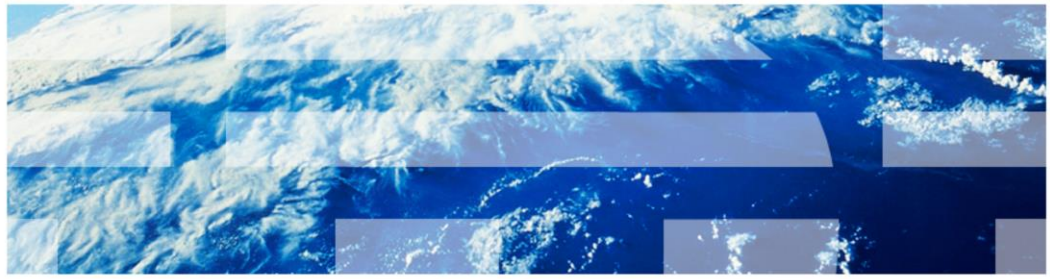
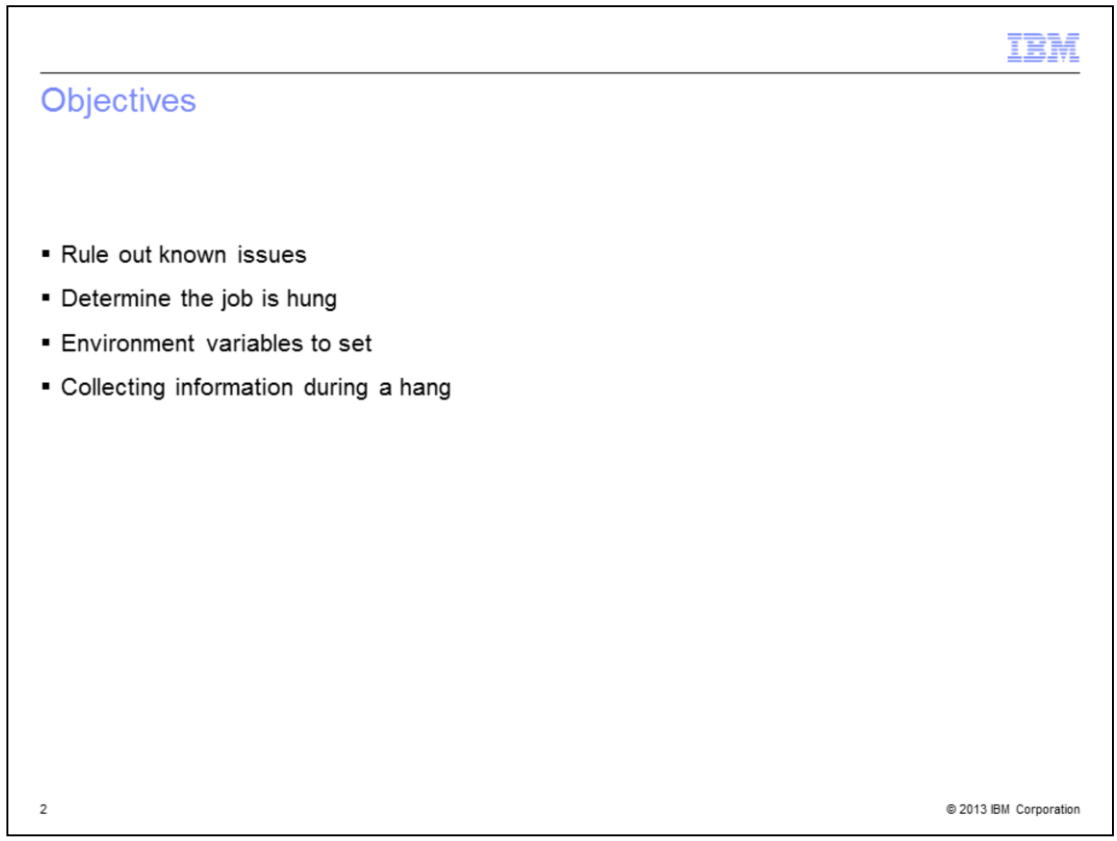
IBM

InfoSphere DataStage

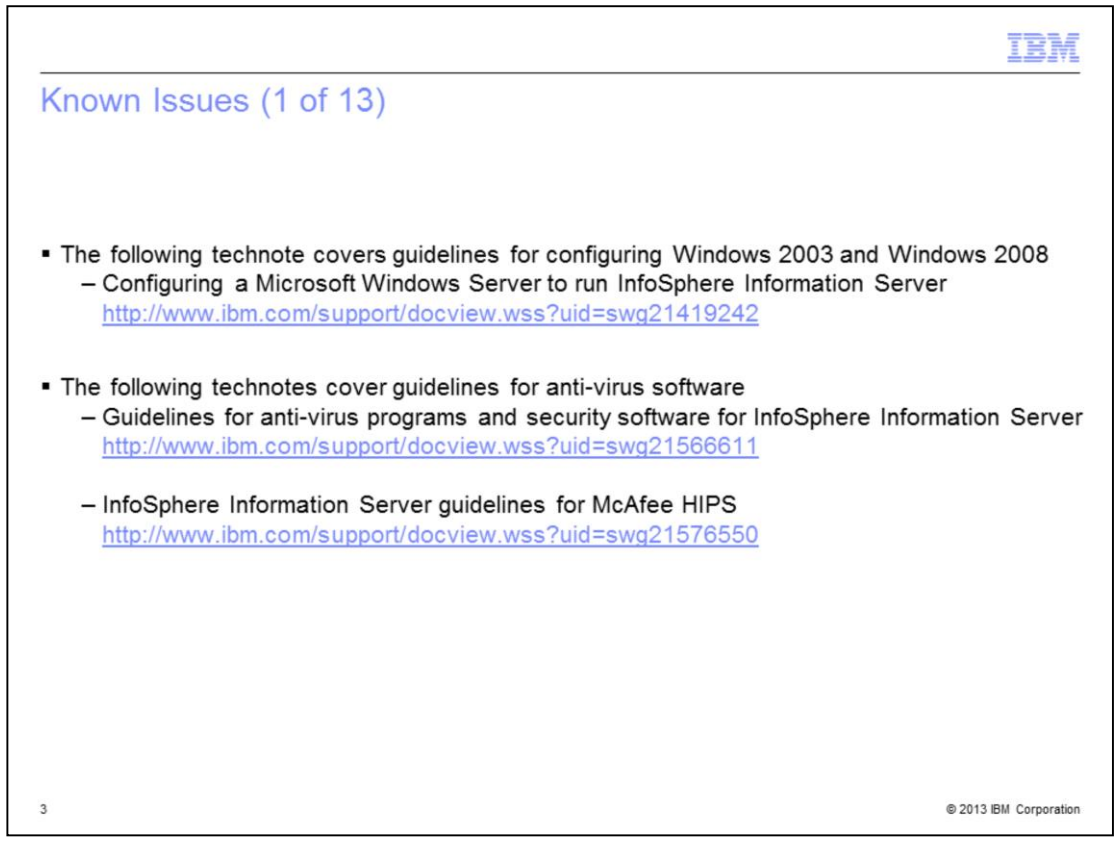
Diagnosing a hung Information Server DataStage Parallel Job



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.



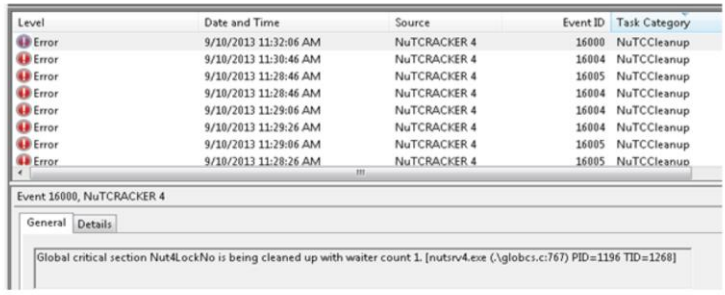
The objectives of this presentation are to rule out known issues, show how to determine if a parallel job is hung, and if so, it will describe the environment variables that need to be set and what information to collect during the hang.



The technotes that are displayed on this slide describe guidelines to configure your Windows server and anti-virus software. Anti-virus software can impact performance and interfere with Information Server. It is important to exclude Information Server directories from scans. There are also guidelines for McAfee Host Intrusion Prevention for Server with InfoSphere Information Server. See the Technotes that are listed on this slide for additional information.

Known Issues (2 of 13)

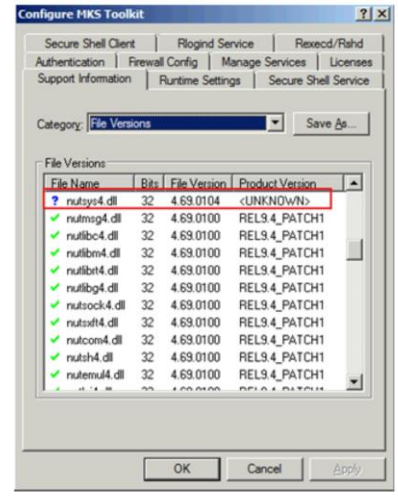
- DataStage 8 Only
  - MKS Toolkit can be the source of job hangs
  - Check for Nutcracker errors in the Windows application event log
    - Global critical section Nut4SemIdC is being cleaned up with waiter count 1. [nutsrv4.exe (.\globcs.c:767) PID=pppp TID=tttt]
    - Process pppp appears to have hung. Cleaning up global critical sections held by this process. [nutsrv4.exe (sem.cpp:1160) PID=pppp TID=tttt]
    - Failed to create fork() child process. [forktest.exe (fork.c:719) PID=5780 TID=5088]
  - Upgrading to MKS Toolkit 9.4 FP1 + HotFix CFS31959



