locations to look for an existing core function call stack would be:**
- **1) the OS error report:**
  - AIX - errpt
  - Linux/HP - /var/messages
  - Solaris - /var/adm/messages
  - Windows – system log/error log
- **2) TSM's dsmserv.err log located by default in the server install directory.**
- **3) Activity Log near time of crash.**
  - There may be an ANR9999D message that will have a "callchain" of the functions.

Tivoli software IBM Software Services for Tivoli 7 IBM

There are three primary locations to determine if a core function call stack already exists. The first would be the OS error report such as the AIX errpt or the /var/messages file. The second place would be the dsmserv.err log. By default located in the TSM server installation directory. And the third place would be in the Activity log. Many times you will see an ANR9999D message that will have the call chain of the function call stack.



Importance of a function call stack

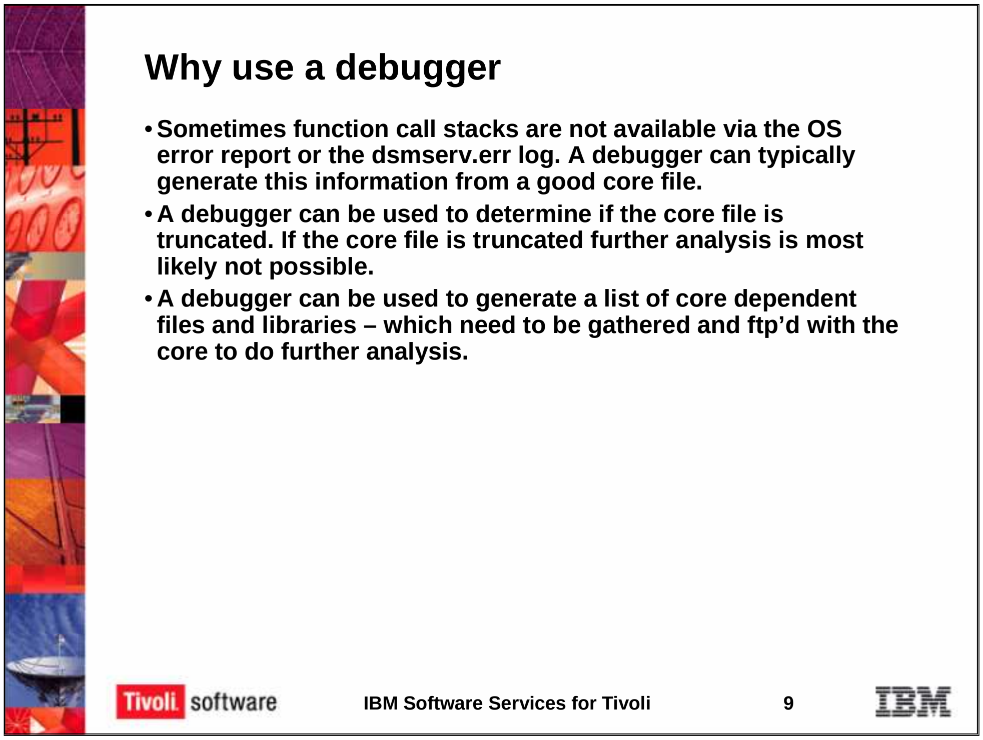
If a function call stack can be located:

- **It can be searched and compared to known issues. Speeding time to resolution**
- **May be able to determine from function calls where the problem occurred and whether it is outside TSM**
- **May not need to gather/send files needed for further core analysis**

Tivoli software IBM Software Services for Tivoli 8

It's important to get a function call stack from a core for various reasons. First with the function call stack we can research and make a determination whether it's a known issue, i.e. an apar, whether there is a local fix, whether an upgrade is needed without having to send in a core and the dependent files and libraries.

The second thing is that we can also determine from a function call stack where the problem is occurring. Many times if it's not a known issue we can determine the core blew up in a certain function and that function is inside TSM or that function is outside TSM.



