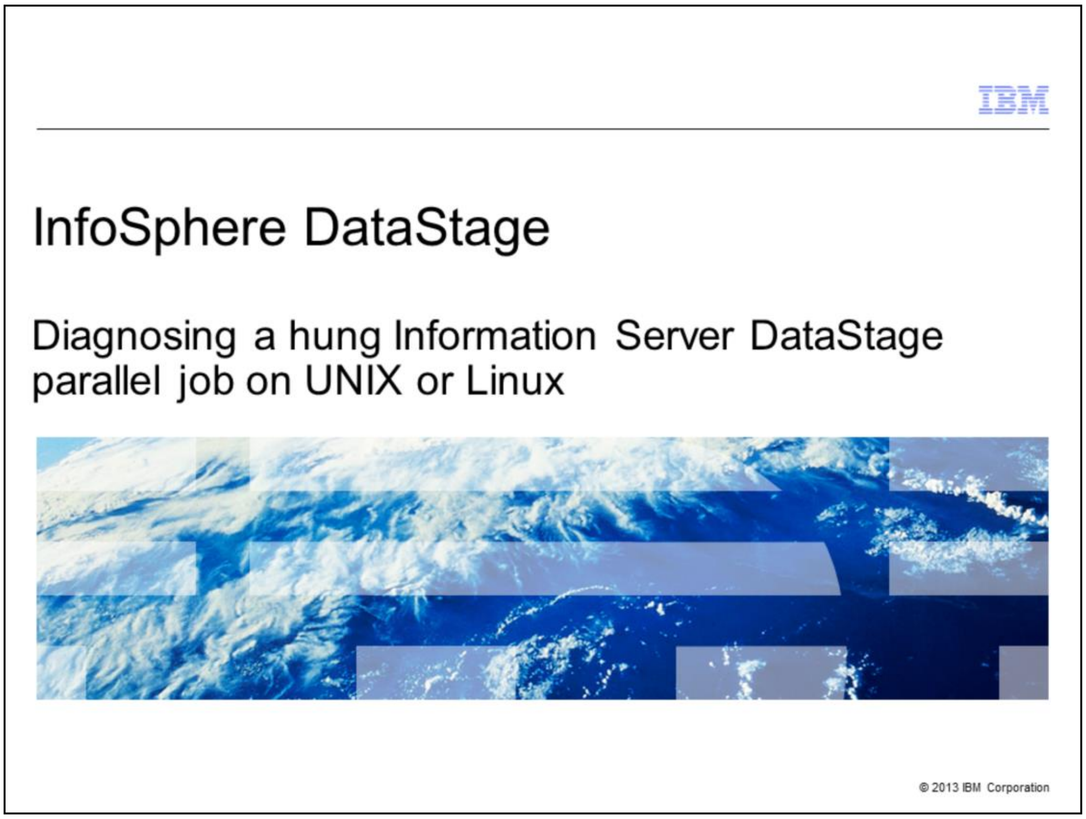
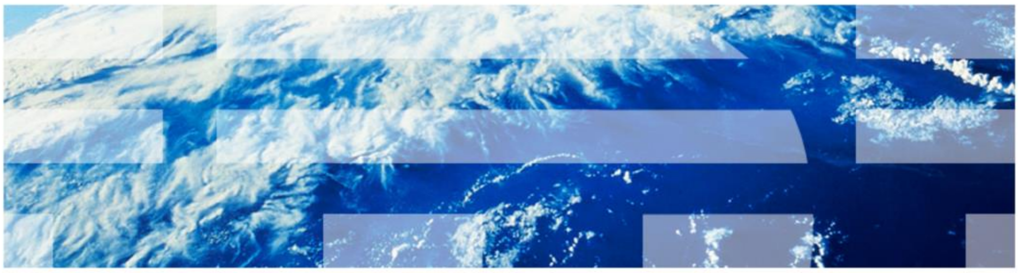
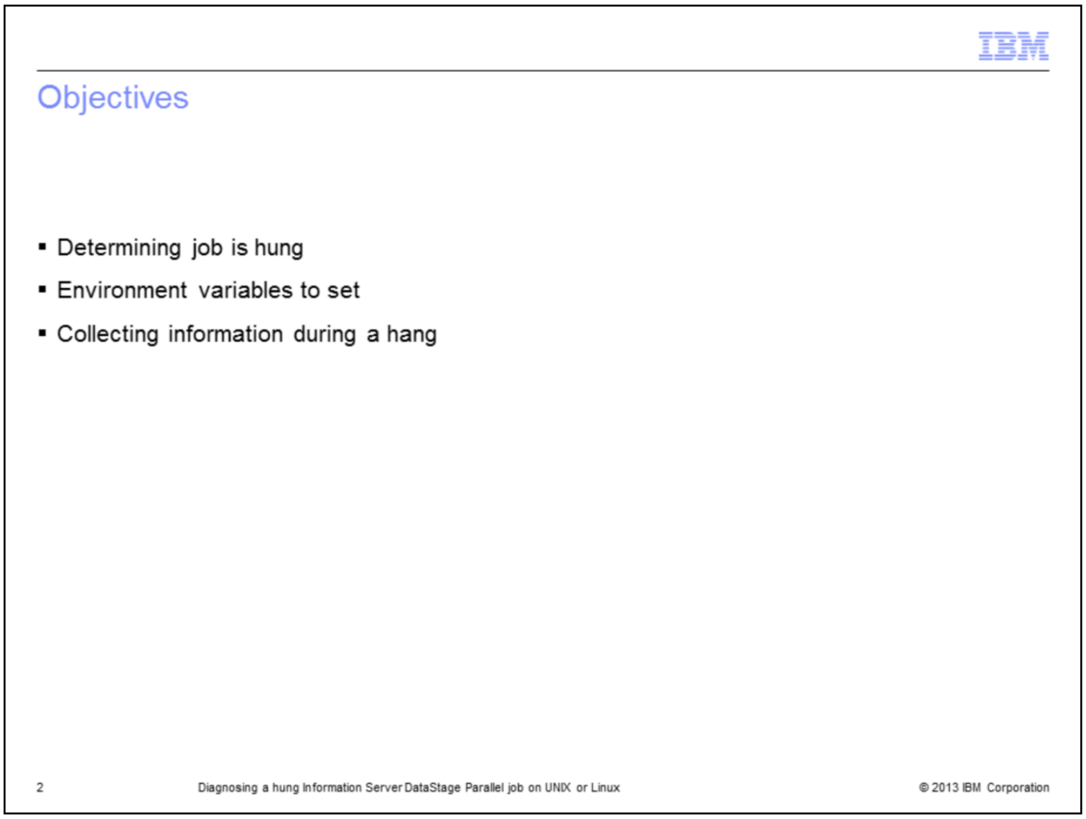


InfoSphere DataStage

Diagnosing a hung Information Server DataStage parallel job on UNIX or Linux



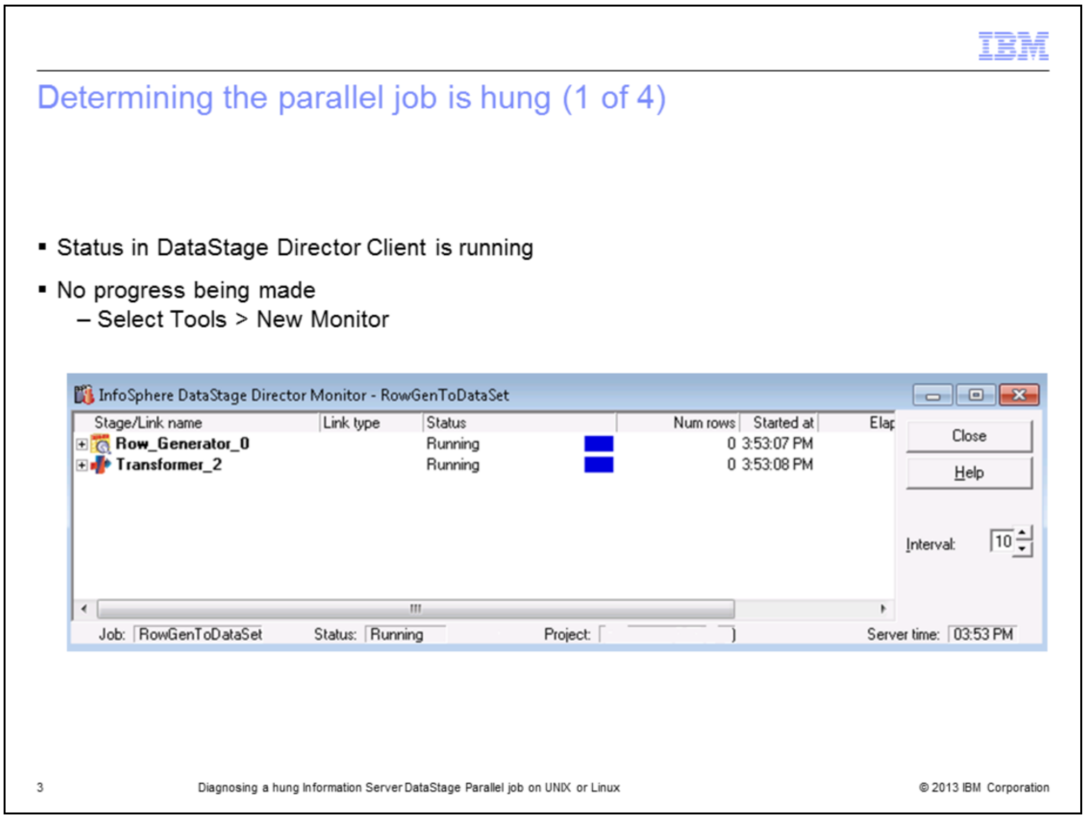
This presentation discusses how to diagnose hung Information Server DataStage® parallel jobs. This presentation is relevant for Information Server version 8.1 through version 11.3.