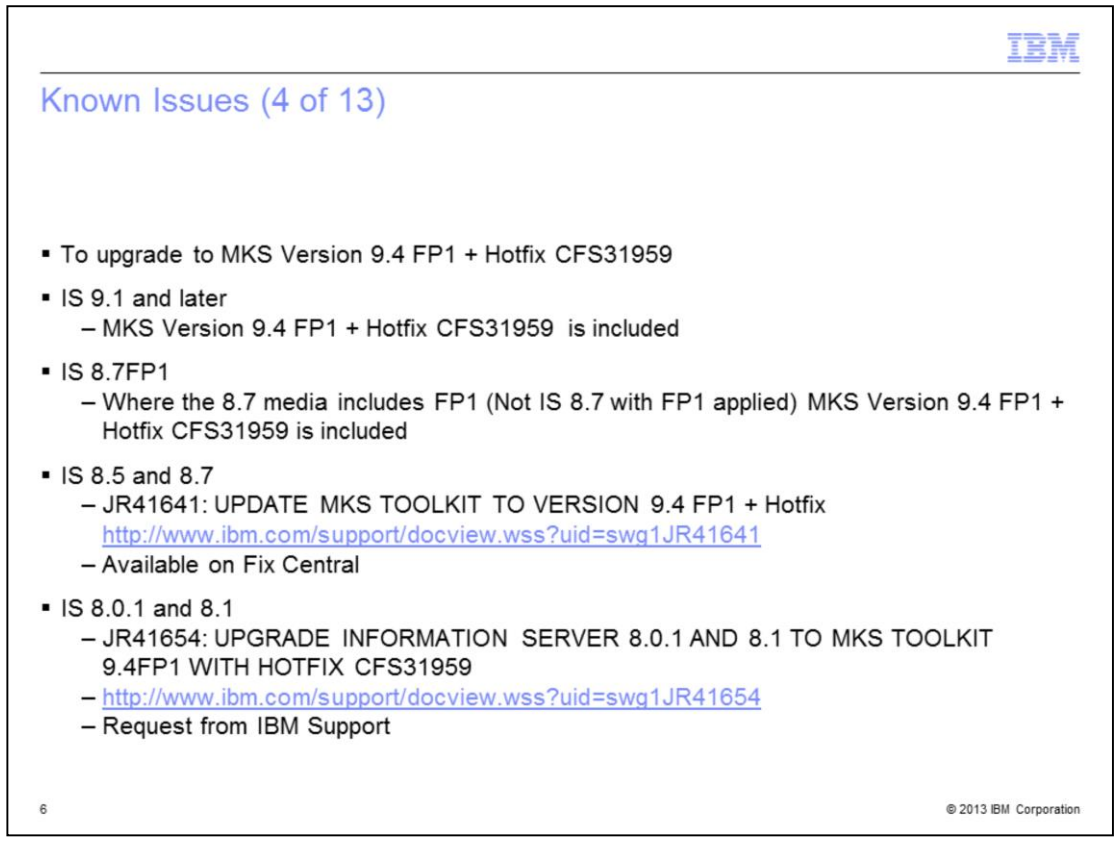
At version 8 of DataStage, the MKS Toolkit can be the source of job hangs. It is recommended that any busy Windows system be upgraded to the latest MKS Toolkit version, but even more so if the errors listed on this slide are seen in the Windows application event log. Upgrading to MKS Toolkit 9.4 FP1 + HotFix CFS31959 will resolve the errors seen on the this slide.

Known Issues (3 of 13)

- Check the version of MKS Toolkit on the engine tier
  - Select Control Panel > Configure MKS Toolkit
  - Select the Support Information tab
  - Click on the Category drop down list
  - Select File Versions
  - Find the version for the nutsys4.dll for 64 and 32 bit
  - Version after FP1 + HotFix CFS31959 is applied
    - nutsys4.dll 64 and 32 bit => 4.69.0104



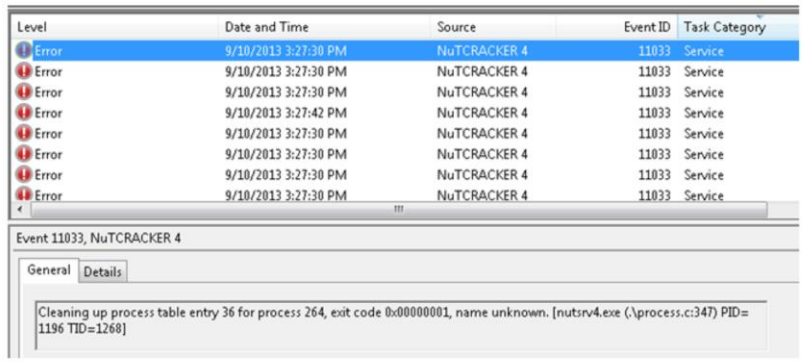
If the exact version of the MKS toolkit is not known, follow the steps listed on this slide to check the version of MKS Toolkit installed on the engine tier. When FP1 + HotFix CFS31959 is applied, the nutsys4.dll will be version 4.69.0104 for both 64 and 32 bit.



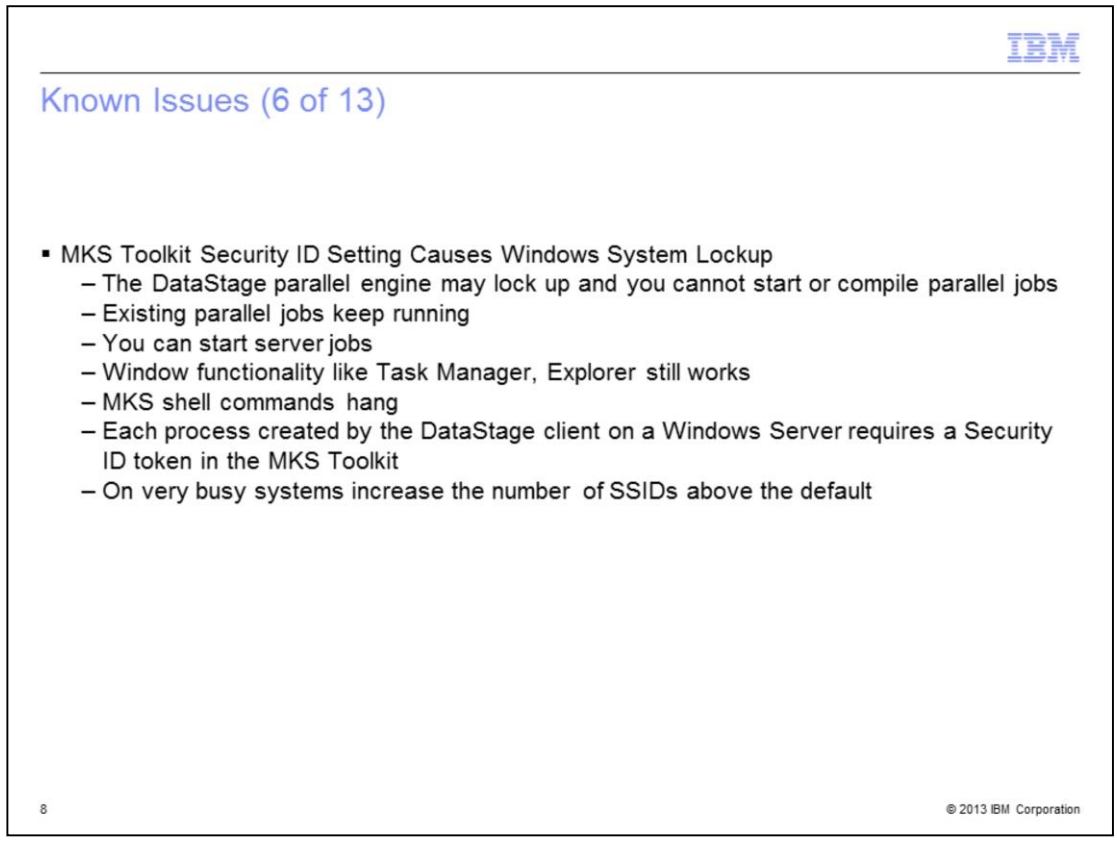
If you determined that you need to upgrade to MKS Version 9.4 FP1 + Hotfix CFS31959, you need to apply the appropriate patch for your release of Information Server. This slide displays the fix that is required for the different versions of DataStage. The MKS upgrades use the MKS installer, not the suite patch installer. Follow directions carefully. At Information Server version 9.1 and later MKS 9.4 FP1 + Hotfix CFS31959 is installed by default. Starting at IS 9.1 MKS is no longer used in the parallel engine layer, except where scripts are required. Instead native Windows APIs are called. MKS is still installed since the suite relies on MKS shell services and system tools.

Known Issues (5 of 13)

- The following errors are normal for a busy system
  - Cleaning up process table entry 358 for process pppp, exit code 0x000000ff, name unknown. [nutsrv4.exe (.\process.c:347) PID=pppp TID=ttt]
  - SIGKILL signal has not caused process to die. Child watcher committing suicide. [osh.exe (.\ncchild.c:544) PID=pppp TID=tttt]



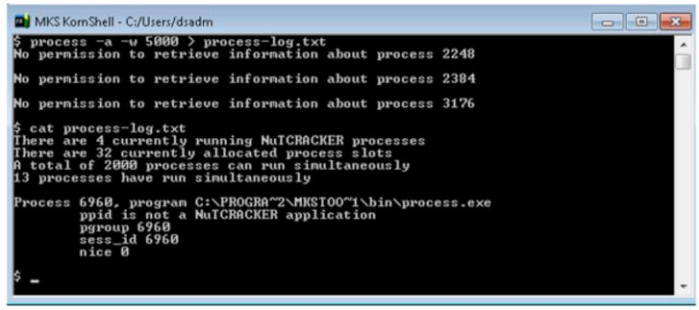
It is important to note that not all errors in the Application log indicate a problem. This slide displays errors from the Windows application event log that are normal for a busy Windows system and do not indicate an issue.