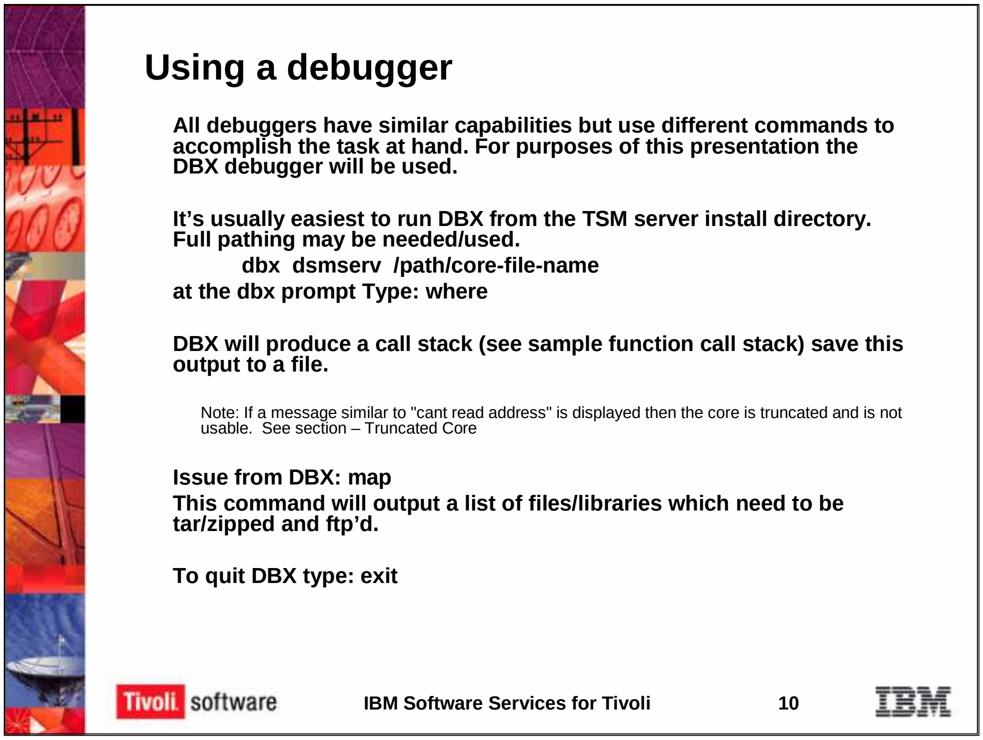
Why use a debugger

- Sometimes function call stacks are not available via the OS error report or the dsmserv.err log. A debugger can typically generate this information from a good core file.
- A debugger can be used to determine if the core file is truncated. If the core file is truncated further analysis is most likely not possible.
- A debugger can be used to generate a list of core dependent files and libraries – which need to be gathered and ftp'd with the core to do further analysis.

Tivoli software IBM Software Services for Tivoli 9

If a function call stack is not available in the primary files discussed earlier, a debugger is a very useful tool because it can be used to generate the function call stack from a core. It can also be determined whether the core file is truncated and whether further analysis can be performed on the core.

The debugger can also provide the list of dependent files and libraries that would be needed so that further core analysis can be performed - as long as the core is not truncated.



Using a debugger

All debuggers have similar capabilities but use different commands to accomplish the task at hand. For purposes of this presentation the DBX debugger will be used.

It’s usually easiest to run DBX from the TSM server install directory. Full pathing may be needed/used.

```
        dbx  dsmserv  /path/core-file-name
```

at the dbx prompt Type: where

DBX will produce a call stack (see sample function call stack) save this output to a file.

Note: If a message similar to "cant read address" is displayed then the core is truncated and is not usable. See section – Truncated Core

Issue from DBX: map

This command will output a list of files/libraries which need to be tar/zipped and ftp’d.