Objectives

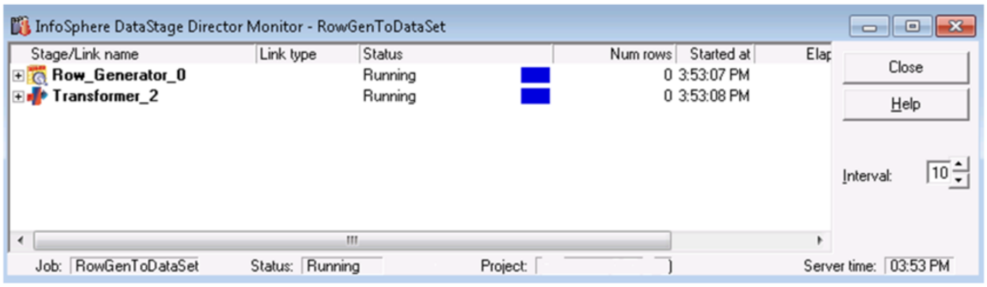
- Determining job is hung
- Environment variables to set
- Collecting information during a hang

The objectives of this presentation are to show how to determine whether a parallel job is hung, and if so, it describes the environment variables that need to be set and what information to collect during the hang.



Determining the parallel job is hung (1 of 4)

- Status in DataStage Director Client is running
- No progress being made
  – Select Tools > New Monitor



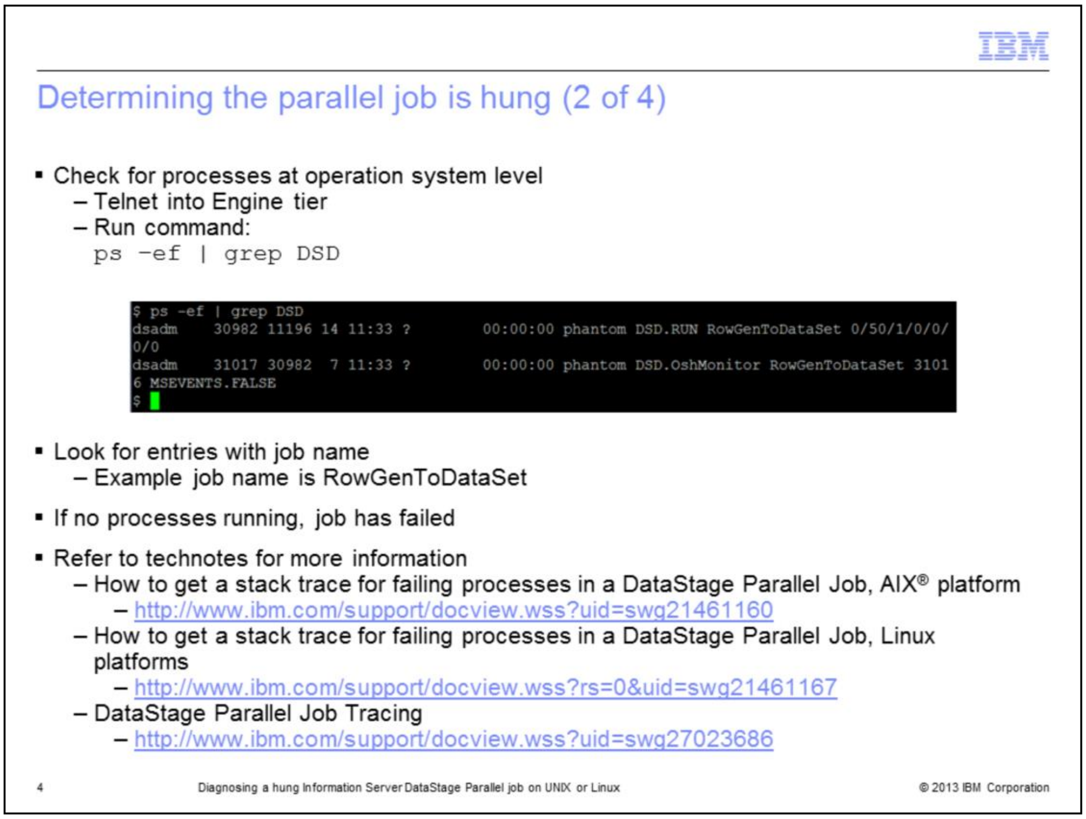
InfoSphere DataStage Director Monitor - RowGenToDataSet

Stage/Link name	Link type	Status		Num rows	Started at	Elap
⊞ Row_Generator_0		Running	■	0	3:53:07 PM	
⊞ Transformer_2		Running	■	0	3:53:08 PM	

Interval: 10

Job: RowGenToDataSet Status: Running Project: Server time: 03:53 PM

If a job is hung, the DataStage Director shows that the job is running but the job monitor will not show any progress. First, open the DataStage Director, click the Tools menu, and then click New Monitor. If rows are still being processed, even if it is very slow, the job is not hung.



IBM

Determining the parallel job is hung (2 of 4)

- Check for processes at operation system level
  - Telnet into Engine tier
  - Run command:
    `ps -ef | grep DSD`

```
$ ps -ef | grep DSD
dsadm    30982 11196 14 11:33 ?        00:00:00 phantom DSD.RUN RowGenToDataSet 0/50/1/0/0/
0/0
dsadm    31017 30982  7 11:33 ?        00:00:00 phantom DSD.OshMonitor RowGenToDataSet 3101
6 MSEVENTS.FALSE
$
```

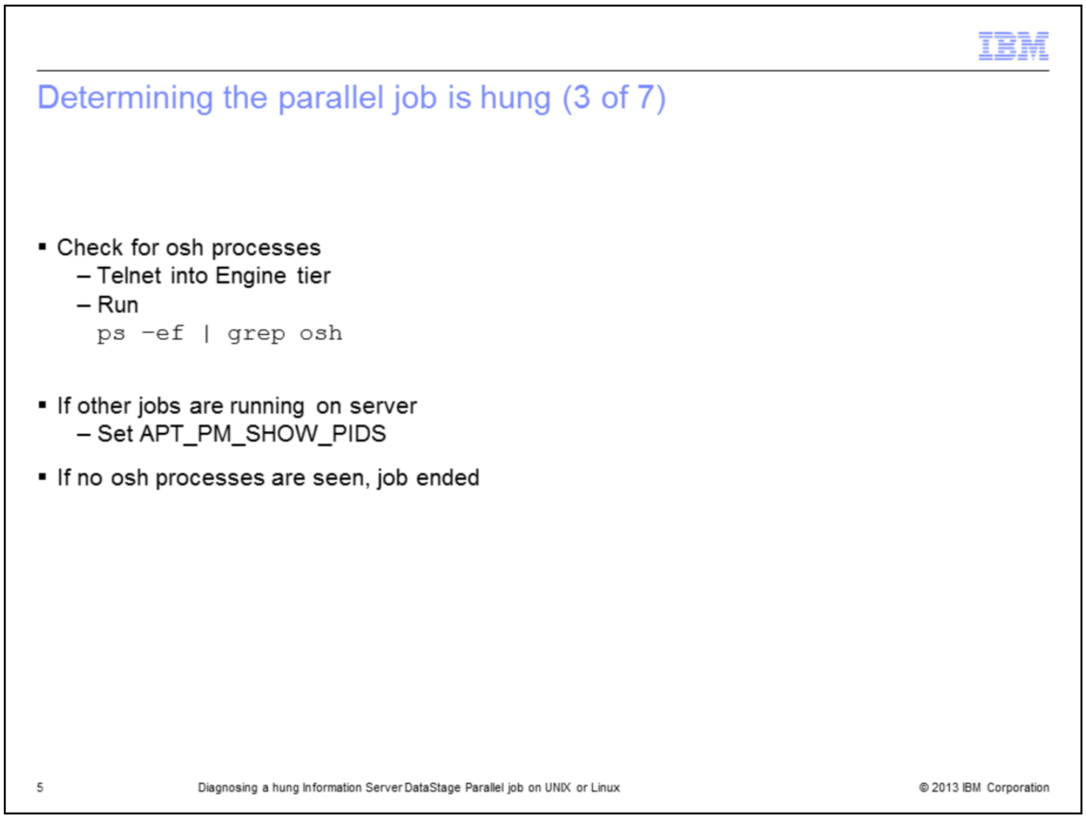
- Look for entries with job name
  - Example job name is RowGenToDataSet
- If no processes running, job has failed
- Refer to technotes for more information
  - How to get a stack trace for failing processes in a DataStage Parallel Job, AIX® platform
    - http://www.ibm.com/support/docview.wss?uid=swg21461160
  - How to get a stack trace for failing processes in a DataStage Parallel Job, Linux platforms
    - http://www.ibm.com/support/docview.wss?rs=0&uid=swg21461167
  - DataStage Parallel Job Tracing
    - http://www.ibm.com/support/docview.wss?uid=swg27023686

4 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

If the job appears to be hung, the next step is to check for job processes at the operating system level. Telnet into the Engine tier and run the ps command that is displayed on this slide.