Another issue that you might encounter is the InfoSphere DataStage parallel engine on Microsoft Windows might 'lock up'. Windows functionality, for example Explorer, Task Manager, and so on, will still work. Existing parallel jobs, seen as OSH.exe in task manager, keep running, but you cannot start or compile any parallel jobs. However, you can start InfoSphere DataStage server jobs. Each process that is created by the DataStage client on a Windows server requires a Security ID token in the MKS Toolkit. On very busy systems, it is possible to run out of tokens, which will cause the parallel engine to lock up. It is necessary to increase the number of SSIDs above the default to prevent lock up.

Known Issues (7 of 13)

- Diagnosing the issue
  - Obtain an MKS NuTCRACKER process report dump.
    - Open a Korn shell by clicking **Start->Run->ksh**
    - From the Korn shell enter the following command:
      `process -a -w 5000`



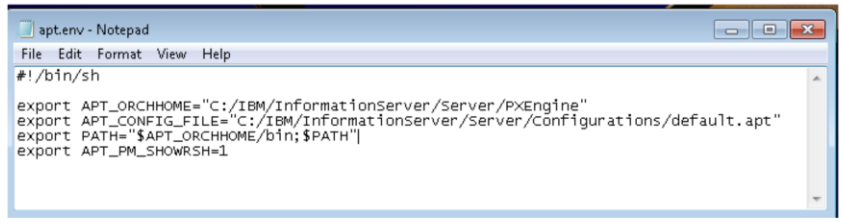
Note: it is possible this command may hang and not produce results

The process command reports how many NuTCRACKER platform processes are running and displays information on active NuTCRACKER platform processes. In the example above there are 4 NuTCRACKER processes currently running. It also shows how many processes can run simultaneously.

It is possible the process command may hang and not produce any results. If this happens continue to the next test.

Known Issues (8 of 13)

- Check that you can successfully run a parallel job as follows:
  - Create an environment file in the PXEngine directory
  - Use an editor such as notepad
  - Save the file as apt.env

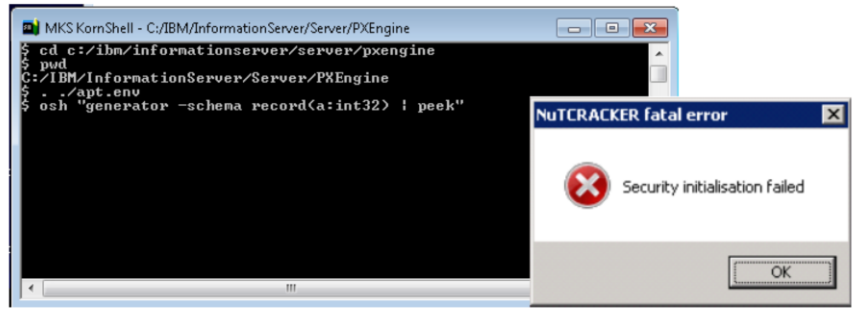


10

The next test is to see if you can successfully run a parallel job. To run a simple parallel job from command line first create an environment file in the PXEngine directory, by default this is C:\IBM\InformationServer\Server\PXEngine. Create the file using an editor such as Windows Notepad and add the lines shown above. Then save the file as apt.env. Note, if you are using Windows Notepad be sure to set the "**Save as type**" to "**All Files**" and enter **apt.env** as the File name. Do not use a .TXT file name extension.

Known Issues (9 of 13)

- Test the parallel engine by executing a simple parallel job as follows:
- Invoke Korn shell select Start> Run> ksh
- Change to the PXEngine directory
  `cd C:/IBM/InformationServer/Server/PXEngine`
- Source the apt.env environment file
  `. ./apt.env`
- Run a test
  `osh "generator -schema record(a:int32) | peek"`



11

After creating the environment file, you can test the parallel engine by executing a simple parallel job from the shell or command level as follows; invoke the Korn shell by selecting Start> Run> ksh. Then change directory to the PXEngine directory, by default C:\IBM\InformationServer\Server\PXEngine. Note from the Korn shell prompt you need to use forward slashes instead of backward slashes. Also adjust the path if you installed to a different directory or drive letter. Now source the apt.env file you created by typing ". ./apt.env". The last step is to run the test parallel job using the following command:

osh "generator –schema record(a:int32) | peek"

The job should execute normally and you should not experience any hang or error message dialog box. If an error dialog appears with the error, "Security initialisation failed" when trying to run the osh test, you will need to increase the number of MKS NuTCRACKER security IDs as described in the following slides.

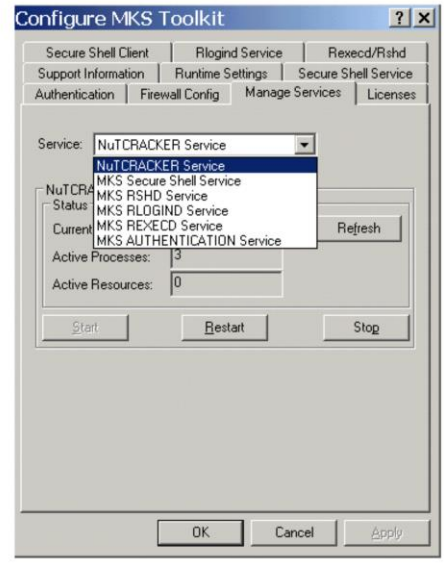
Known Issues (10 of 13)

- Resolving the issue
- The problem can be avoided by changing a parameter in the MKS Control Panel
  - Follow these steps
    - Make sure there are no uvsh.exe or osh.exe processes running on the server
      - This will prevent changed settings from taking effect
    - Stop all the DataStage services
      - InfoSphere Engine Resource Service
      - DataStage Telnet Service
      - DSRPC Service

The issue can be resolved by changing the Max Number of Security ID parameter in the MKS control panel. To do this, you must be sure that there are no uvsh.exe or osh.exe processes running. If these processes are running, it will prevent the changes from taking effect. Next, you need to stop all of the DataStage services.

Known Issues (11 of 13)

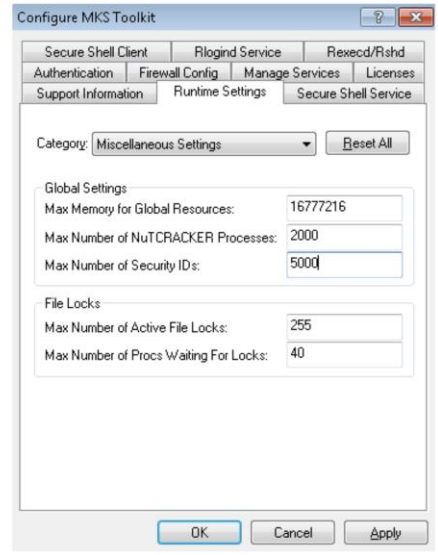
- Resolving the issue continued
  - Open Microsoft Windows control panel
  - Click Configure MKS Toolkit
  - Select the 'Manage Services' tab
  - Starting with the bottom service shown in the dropdown, Stop each MKS Service.
  - Click the 'Refresh' button
    - 'Active Processes' box should display zero



Next, you need to open the Microsoft Windows control panel and click Configure MKS Toolkit. Next, click the Manage Services tab. Click the Services drop down. Starting at the bottom of the list and working backwards, stop each MKS service that is listed in the drop-down. Click the refresh button after all the services have been stopped. The Active Processes box should now display 0.

Known Issues (12 of 13)

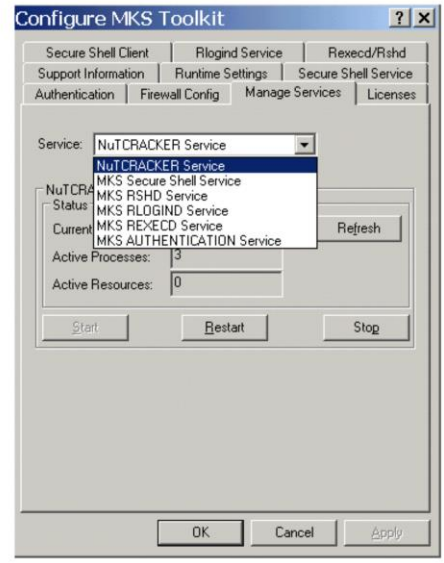
- Resolving the issue continued
  - Select the "Runtime Settings" tab
  - Select 'Miscellaneous Settings' from the Category dropdown
  - Set Max Number of Security IDs' to 5000



Next, you need to click the Runtime Settings tab. Click the Category dropdown and select Miscellaneous Settings. Change the Max Number of Security IDs to 5000. The default is 2500.

Known Issues (13 of 13)

- Resolving the issue continued
  - Select the 'ManageServices' tab
  - Restart the services in the order shown in the dropdown, starting at the top
  - Restart the DataStage Services

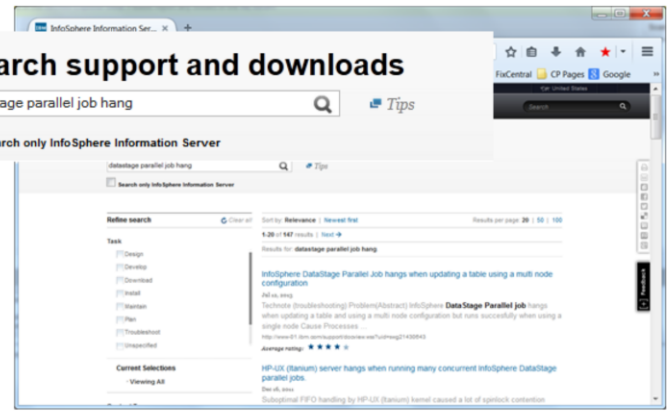


After the parameter is changed, restart the MKS services and then restart the DataStage services. Select the ManageServices tab and select each MKS service in the list starting at the top and working down the list. For each service, click the service name in the drop-down and then click the Restart button. Click the OK button when finished and restart all of the DataStage services.