To quit DBX type: exit

Tivoli software   IBM Software Services for Tivoli   10   IBM

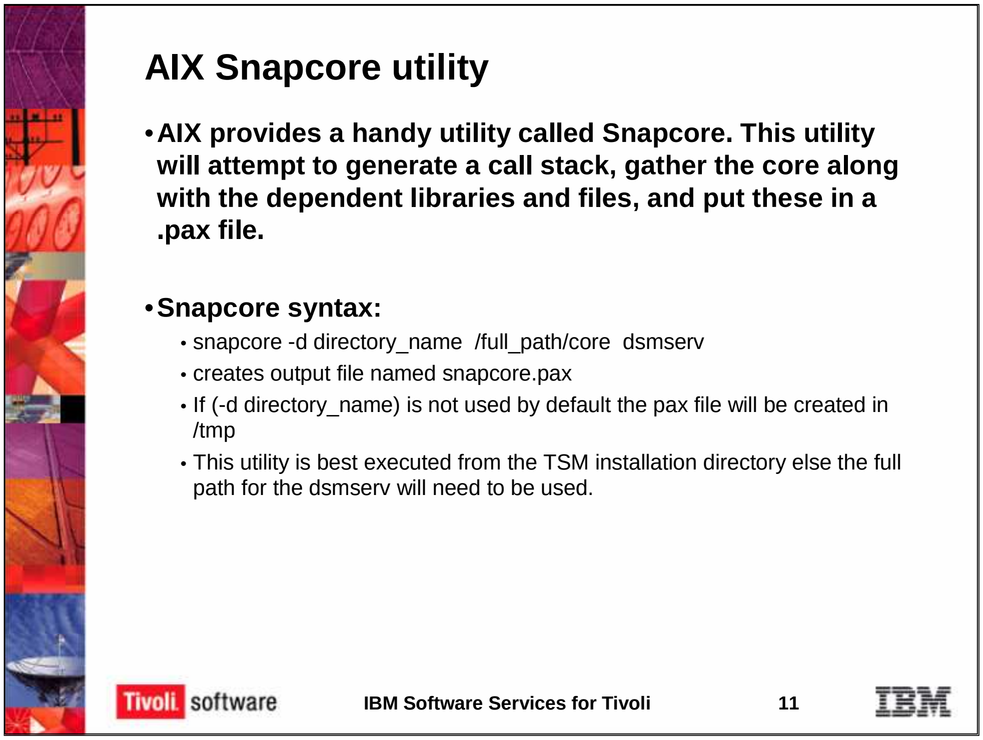
You may use a debugger of your choice. They all have similar capabilities but use different commands to accomplish the same task. For the purpose of this presentation we will use the DBX debugger to illustrate. From the TSM server installation directory you can see we simply issue the comand:

```
dbx dsmserv /path/core_file_name
```

You receive a DBX prompt and at that prompt you will type the DBX command "where", this will produce a call stack. You'll want to save this stack because this is what you're going to send in first.

From DBX you would then issues the "Map" command and this is going to provide a list of the dependent files and libraries which will need to be gathered - tar/zipped then ftp'd to IBM.

To quit DBX, you would simply type "exit".



AIX Snapcore utility

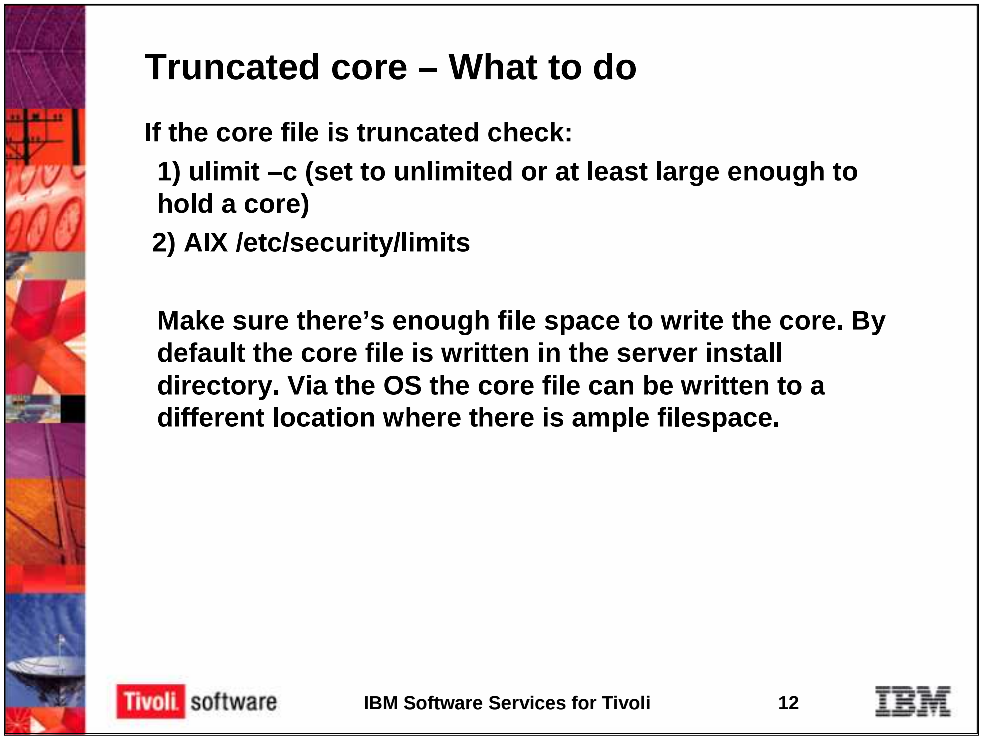
- AIX provides a handy utility called Snapcore. This utility will attempt to generate a call stack, gather the core along with the dependent libraries and files, and put these in a .pax file.