The DSD.RUN process is the first process kicked off and starts the other related processes. The DSD.OshMonitor collects information on the row counts.

If no processes are seen, then most likely the job failed, but was not able to update the status of the job before terminating. This is not considered a hung job. Clear the status file from DataStage Director. Next, look for a core file in the project directory with a timestamp that matches the last entry in the job log. If no core file is found, confirm that the operating system is configured to generate core files. The first two technotes that are listed on this slide provide examples of how to get a stack trace for AIX and Linux. If the job failure can be reproduced, see the third link that is listed on the slide, which describes how to trace the job.



Determining the parallel job is hung (3 of 7)

- Check for osh processes
  - Telnet into Engine tier
  - Run

```
ps -ef | grep osh
```

- If other jobs are running on server
  - Set APT_PM_SHOW_PIDS
- If no osh processes are seen, job ended

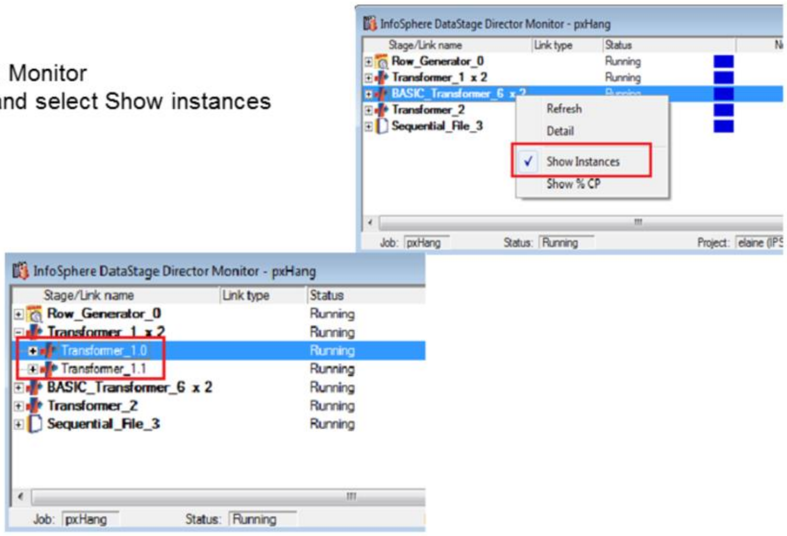
5 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

After locating the DSD.RUN process, the next step is to check for osh processes at the operating system level using the ps command that is displayed on this slide. If other jobs are running on the server, it is difficult to distinguish the osh processes associated with the hung job. The next section describes how to set the environment variable APT_PM_SHOW_PIDS. Setting this environment variable causes the pids to be written to the job log. These pids can then be used to find the osh processes at the OS level.

If no osh processes are returned from the ps command, then the job ended but was not able to update the status before terminating. This is not considered a hung job. Clear the status file from DataStage Director. Next, look for a core file in the Project directory with a timestamp that matches the last entry in the job log and see the technotes on the previous slide for examples on how to get a stack trace.

Determining the parallel job is hung (4 of 7)

- Find PIDs using Monitor
  - Right click and select Show instances



InfoSphere DataStage Director Monitor - pxHang

Stage/Link name	Link type	Status
Row_Generator_0		Running
Transformer_1 x 2		Running
BASIC_Transformer_6 x 2		Running
Transformer_2		
Sequential_File_3		

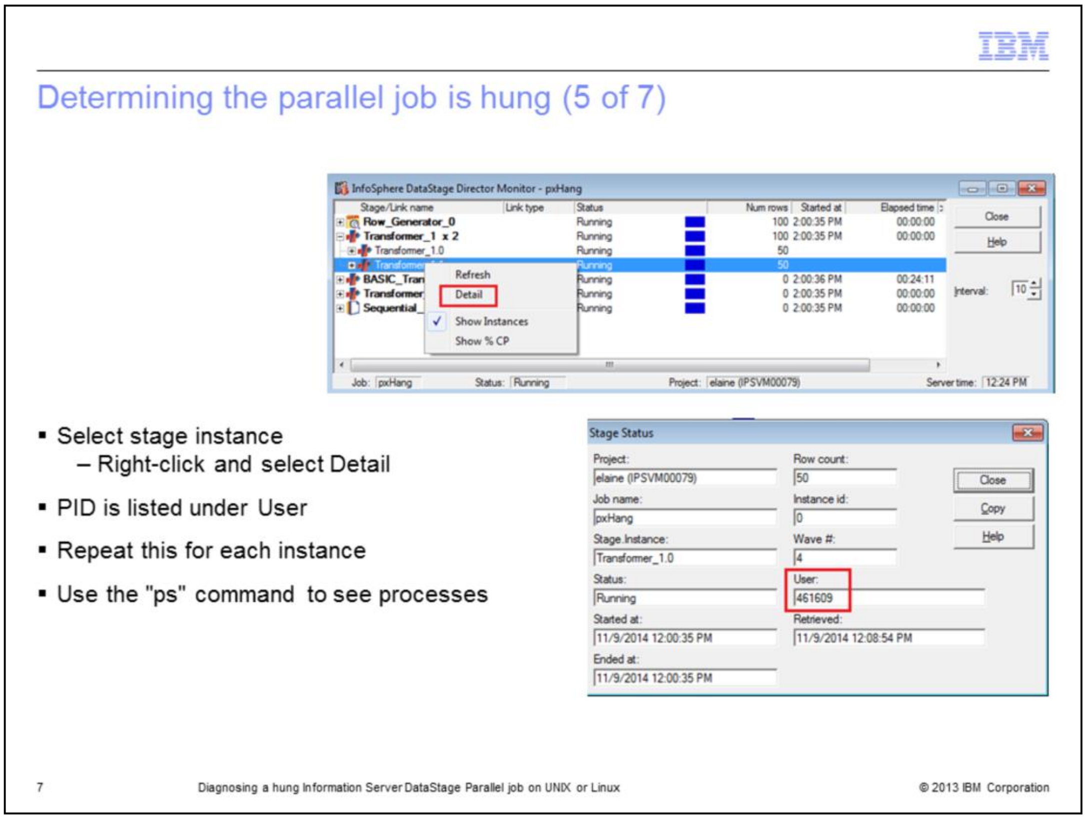
Job: pxHang Status: Running Project: elaine (IPS

InfoSphere DataStage Director Monitor - pxHang

Stage/Link name	Link type	Status
Row_Generator_0		Running
Transformer_1 x 2		Running
Transformer_1.0		Running
Transformer_1.1		Running
BASIC_Transformer_6 x 2		Running
Transformer_2		Running
Sequential_File_3		Running

Job: pxHang Status: Running

Though difficult, you can find the PIDs using the monitor window. First, right-click and select show instances if not already checked. If running with a multi-node configuration file you see each of the instances. The example that is displayed on this slide was run with two nodes.



Determining the parallel job is hung (5 of 7)

InfoSphere DataStage Director Monitor - pxHang

Stage/Link name	Link type	Status		Num rows	Started at	Elapsed time
Row_Generator_0		Running	🟦	100	2:00:35 PM	00:00:00
Transformer_1 x 2		Running	🟦	100	2:00:35 PM	00:00:00
Transformer_1.0		Running	🟦	50		
Transformer		Running	🟦	50		
BASIC_Tran		Running	🟦	0	2:00:36 PM	00:24:11
Transformer		Running	🟦	0	2:00:35 PM	00:00:00
Sequential_		Running	🟦	0	2:00:35 PM	00:00:00

Job: pxHang   Status: Running   Project: elaine (IPSVM00079)   Server time: 12:24 PM