Searching the Knowledge Base

- Search the support portal for known issues
  http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/InfoSphere_Information_Server

- Types of results
  - Technotes
  - APARs
  - Fix lists
  - Documentation
  - Redbooks

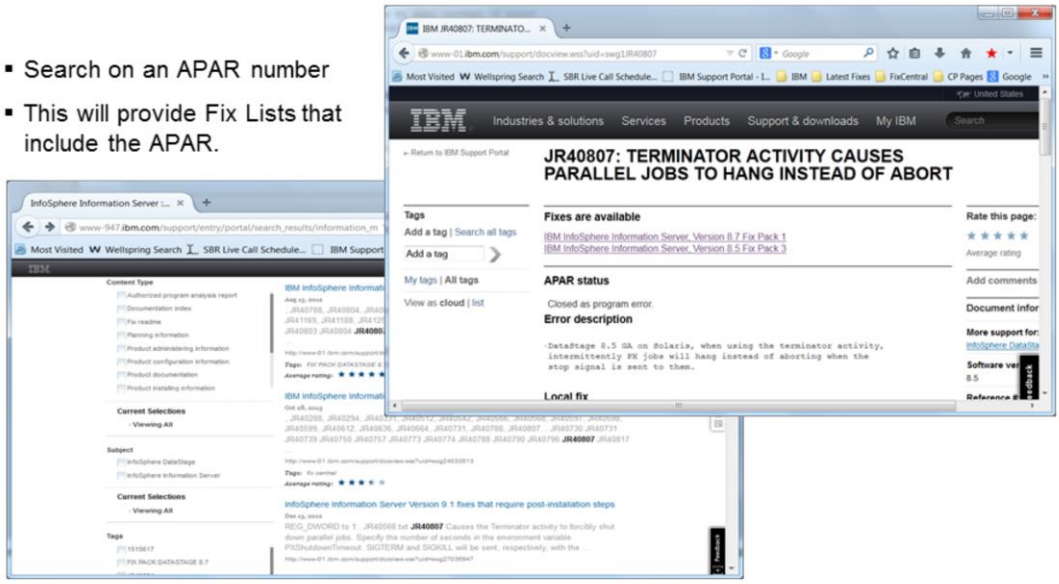


16

When you encounter an issue searching the support portal is a good starting point. Search the support portal to find technotes, APARs, Fix Lists, documentation and Redbooks. To narrow down search results select the check box to search in selected product for example, “Search only InfoSphere Information Server”.

Searching the Knowledge Base

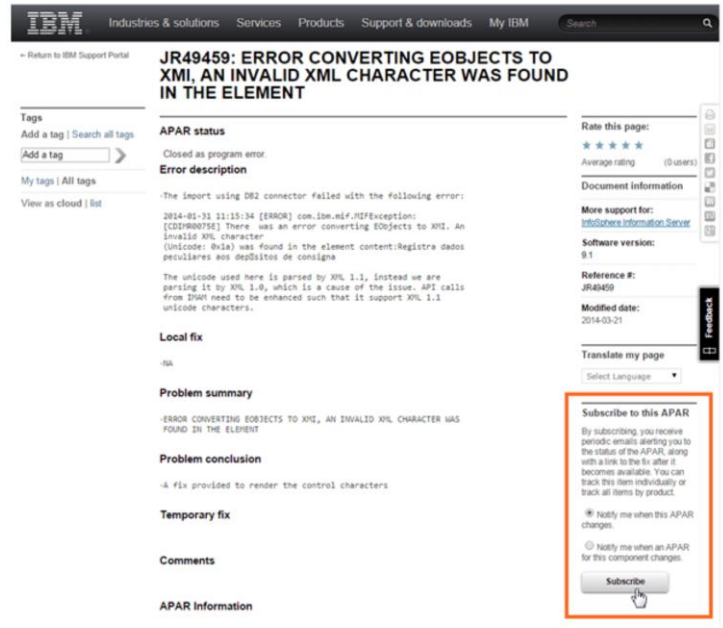
- Search on an APAR number
- This will provide Fix Lists that include the APAR.



If an APAR is found during your search. Searching on the APAR number will display any Fix Lists that include this APAR. Both the search result list and the page on the APAR will display where you can find the fix. If the fix is available in a fix pack the link at the top of the APAR page will bring you directly to the download document for the fix pack.

Notifications

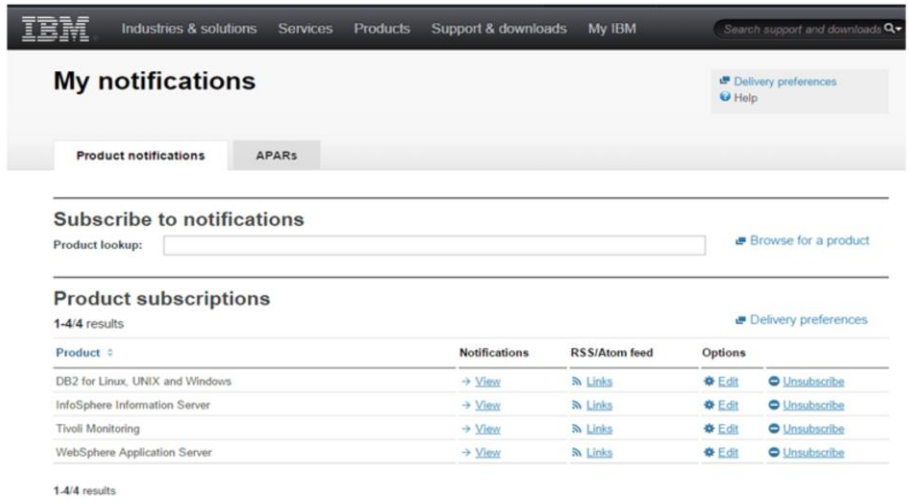
- Click the Subscribe button to be notified when an APAR changes status



Subscribe to an APAR to receive emails regarding the status of the APAR along with a link to the fix after it becomes available.

Notifications – Updated

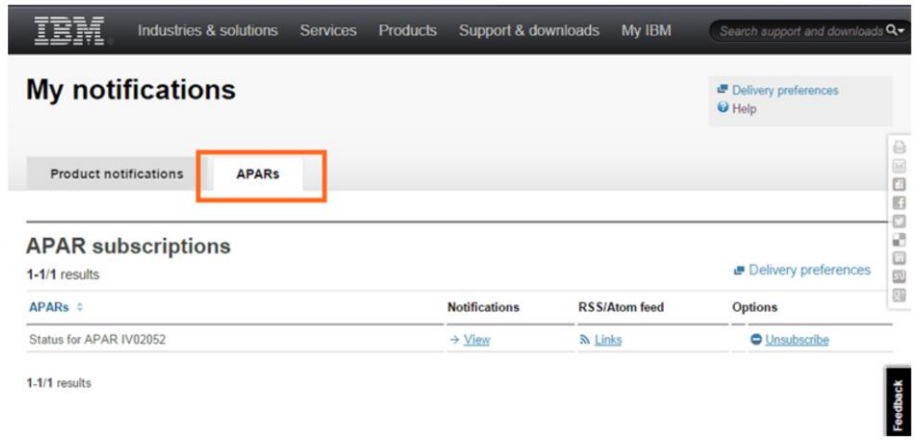
- New simpler My notifications user interface
- This is the product level subscription page where you can manage your subscriptions



There is an updated My Notifications user interface, which is simpler to use. It has a product level subscription page where you can manage subscriptions.

Notifications – Updated

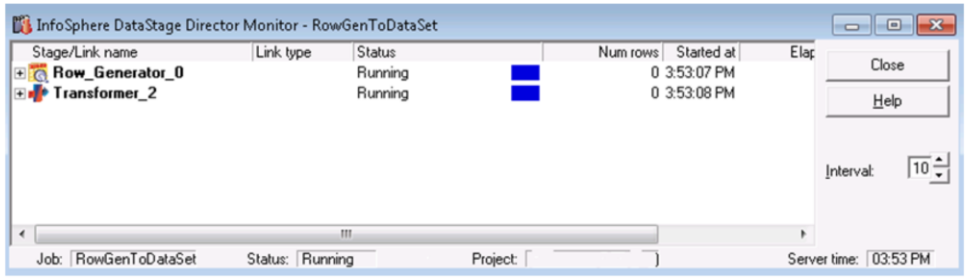
- The APAR tab lists the APARs you have subscribed to



The My notifications interface has an APAR tab, which lists all the APARs you are you have subscribed to.

Determining the parallel job is hung (1 of 8)

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

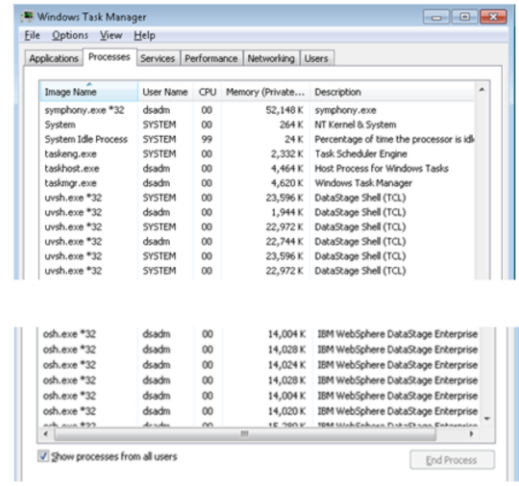


The next topic this presentation discusses is how to determine if a parallel job is hung. If a job is hung, the DataStage Director will show that the job is running but the job monitor will not show any progress. First, you need to open the DataStage Director, click the Tools menu, and then click New Monitor. If rows are still being processed, even if it is very slow, the job is not hung.