- **Snapcore syntax:**
  - snapcore -d directory_name /full_path/core dsmserv
  - creates output file named snapcore.pax
  - If (-d directory_name) is not used by default the pax file will be created in /tmp
  - This utility is best executed from the TSM installation directory else the full path for the dsmserv will need to be used.

Tivoli software IBM Software Services for Tivoli 11 IBM

AIX provides a handy little utility called Snapcore. This utility attempts to generate a call stack, gather the core along with all the dependent files and libraries and puts them in a .pax file.

You can see the snapcore syntax. It's best to issue snapcore from the TSM installation directory.



Truncated core – What to do

If the core file is truncated check:

1) ulimit –c (set to unlimited or at least large enough to hold a core)

2) AIX /etc/security/limits

Make sure there’s enough file space to write the core. By default the core file is written in the server install directory. Via the OS the core file can be written to a different location where there is ample filespace.

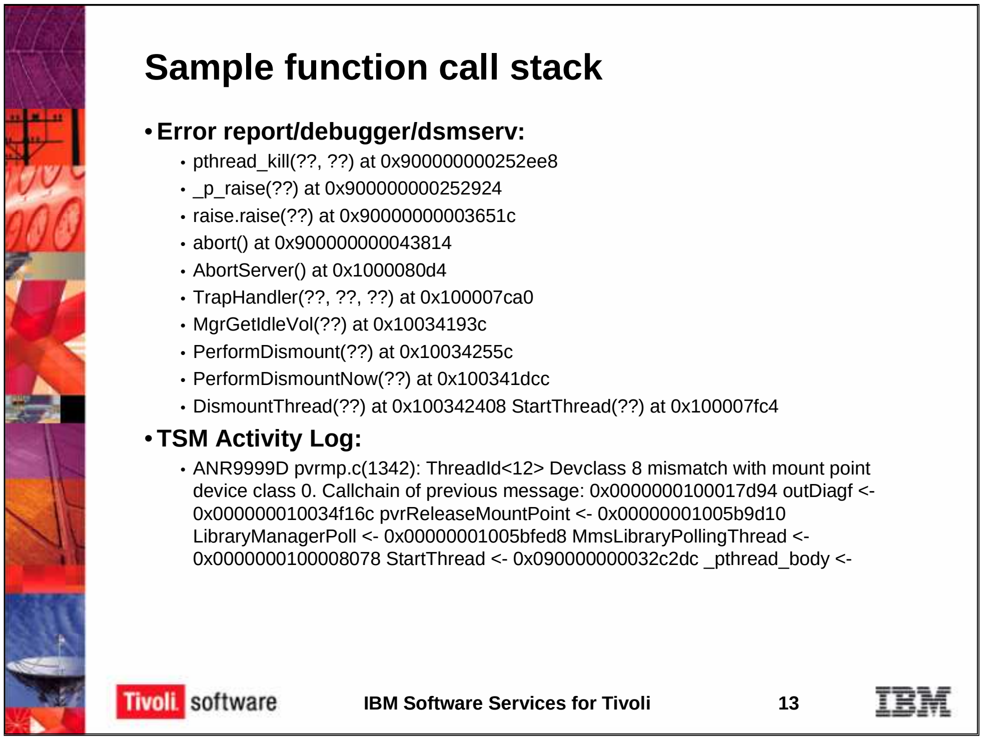
Tivoli software IBM Software Services for Tivoli 12 IBM

In our previous example of using the DBX debugger, when the "where" command is issued, it's at this point in time, if the core file is truncated, we'll receive a message of "cant read address".