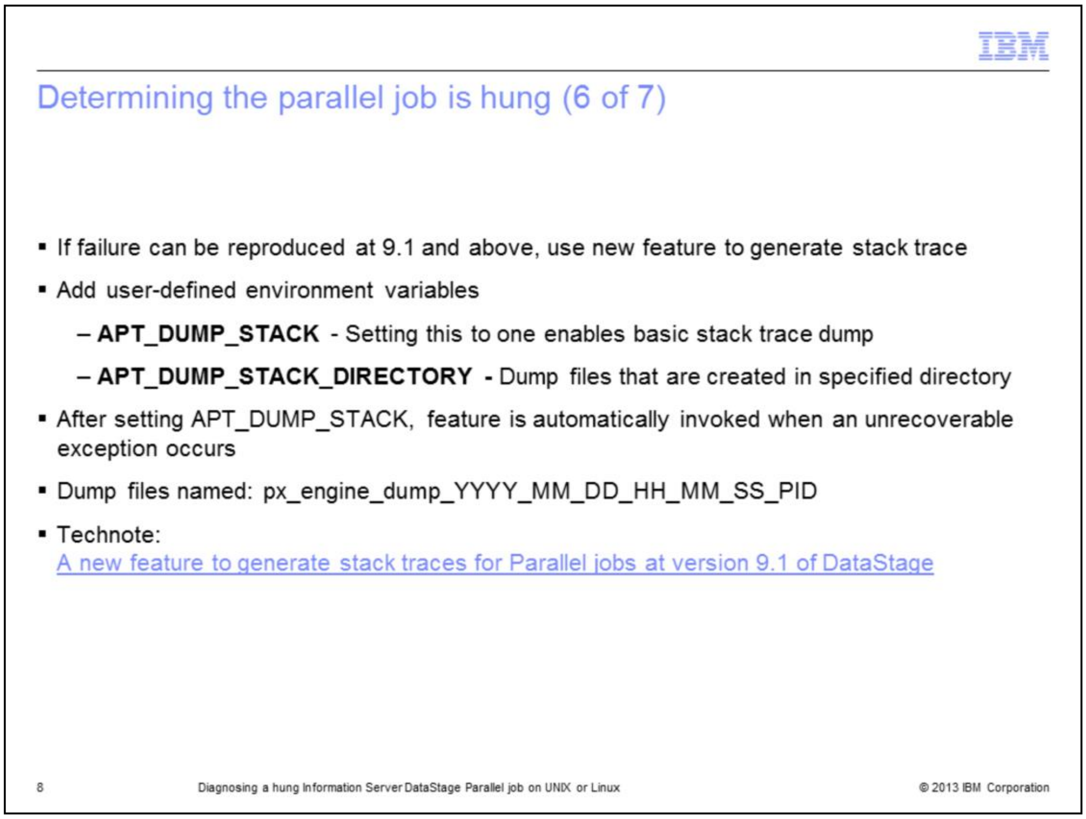
Interval: 10

- Select stage instance
  – Right-click and select Detail
- PID is listed under User
- Repeat this for each instance
- Use the "ps" command to see processes

Stage Status

Project: elaine (IPSVM00079)	Row count: 50
Job name: pxHang	Instance id: 0
Stage.Instance: Transformer_1.0	Wave #: 4
Status: Running	User: 461609
Started at: 11/9/2014 12:00:35 PM	Retrieved: 11/9/2014 12:08:54 PM
Ended at: 11/9/2014 12:00:35 PM	

Next, to find the PID, select the stage instance, right-click and select Detail. In the Stage Status dialog box, the PID is listed under User. Repeat this for each stage and instance. You can then use the “ps” command on UNIX® and Linux® to see the processes.

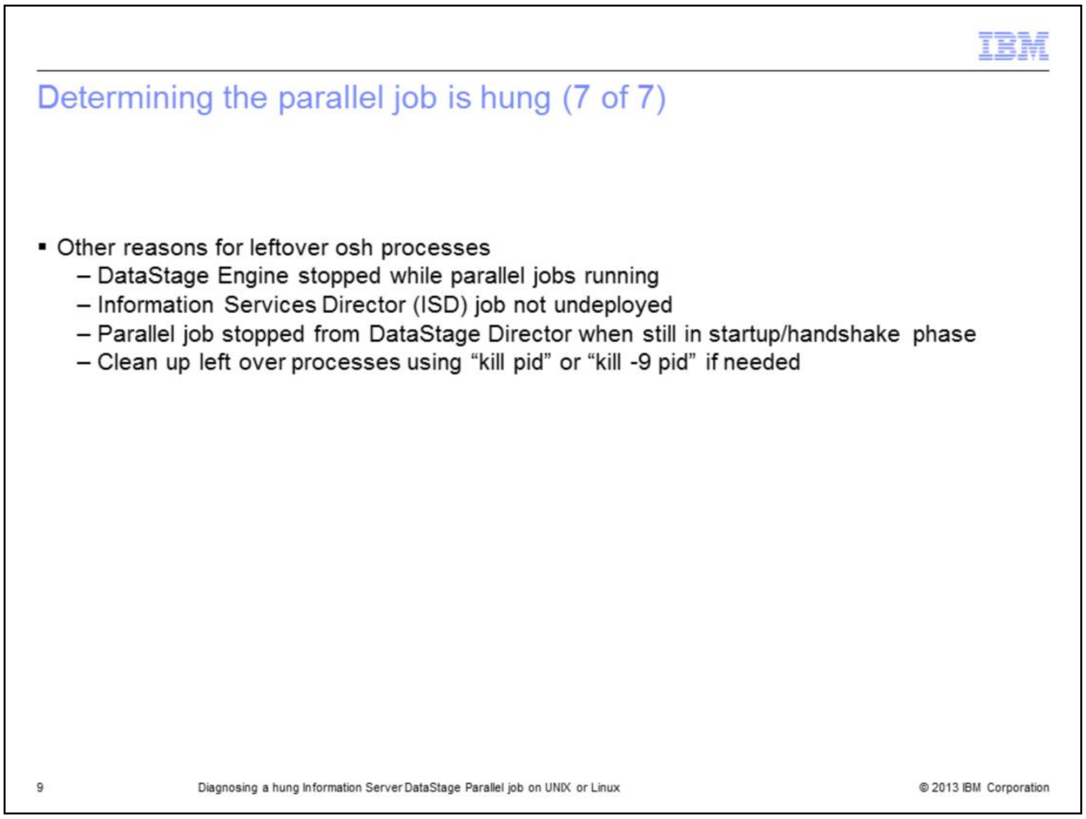


Determining the parallel job is hung (6 of 7)

- If failure can be reproduced at 9.1 and above, use new feature to generate stack trace
- Add user-defined environment variables
  - **APT_DUMP_STACK** - Setting this to one enables basic stack trace dump
  - **APT_DUMP_STACK_DIRECTORY -** Dump files that are created in specified directory
- After setting APT_DUMP_STACK, feature is automatically invoked when an unrecoverable exception occurs
- Dump files named: px_engine_dump_YYYY_MM_DD_HH_MM_SS_PID
- Technote:
  A new feature to generate stack traces for Parallel jobs at version 9.1 of DataStage

8 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

If the failure can be reproduced, starting at version 9.1 of DataStage, there is a new facility to generate stack traces and capture other valuable information for parallel jobs. The feature can be invoked by adding the user-defined environment variables APT_DUMP_STACK and APT_DUMP_STACK_DIRECTORY. Set APT_DUMP_STACK to one to enable basic stack trace dump. Set APT_DUMP_STACK_DIRECTORY to a valid path where files will be written. If undefined or not set to a valid path then the dump files will default to /tmp. If the job is successful, a dump is not created, therefore, you can leave this set to capture a dump for an intermittent issue.



Determining the parallel job is hung (7 of 7)

- Other reasons for leftover osh processes
  - DataStage Engine stopped while parallel jobs running
  - Information Services Director (ISD) job not undeployed
  - Parallel job stopped from DataStage Director when still in startup/handshake phase
  - Clean up left over processes using “kill pid” or “kill -9 pid” if needed

9 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

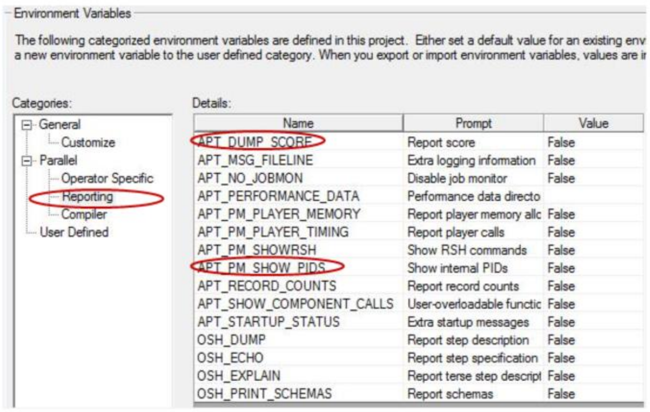
There are other reasons that leftover osh processes might be seen even though the job is not hung. One reason is when the DataStage Engine is stopped while parallel jobs are running or an Information Service Director™ job, at earlier releases referred to as WISD or RTI, is not undeployed before stopping the DataStage Engine. To prevent this, always check for running jobs and undeploy any Information Service Director jobs before stopping the DataStage engine.

Another reason is when a parallel job is stopped from the DataStage Director during the startup phase. This is the phase where the conductor communicates with section leaders, the section leader communicates with players, or players communicate with players.

In both of these cases, the leftover processes can be cleaned up.