Determining the parallel job is hung (2 of 8)

On Windows

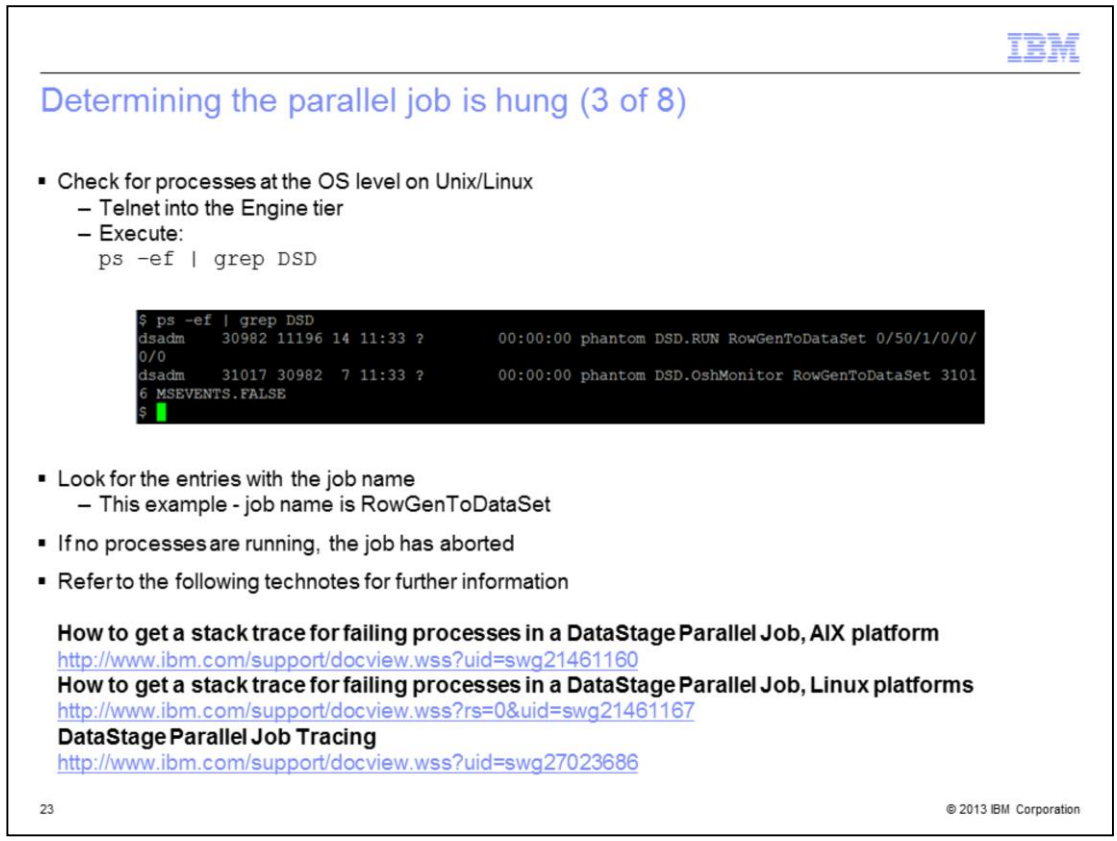
- Check for processes at operating system level
- Open Task Manager on the engine tier and look for
  - osh.exe
  - uvsh.exe
- Aborted job
  - No osh processes seen
  - Status file not updated
    - Clear status file from Director
- Difficult to associate osh processes with a particular running job
  - Show PIDs for jobs
    - Set APT_PM_SHOW_PIDS



22

If the job appears to be hung, the next step is to check for job processes at the operating system level. On Windows open Task Manager on the engine tier and look for uvsh.exe and osh.exe processes. 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.

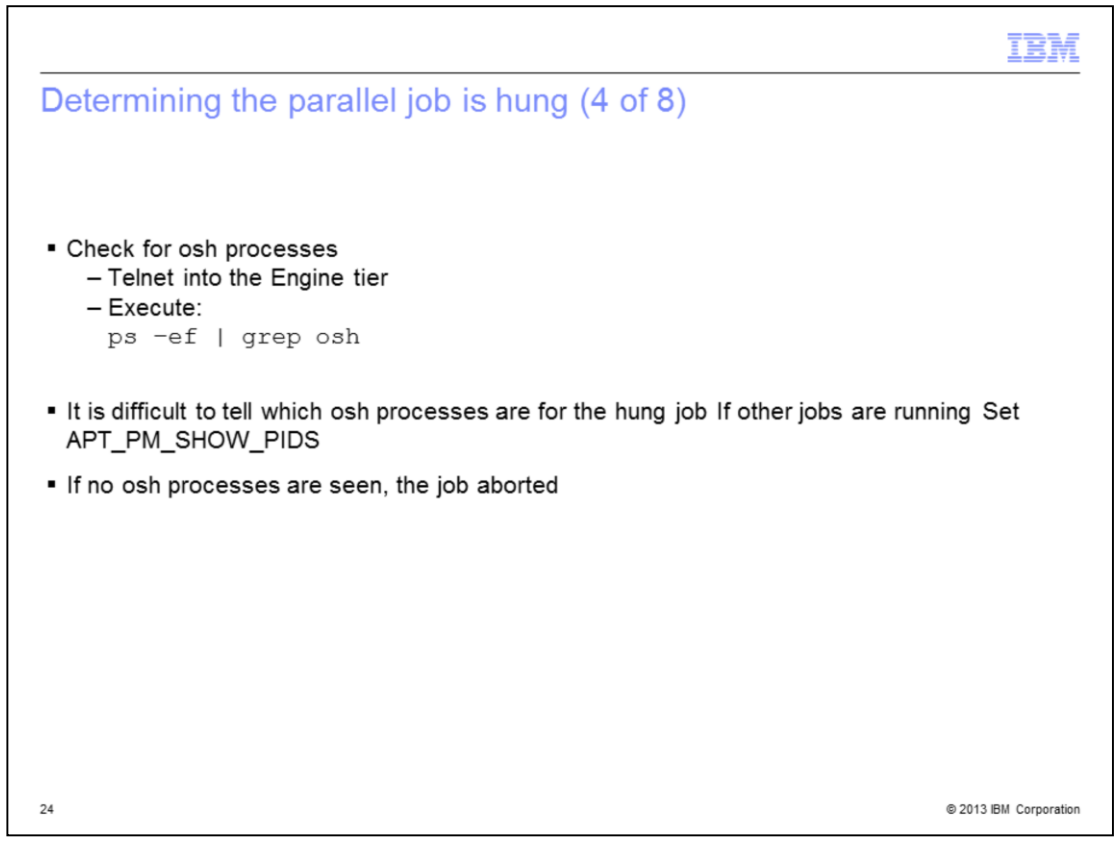
If other jobs are running on the server it is difficult to distinguish the osh processes associated with the hung job. The next section describes how to set the environment variable APT_PM_SHOW_PIDS. Setting this environment variable causes the PIDS to be written to the job log, which can then be used to find the osh processes at the OS level.



On Unix or Linux to check for job processes at the operating system level telnet into the Engine tier and execute the ps command on this slide.

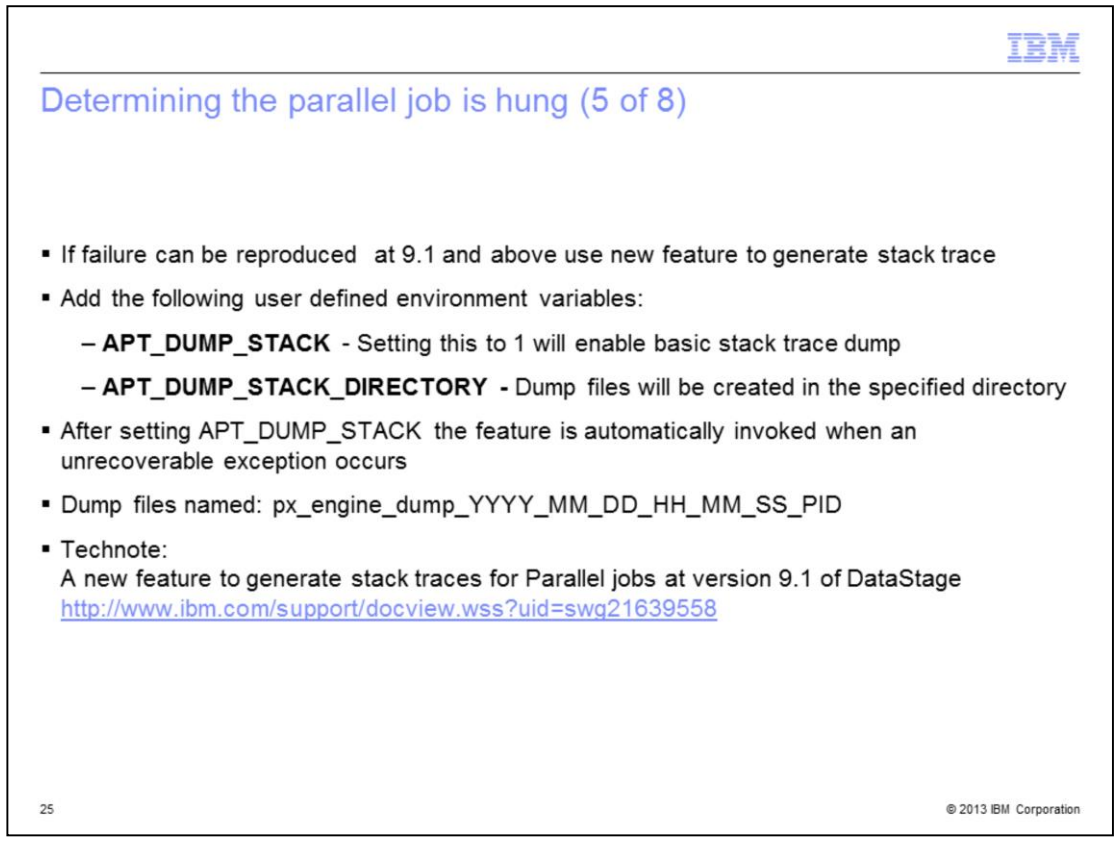
The DSD.RUN process is the first process kicked off and starts the other related processes. The DSD.OshMonitor collects information on the row counts.

If no processes are seen then most likely the job aborted, 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 listed on this slide provide examples of how to get a stack trace for AIX and Linux. If the job abort can be reproduced refer to the third link listed on the slide, which describes how to trace the job.



After locating the DSD.RUN process, the next step is to check for osh processes at the operating system level using the ps command on this slide. It is difficult to distinguish the osh processes associated with the hung job if other jobs are running on the server. 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.