So what do we do? First we need to check the Ulimit -C to make sure it is set to unlimited. The second thing is to check the AIX /etc/security/limits.

The third thing is to make sure there is ample file space to where the core file is being written. If the core file requires 1.5GB of space and only .5GB

of space is available, it'll simply run out of space trying to write the core file. Thus the file is truncated.

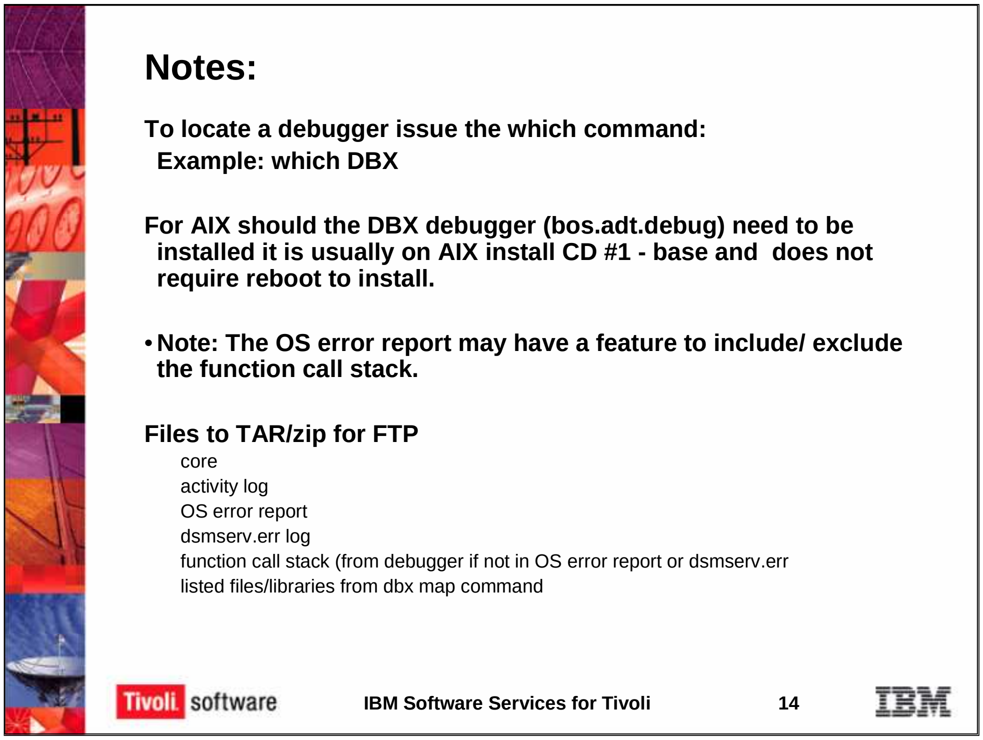


Sample function call stack

- **Error report/debugger/dsmserv:**
  - pthread_kill(??, ??) at 0x900000000252ee8
  - _p_raise(??) at 0x900000000252924
  - raise.raise(??) at 0x90000000003651c
  - abort() at 0x900000000043814
  - AbortServer() at 0x1000080d4
  - TrapHandler(??, ??, ??) at 0x100007ca0
  - MgrGetIdleVol(??) at 0x10034193c
  - PerformDismount(??) at 0x10034255c
  - PerformDismountNow(??) at 0x100341dcc
  - DismountThread(??) at 0x100342408 StartThread(??) at 0x100007fc4
- **TSM Activity Log:**
  - ANR9999D pvrmp.c(1342): ThreadId<12> Devclass 8 mismatch with mount point device class 0. Callchain of previous message: 0x0000000100017d94 outDiagf <- 0x000000010034f16c pvrReleaseMountPoint <- 0x00000001005b9d10 LibraryManagerPoll <- 0x00000001005bfed8 MmsLibraryPollingThread <- 0x0000000100008078 StartThread <- 0x090000000032c2dc _pthread_body <-

IBM Software Services for Tivoli 13

We have some sample function call stacks. The first would be what you would see typically in the AIX errpt or the dsmserv.err log. At the bottom you'll see the ANR9999D message and you can see each of the functions listed.