Environment variables to set (1 of 3)

- Set environment variables at job or project level
  - APT_PM_SHOW_PIDS=True
  - APT_DUMP_SCORE=True
- Variables listed under Parallel > Reporting section



Environment Variables

The following categorized environment variables are defined in this project. Either set a default value for an existing env
a new environment variable to the user defined category. When you export or import environment variables, values are i

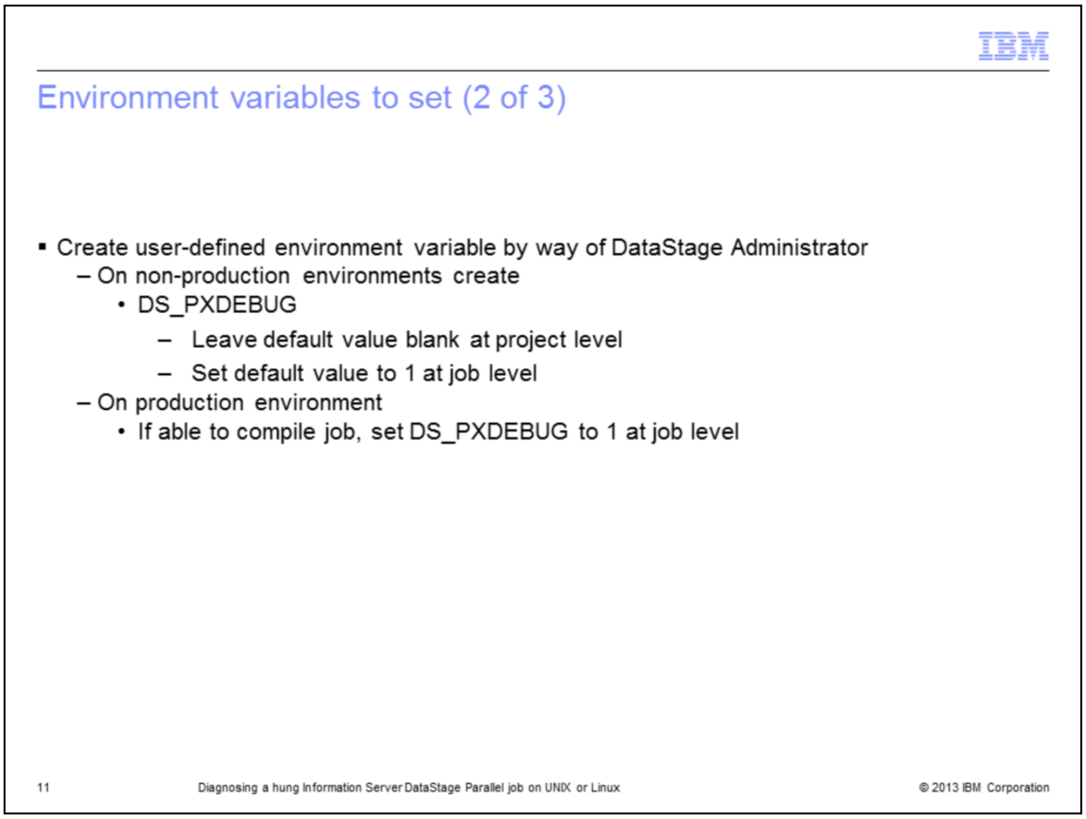
Categories:

- General
  - Customize
- Parallel
  - Operator Specific
  - Reporting
  - Compiler
- User Defined

Details:

Name	Prompt	Value
APT_DUMP_SCORE	Report score	False
APT_MSG_FILELINE	Extra logging information	False
APT_NO_JOBMON	Disable job monitor	False
APT_PERFORMANCE_DATA	Performance data directo	
APT_PM_PLAYER_MEMORY	Report player memory allc	False
APT_PM_PLAYER_TIMING	Report player calls	False
APT_PM_SHOWRSH	Show RSH commands	False
APT_PM_SHOW_PIDS	Show internal PIDs	False
APT_RECORD_COUNTS	Report record counts	False
APT_SHOW_COMPONENT_CALLS	User-overloadable functic	False
APT_STARTUP_STATUS	Extra startup messages	False
OSH_DUMP	Report step description	False
OSH_ECHO	Report step specification	False
OSH_EXPLAIN	Report terse step descript	False
OSH_PRINT_SCHEMAS	Report schemas	False

Once it is determined that the job is hung, setting the environment variables that are listed on this slide and on the next couple of slides, provides the information needed the next time that the job hangs.

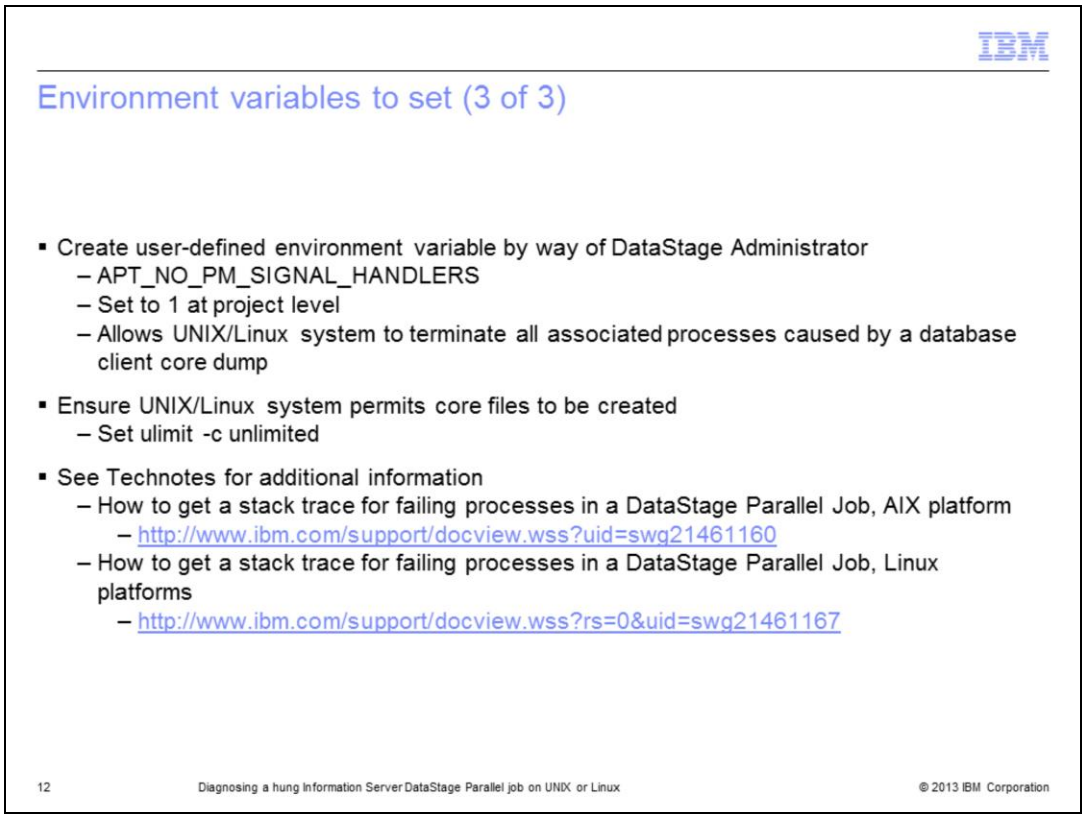


Environment variables to set (2 of 3)

- Create user-defined environment variable by way of DataStage Administrator
  - On non-production environments create
    - DS_PXDEBUG
      - Leave default value blank at project level
      - Set default value to 1 at job level
  - On production environment
    - If able to compile job, set DS_PXDEBUG to 1 at job level

11 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

Next, create a user-defined environment variable that is called DS_PXDEBUG if the job is not running in a production environment, and set the default value to 1 at the job level. If it is a production environment and there is the ability to compile a job, set DS_PXDEBUG to 1 at the job level as well. Avoid setting DS_PXDEBUG at the project level because it will greatly impact the performance of jobs and add a lot of debug information to all of the job logs.



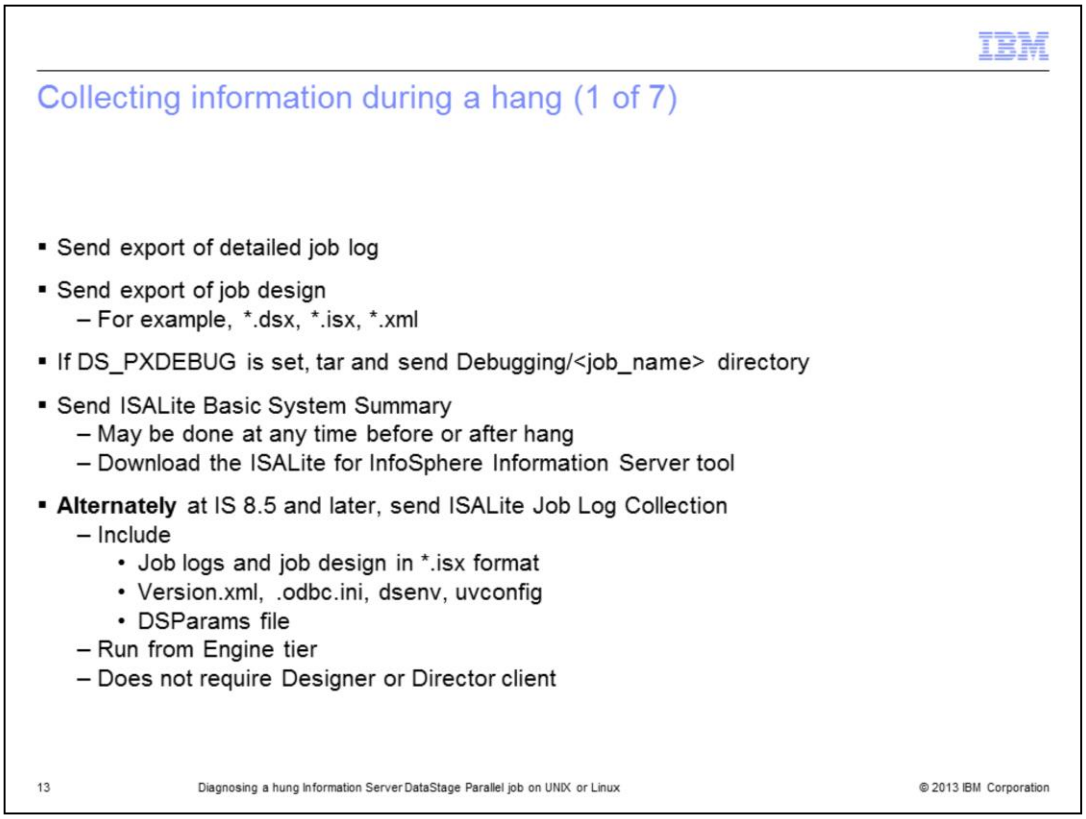
Environment variables to set (3 of 3)