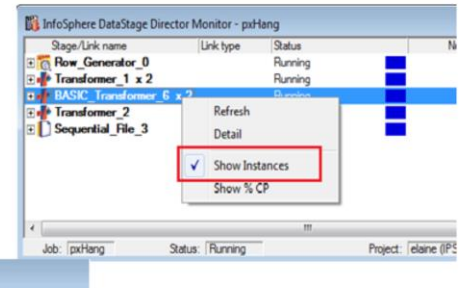
If no osh processes are returned from the ps command then the job ended but was not able to update the status before terminating. This is not considered a hung job. Look for a core file in the Project directory with a timestamp that matches the last entry in the job log and see the technotes on the previous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

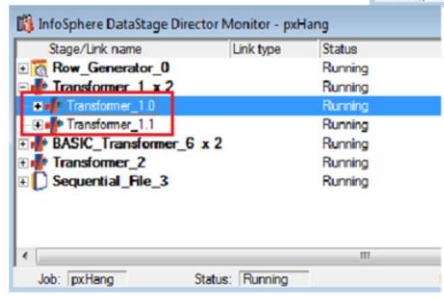


If the failure can be reproduced, starting at version 9.1 of DataStage, there is a new facility to generate stack traces and capture other valuable information for parallel jobs. The feature can be invoked by adding the user defined environment variables APT_DUMP_STACK and APT_DUMP_STACK_DIRECTORY. Set APT_DUMP_STACK to 1 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 on Unix/Linux and %TEMP% on Windows. If the job is successful a dump will not be created therefore you can leave this set to capture a dump for an intermittent issue.

Determining the parallel job is hung (6 of 8)

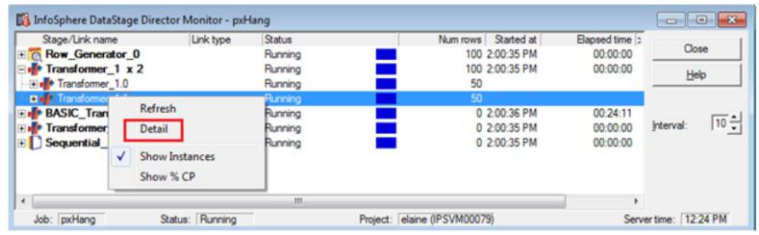
- Find the PIDs using Monitor
  - Right click and select Show instances



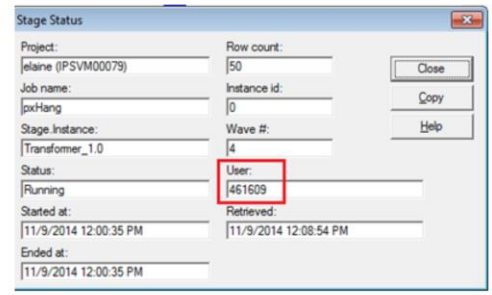


Though difficult you can find the PIDs using the monitor window. First right click and select show instances if not already checked. If running with a multi-node configuration file you will see each of the instances. The example above was run with 2 nodes.

Determining the parallel job is hung (7 of 8)

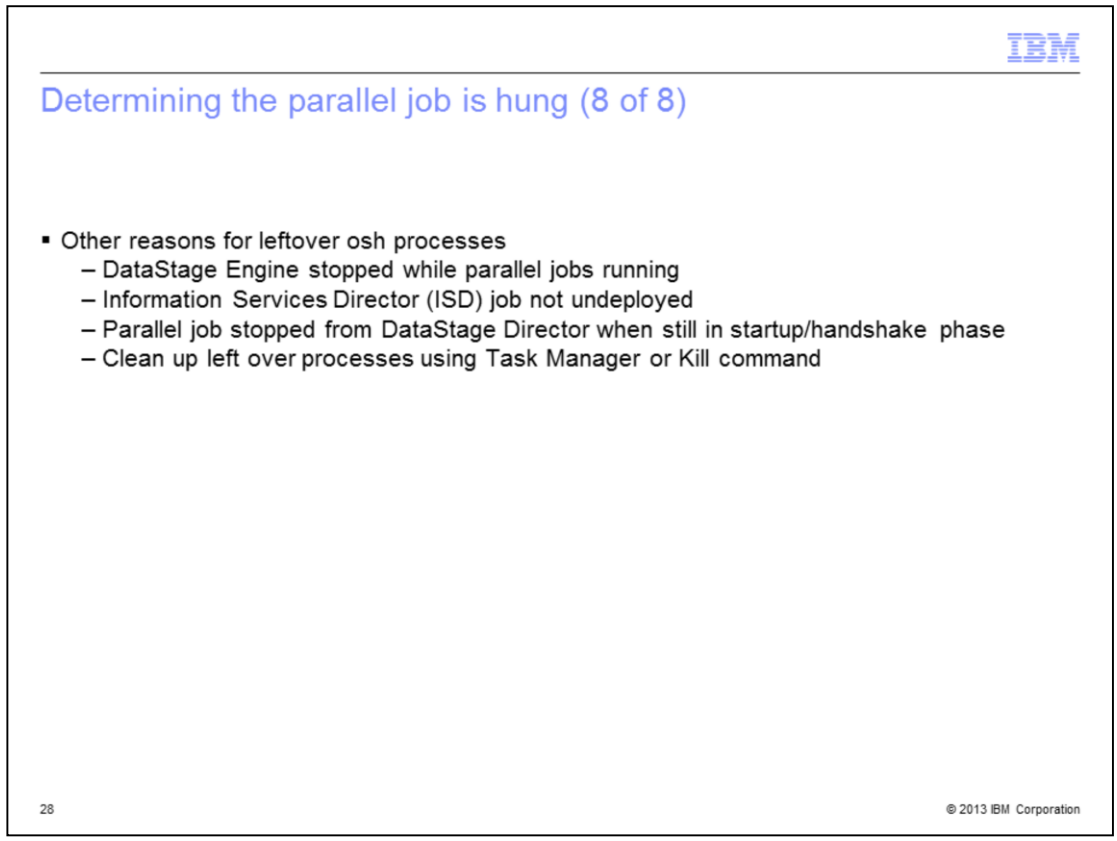


- To find the PID
  - Right click on the stage instance and select Detail
  - PID is listed under User
  - Repeat this for each instance
- Use Task Manager on Windows or the “ps” command on Unix/Linux to see the processes



27

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 Task Manager on Windows or the “ps” command on Unix/Linus to see the processes.



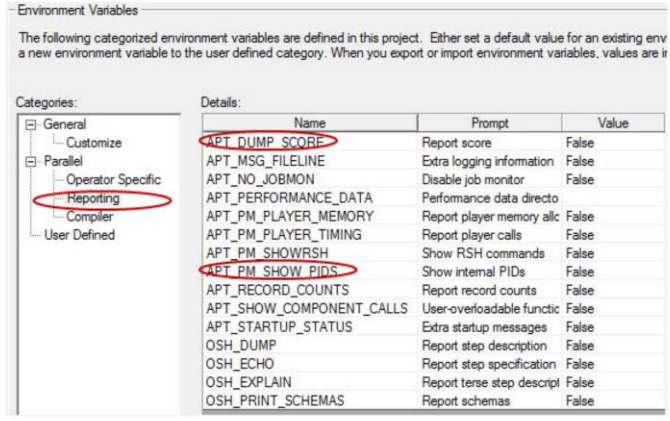
There are other reasons that leftover osh processes may be seen even though the job is not hung. One example is when the DataStage Engine is stopped while parallel jobs are running or an Information Service Director™ (ISD) job, is not undeployed before stopping the DataStage Engine. To prevent this check for running jobs and undeploy any ISD 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 using Task Manager on Windows or the kill command on Unix/Linux.

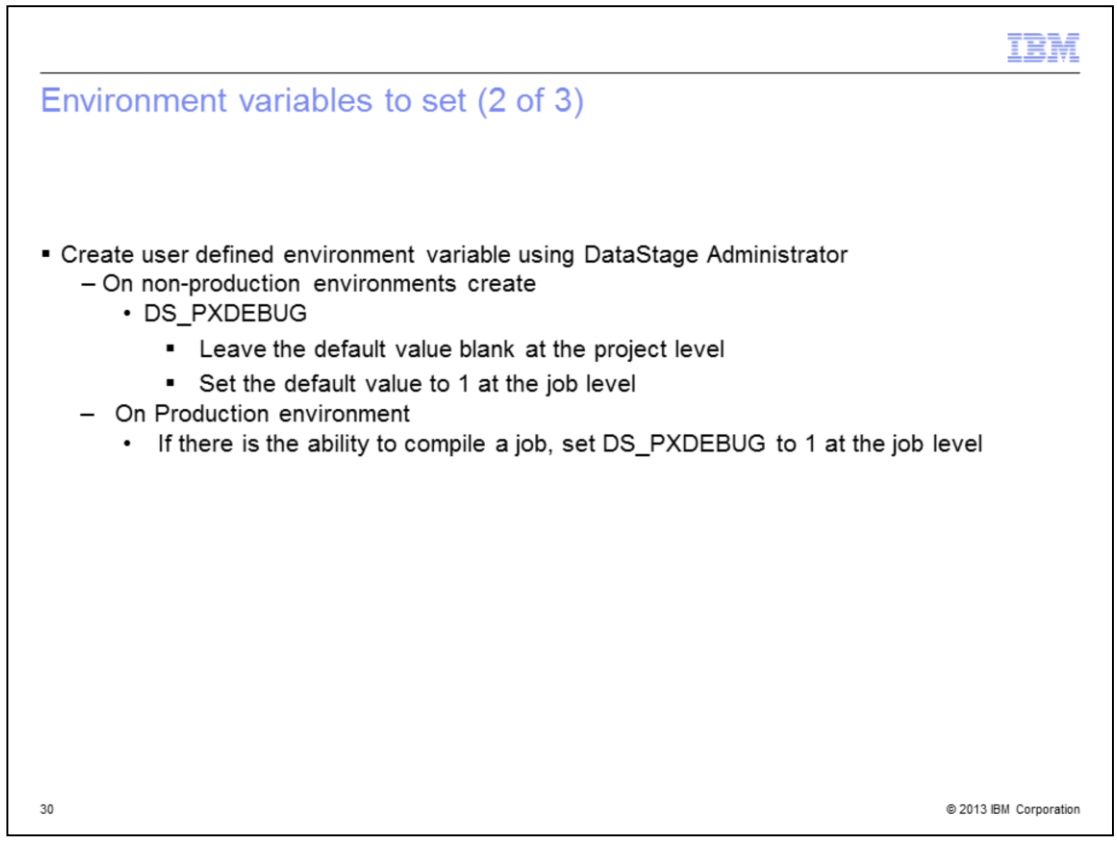
Environment variables to set (1 of 3)