Notes:

To locate a debugger issue the which command:
Example: which DBX

For AIX should the DBX debugger (bos.adt.debug) need to be installed it is usually on AIX install CD #1 - base and does not require reboot to install.

- **Note: The OS error report may have a feature to include/ exclude the function call stack.**

Files to TAR/zip for FTP

core
activity log
OS error report
dsmserv.err log
function call stack (from debugger if not in OS error report or dsmserv.err
listed files/libraries from dbx map command

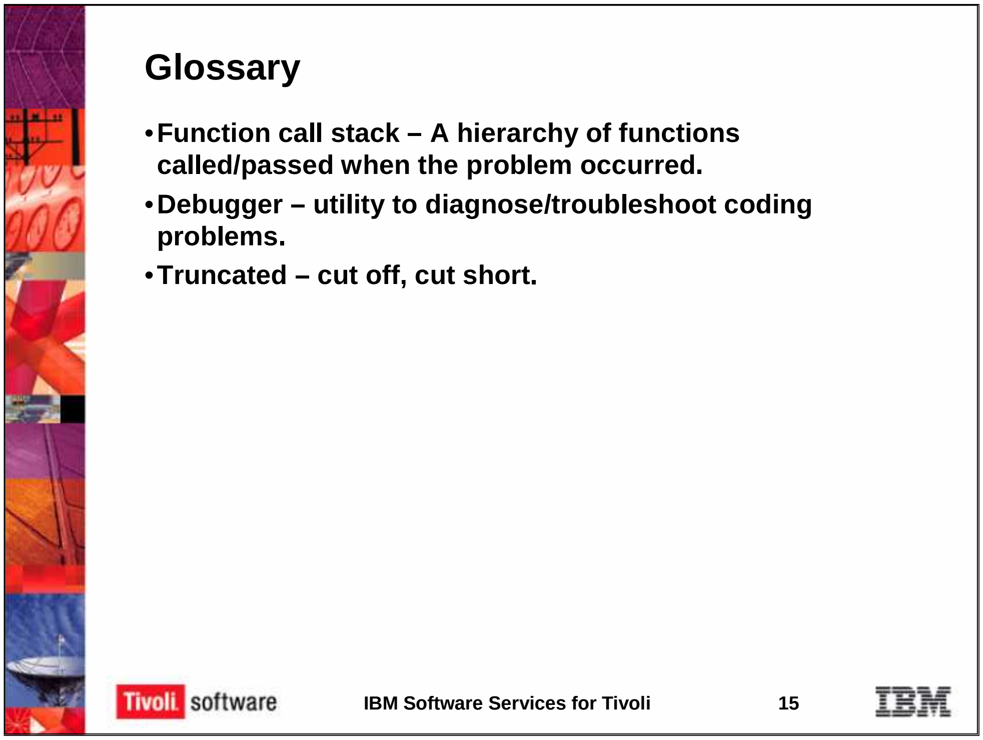
Tivoli software IBM Software Services for Tivoli 14 IBM

We have some notes. To locate a debugger you would simply issue "which DBX" or "which GBX" this will let you know if it's installed on the system.

In the case of AIX if DBX is not installed it can be installed from the base CD, the bos.adt.debug fileset. It does not require reboot for installation.

Another note is the OS error report may have the ability to enable or disable the function call stack.

Then there's the list of the files to be tar/zipped should we need to do core analysis.



Glossary

- Function call stack – A hierarchy of functions called/passed when the problem occurred.
- Debugger – utility to diagnose/troubleshoot coding problems.
- Truncated – cut off, cut short.

Tivoli software IBM Software Services for Tivoli 15

As the final slide of our presentation we have a glossary. Each of these terms are explained. Thank you.