- Create user-defined environment variable by way of DataStage Administrator
  - APT_NO_PM_SIGNAL_HANDLERS
  - Set to 1 at project level
  - Allows UNIX/Linux system to terminate all associated processes caused by a database client core dump
- Ensure UNIX/Linux system permits core files to be created
  - Set ulimit -c unlimited
- See Technotes for additional information
  - How to get a stack trace for failing processes in a DataStage Parallel Job, AIX platform
    - http://www.ibm.com/support/docview.wss?uid=swg21461160
  - How to get a stack trace for failing processes in a DataStage Parallel Job, Linux platforms
    - http://www.ibm.com/support/docview.wss?rs=0&uid=swg21461167

12 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

Many times a hang is caused when a database client core dumps. When this occurs, often the database operators or connectors will sit and wait forever for a response from the client that will never be sent due to the core dump and therefore, the job hangs. Setting APT_NO_PM_SIGNAL_HANDLERS allows the UNIX or Linux system to terminate all the processes that are associated with the core dump and a core file is generated.

If setting APT_NO_PM_SIGNAL_HANDLERS results in a core file being generated, ensure that the system permits core files to be created, and gather a stack trace on the core file. See the technotes listed on this slide for examples on AIX and Linux.



IBM

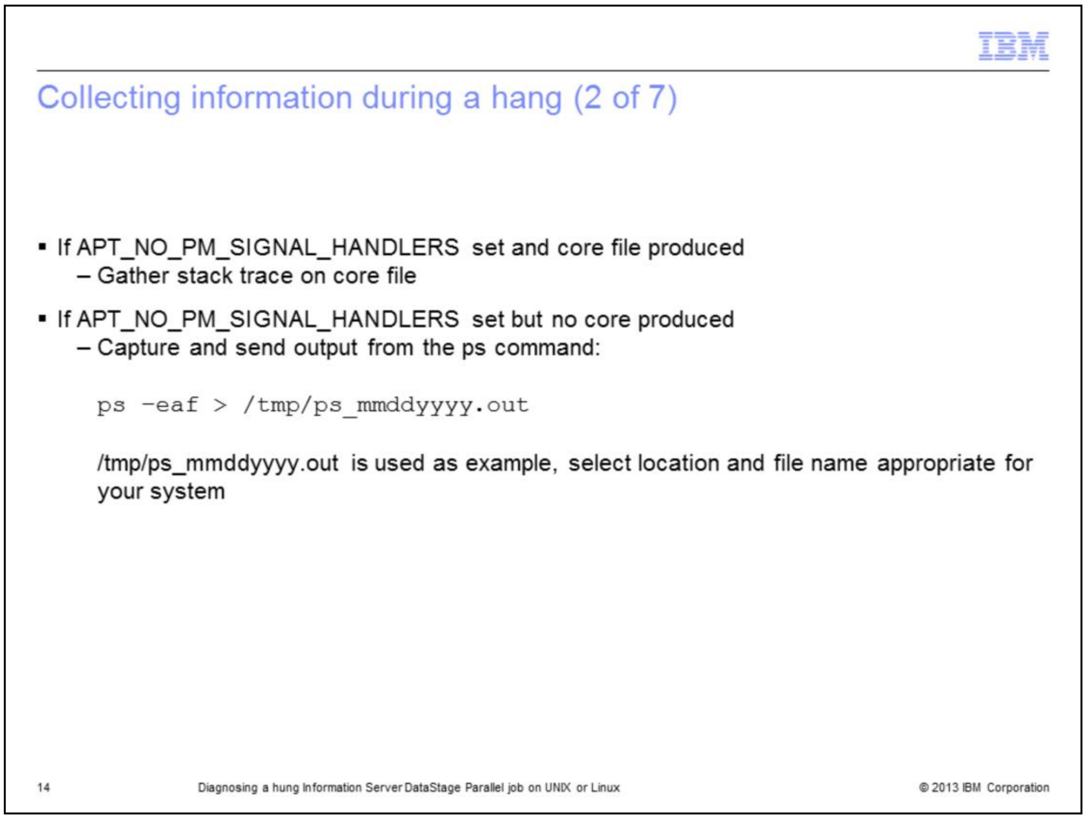
Collecting information during a hang (1 of 7)

- Send export of detailed job log
- Send export of job design
  - For example, *.dsx, *.isx, *.xml
- If DS_PXDEBUG is set, tar and send Debugging/<job_name> directory
- Send ISALite Basic System Summary
  - May be done at any time before or after hang
  - Download the ISALite for InfoSphere Information Server tool
- **Alternately** at IS 8.5 and later, send ISALite Job Log Collection
  - Include
    - Job logs and job design in *.isx format
    - Version.xml, .odbc.ini, dsenv, uvconfig
    - DSParams file
  - Run from Engine tier
  - Does not require Designer or Director client

13 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

Once the environment variables have been set and the hang is reproduced, the next step is to collect the log information. It is important to send the detailed job log and an export of the job design of the hanging job. If DS_PXDEBUG is set, tar and send the Debugging/<job_name> directory located under the project directory. The ISALite Basic System Summary can be done at any time before or after the hang.

Alternately, at Information Server version 8.5 and later, send an ISALite Job Log Collection. This collects all the job related information requested in one step and it includes additional information such as Version.xml, .odbc.ini, dsenv, uvconfig, DSParams, and more. It also includes a collection report.



Collecting information during a hang (2 of 7)

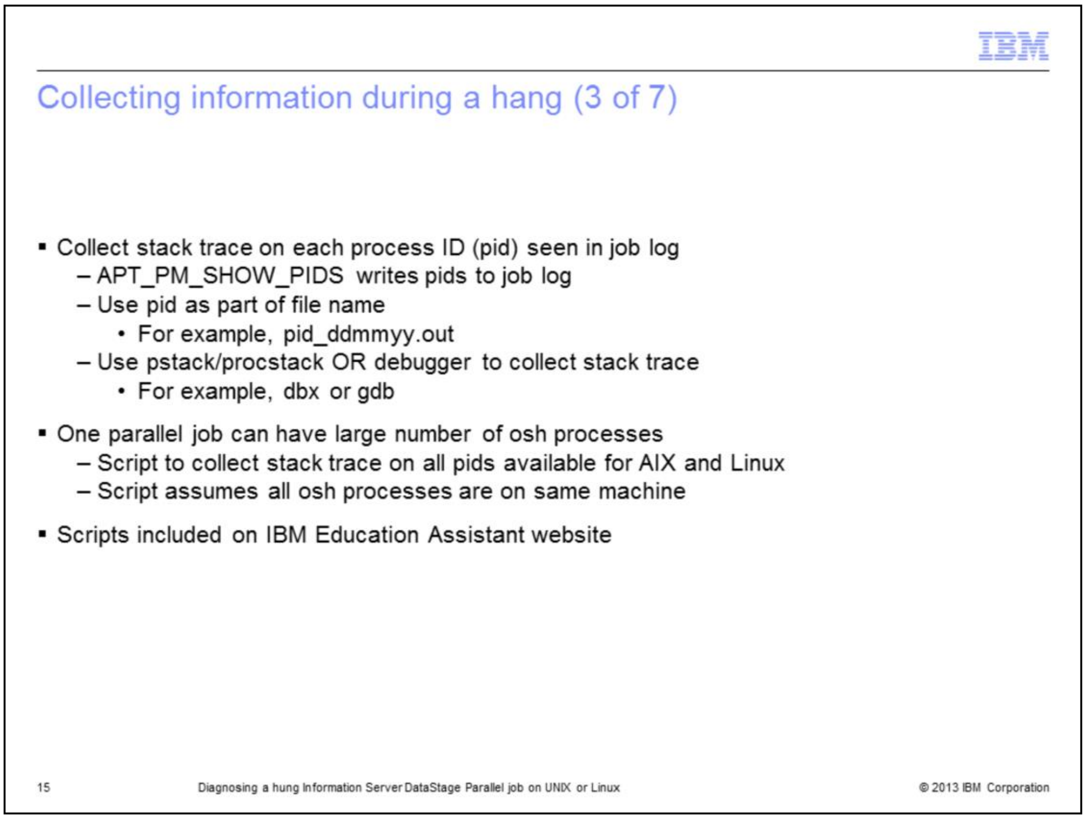
- If APT_NO_PM_SIGNAL_HANDLERS set and core file produced
  - Gather stack trace on core file
- If APT_NO_PM_SIGNAL_HANDLERS set but no core produced
  - Capture and send output from the ps command:

```
ps -eaf > /tmp/ps_mmddyyyy.out
```

/tmp/ps_mmddyyyy.out is used as example, select location and file name appropriate for your system

14 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

If APT_NO_PM_SIGNAL_HANDLERS is set in the job and a core file is produced, get a stack trace on the core file and send that along with the logs and job export. If the job does not end and core dump, run the ps command shown on this slide to capture all the system processes and send the output file to Support with the rest of the information collected.