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

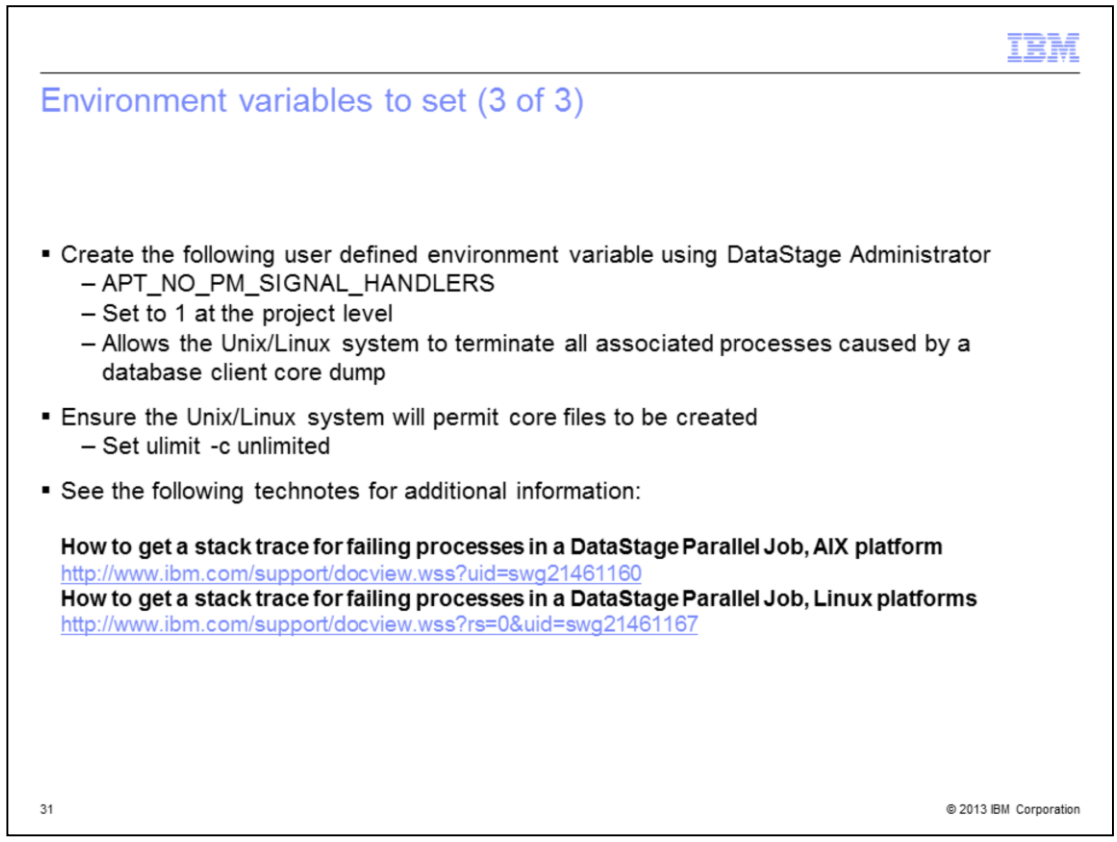


29

Once it is determined that the job is hung, setting the environment variables APT_PM_SHOW_PIDS and APT_DUMP_SCORE to true will provide information needed the next time the job hangs. These environment variables can be set at the job or project level.

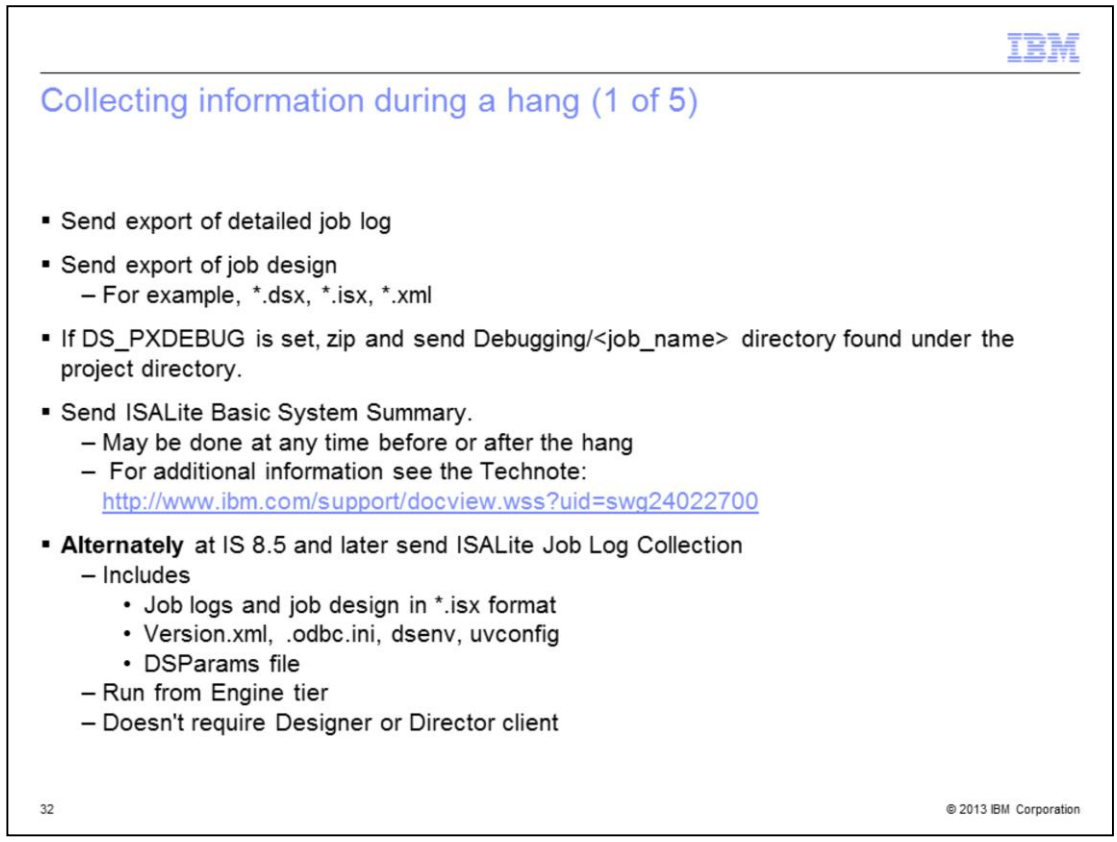


Next, on a non-production environment or a production environment with the ability to compile, create a user defined environment variable called DS_PXDEBUG. Set the default value to 1 at the job level. It is not recommended that DS_PXDEBUG be set at the project level because it will greatly impact the performance of jobs and add a lot of debug information to all of the job logs.



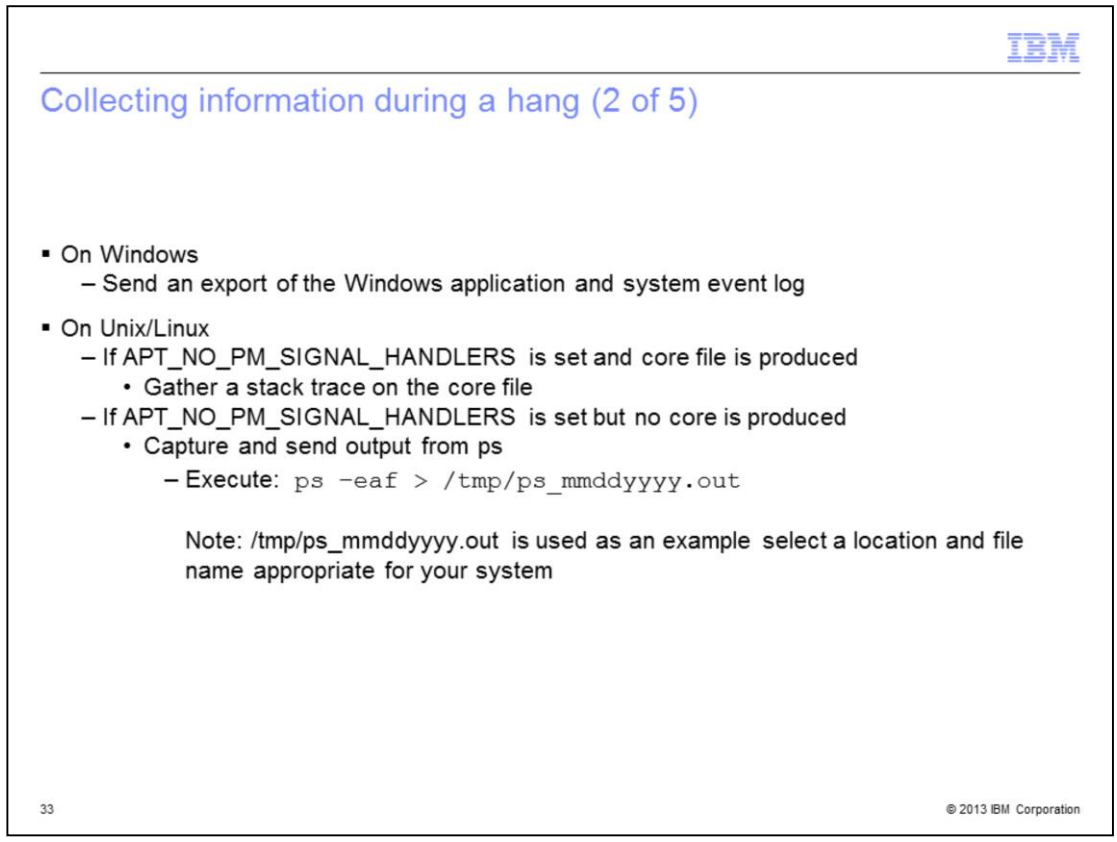
On Unix or Linux systems there is an additional environment variable. Many times a hang is caused when a database client core dumps. When this occurs, often the database operators or connectors will sit and wait forever for a response from the client that will never be sent due to the core dump and therefore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

If setting APT_NO_PM_SIGNAL_HANDLERS results in a core file being generated, ensure the system will permit 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.