Collecting information during a hang (3 of 7)

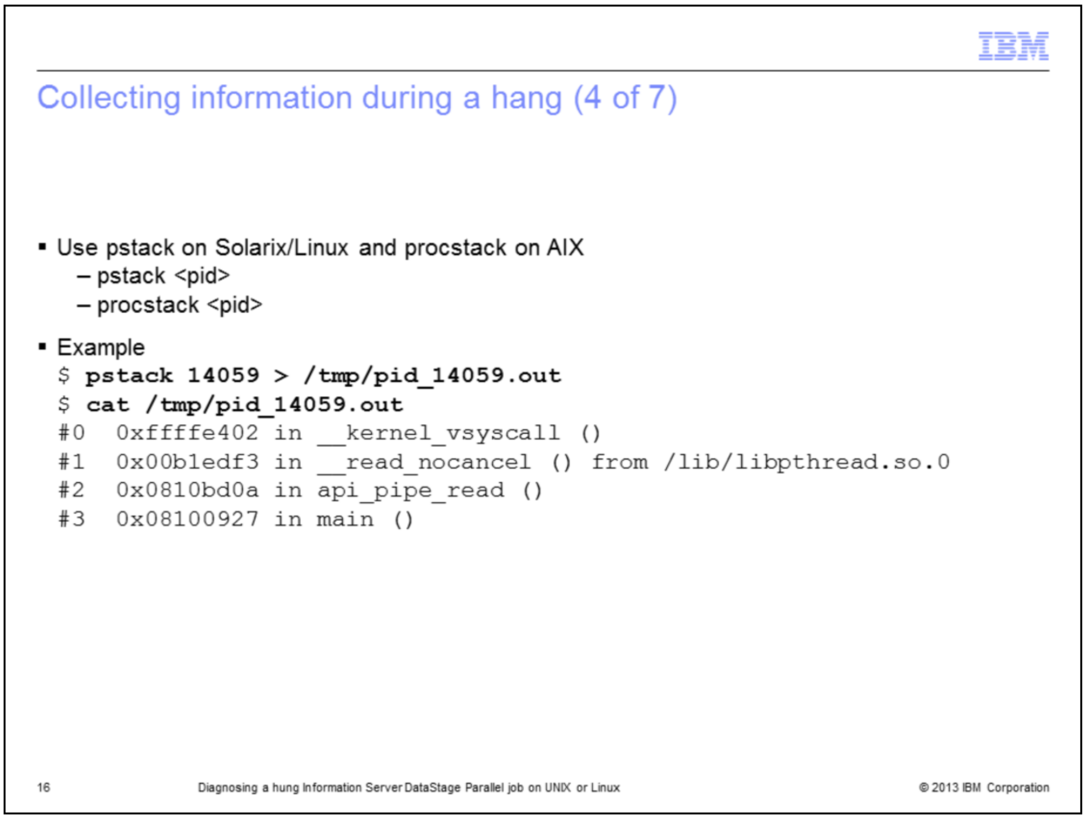
- Collect stack trace on each process ID (pid) seen in job log
  - APT_PM_SHOW_PIDS writes pids to job log
  - Use pid as part of file name
    - For example, pid_ddmmyy.out
  - Use pstack/procstack OR debugger to collect stack trace
    - For example, dbx or gdb
- One parallel job can have large number of osh processes
  - Script to collect stack trace on all pids available for AIX and Linux
  - Script assumes all osh processes are on same machine
- Scripts included on IBM Education Assistant website

15 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

If the job processes are hung, the call stack from each osh process in the job provides information on the state of the osh process at the time of the hang. Collecting the stack traces is critical for debugging the issue. A stack trace is needed for each pid listed in the job log.

There can be a large number of pids that need stack traces. A script is available to automate this process on AIX and Linux. The script assumes that all osh processes are on the same machine.

The scripts are available for download from this IBM Education Assistant site.



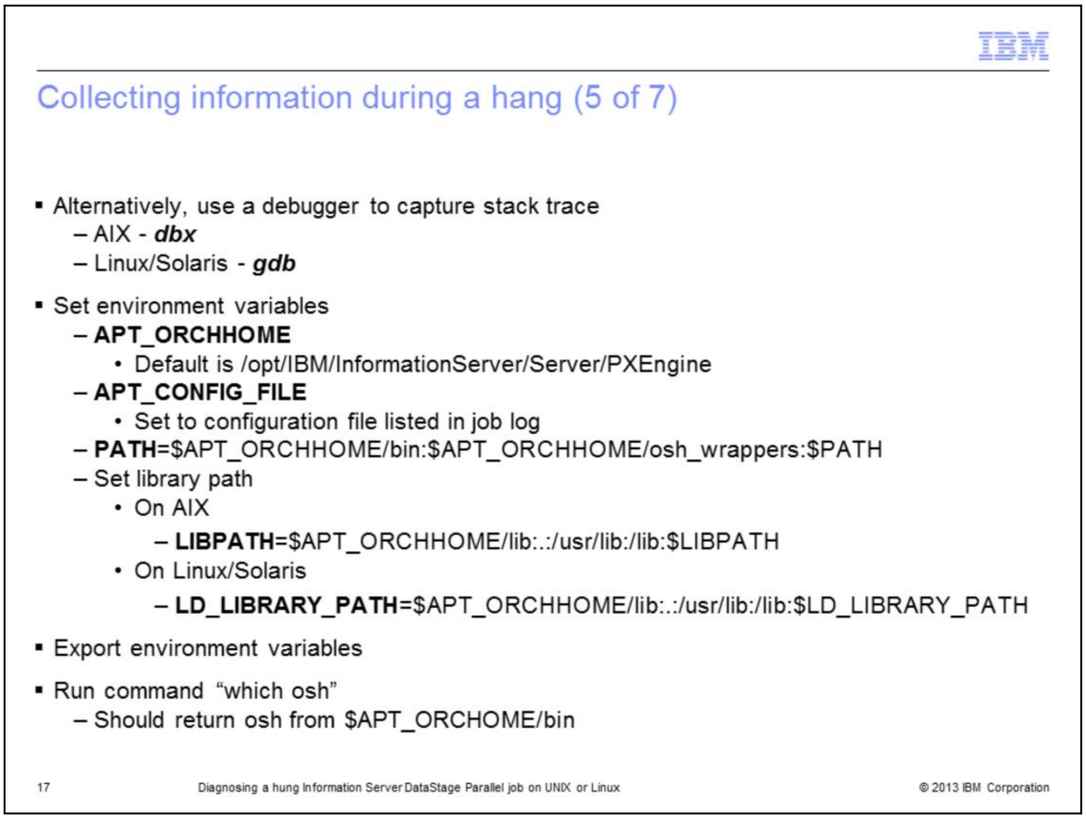
Collecting information during a hang (4 of 7)

- Use pstack on Solarix/Linux and procstack on AIX
  - pstack <pid>
  - procstack <pid>
- Example

```
$ pstack 14059 > /tmp/pid_14059.out
$ cat /tmp/pid_14059.out
#0  0xffffe402 in __kernel_vsyscall ()
#1  0x00b1edf3 in __read_nocancel () from /lib/libpthread.so.0
#2  0x0810bd0a in api_pipe_read ()
#3  0x08100927 in main ()
```

16 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

Use pstack on Linux or Solaris and procstack on AIX. See the information displayed on this slide for the syntax and an example. Remember to include the pid in the file name.

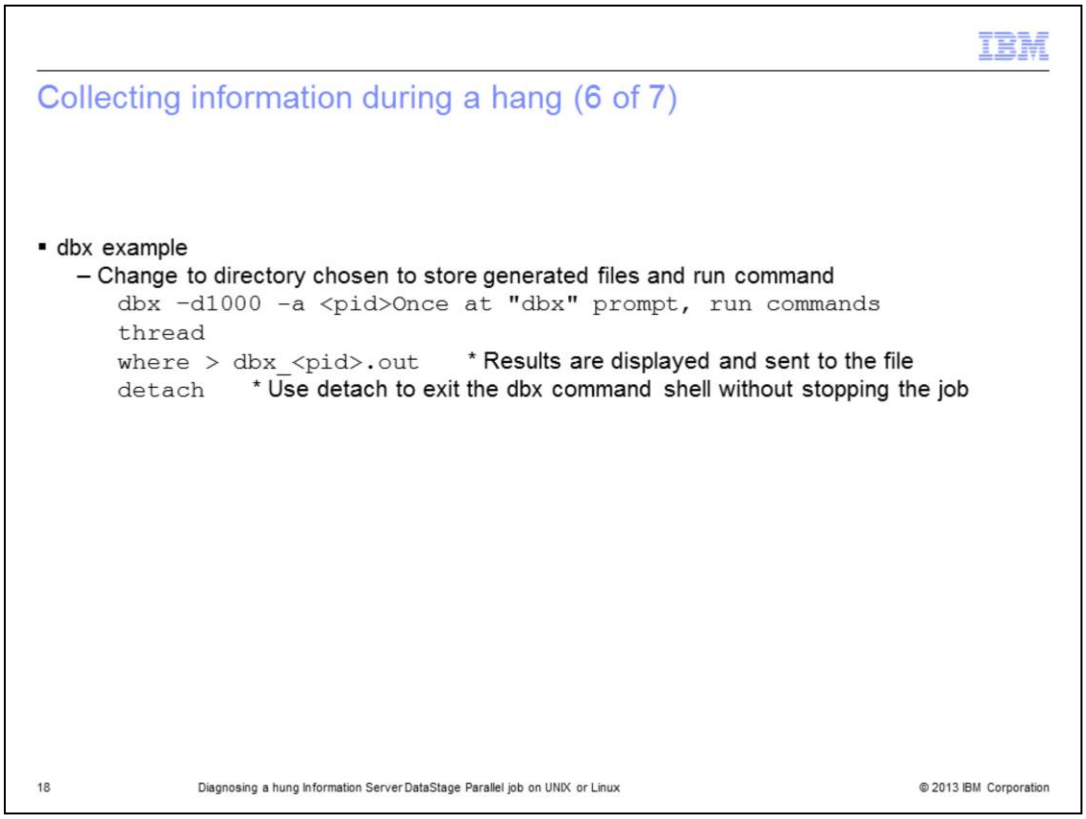


Collecting information during a hang (5 of 7)

- Alternatively, use a debugger to capture stack trace
  - AIX - ***dbx***
  - Linux/Solaris - ***gdb***
- Set environment variables
  - **APT_ORCHHOME**
    - Default is /opt/IBM/InformationServer/Server/PXEngine
  - **APT_CONFIG_FILE**
    - Set to configuration file listed in job log
  - **PATH**=$APT_ORCHHOME/bin:$APT_ORCHHOME/osh_wrappers:$PATH
  - Set library path
    - On AIX
      - **LIBPATH**=$APT_ORCHHOME/lib:.:/usr/lib:/lib:$LIBPATH
    - On Linux/Solaris
      - **LD_LIBRARY_PATH**=$APT_ORCHHOME/lib:.:/usr/lib:/lib:$LD_LIBRARY_PATH
- Export environment variables
- Run command "which osh"
  - Should return osh from $APT_ORCHOME/bin

17 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

Alternatively, a debugger can be used to capture the stack trace on each pid in the job log. The tool used on Linux and Solaris is gdb and on AIX it is dbx. Set and export the environment variables as displayed on this slide. Then, confirm "which osh" returns the expected location.



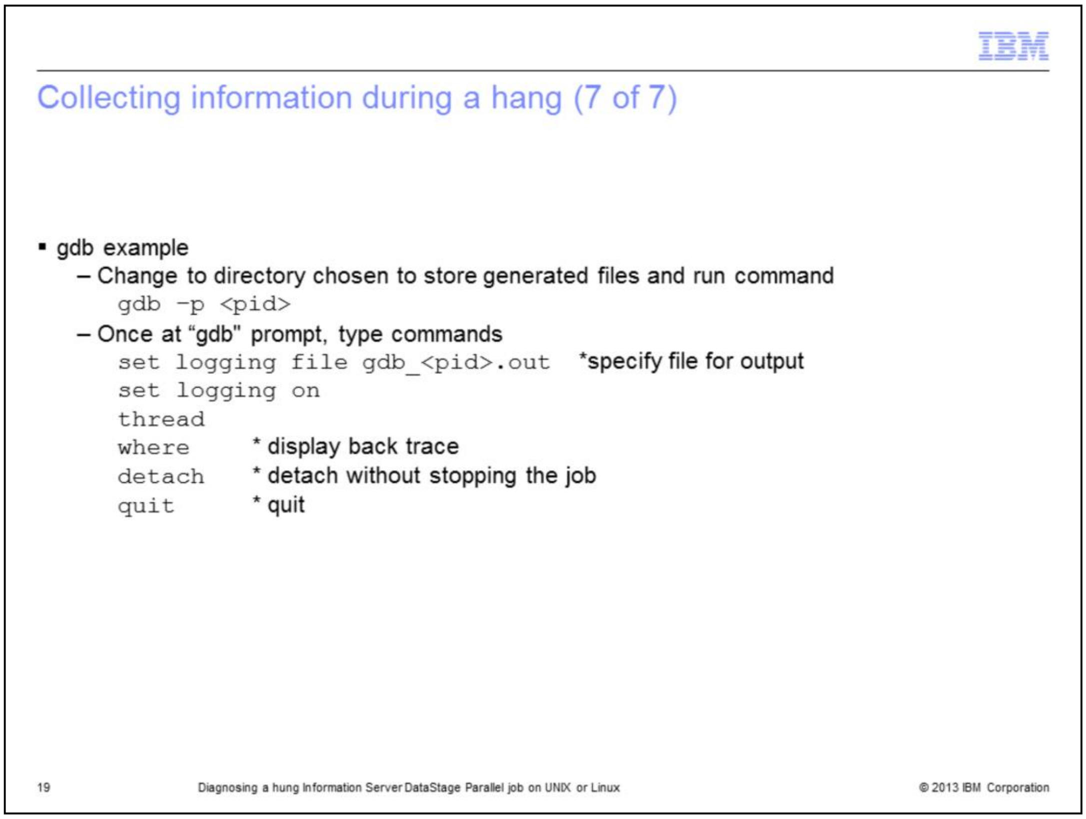
Collecting information during a hang (6 of 7)

- dbx example
  - Change to directory chosen to store generated files and run command

```
dbx -d1000 -a <pid>Once at "dbx" prompt, run commands
thread
where > dbx_<pid>.out     * Results are displayed and sent to the file
detach     * Use detach to exit the dbx command  shell without stopping the job
```

18 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

This slide has an example of using the debugger dbx to collect a stack trace and send the output to a file. It is important to use detach to exit the dbx command shell and not exit. The job is stopped if exit is used.



Collecting information during a hang (7 of 7)

- gdb example
  - Change to directory chosen to store generated files and run command

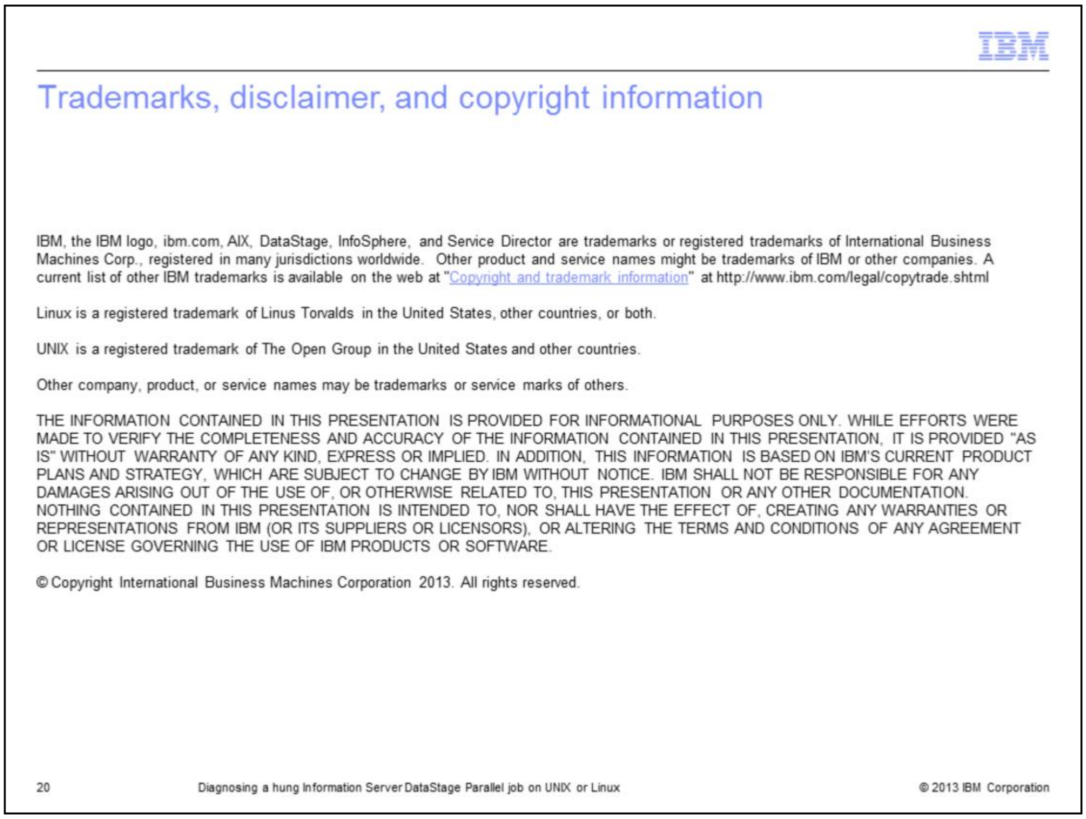
```
gdb -p <pid>
```

  - Once at "gdb" prompt, type commands

```
set logging file gdb_<pid>.out   *specify file for output
set logging on
thread
where        * display back trace
detach       * detach without stopping the job
quit         * quit
```

19 Diagnosing a hung Information Server DataStage Parallel job on UNIX or Linux © 2013 IBM Corporation

This slide displays an example of using the debugger gdb to collect a stack trace. Capture the output to a file with the pid as part of the file name, for example, gdb_pid.out. It is important to use detach before quitting the command shell. The job is stopped if quit is used without detaching first.



Trademarks, disclaimer, and copyright information