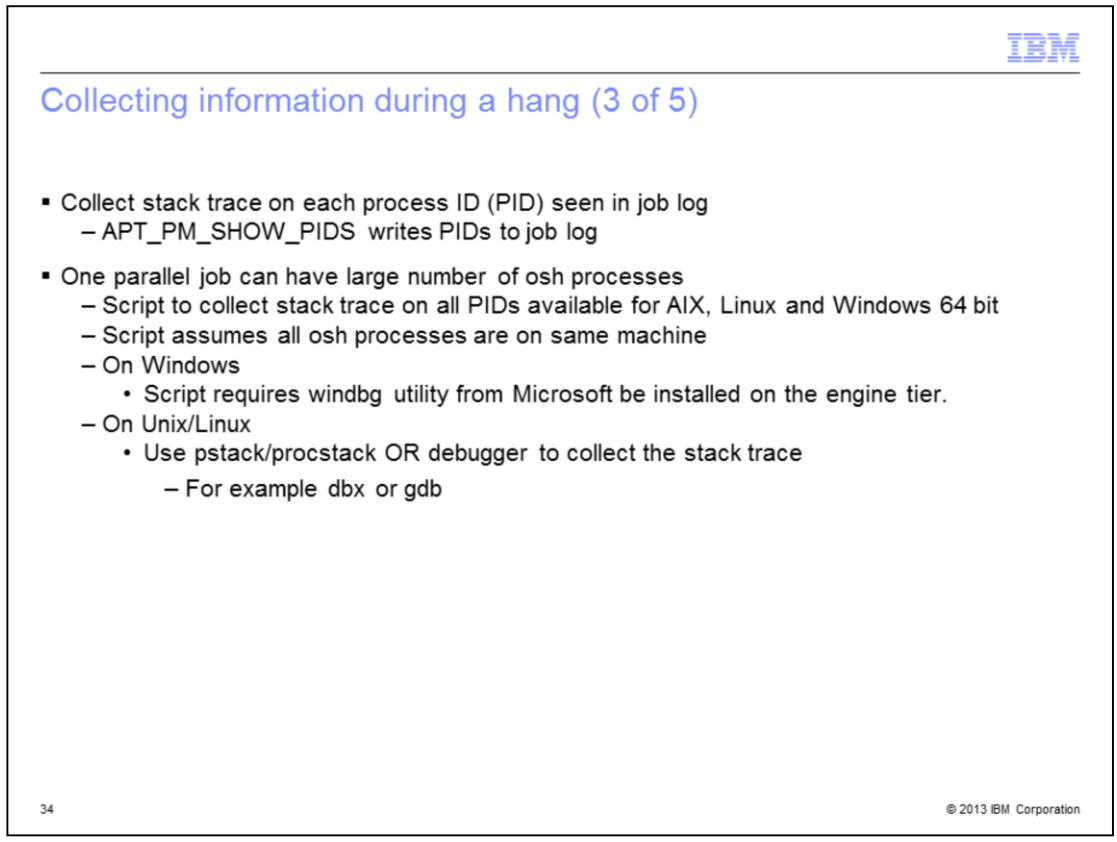
Once the environment variables have been set and the hang is reproduced, the next step is to collect the log information. It is important to send the detailed job log, and an export of the job design of the hanging job. The ISALite Basic System Summary can be done at any time before or after the hang.

Alternately at IS 8.5 and later send an ISALite Job Log Collection. This will collect all the job related information requested above 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.



There are a couple of operating dependent items. On Windows an export of the Windows application and system event log, and an ISALite Basic System Summary. The export of the application and system event logs need to include the timeframe during the job hang.

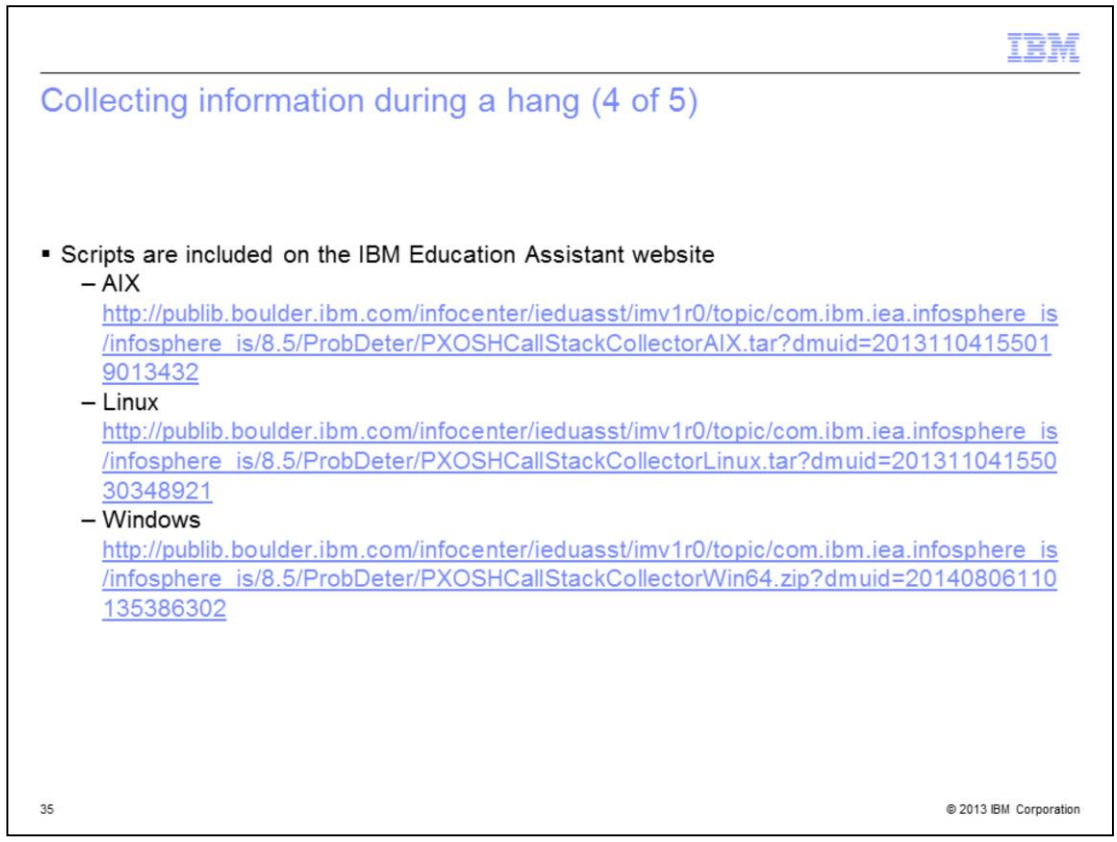
On Unix/Linux if APT_NO_PM_SIGNAL_HANDLERS is set in the job and a core file is produced, get a stack trace on the core file and send that along with the logs and job export. If the job does not end and core dump, run the ps command shown on this slide to capture all the system processes and send the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



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 that is 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, Linux and Windows 64 bit. The script assumes that all osh processes are on the same machine.

Each includes a README.txt with instructions on how to run the script.



The script is available for download from the IBM Education Assistant site. The file names are:

PXOSHCallStackCollectorAIX.tar

PXOSHCallStackCollectorLinux.tar

PXOSHCallStackCollectorWin64.zip

Collecting information during a hang (5 of 5)

- 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 ()
```


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

IBM

Questions

Additional Information - APARs

DataStage Parallel Engine

- JR40807: Terminator activity causes parallel jobs to hang instead of abort
  http://www.ibm.com/support/docview.wss?uid=swg1JR40807
  - Fix available in 8.5 Fix Pack 3
  - 8.7 Fix Pack 1
  - 9.1 GA and later
- JR47614: Fix problems with threaded tsort stage shutdown on windows
  http://www.ibm.com/support/docview.wss?uid=swg1JR47614
  - Supercedes: JR45272, JR44056
- JR39834: A job that reads from a data set and uses a copy stage to distribute the data to multiple funnel stages can hang
  http://www.ibm.com/support/docview.wss?uid=swg1JR39834
  - Fix available in 8.5 FP2
  - 8.7 GA and later
- JR49663: Job with surrogate key generator operator hangs when $APT_NO_JOBMON=1
  http://www.ibm.com/support/docview.wss?uid=swg1JR49663
  - Contact support for patch

IBM

Additional Information - APARs

Connectivity

- JR45325: Oracle OCI bulk load plugin may hang or crash
  http://www.ibm.com/support/docview.wss?uid=swg1JR45325
  - Fix available in 9.1.2
- JR47461: Oracle connector: uninitialized variable causes memory corruption that may cause job to hang or crash http://www.ibm.com/support/docview.wss?uid=swg1JR47461
  - Fix available in 11.3
- JR36567: Multiple orawrite stages in job corrupt work files and job fails
  http://www.ibm.com/support/docview.wss?uid=swg1JR36567
  - Fix available in 8.1 Fix Pack 2
  - 8.5 Fix Pack 1
  - 8.7 and later
- JR37099: Length of CLOB data has to be greater than 2 bytes in order to process data (DRS Oracle) http://www.ibm.com/support/docview.wss?uid=swg1JR37099
  - Fix available in 8.5 Fix Pack 1
  - 8.7 and later

Additional Information - APARs

Connectivity Cont'd

- JR39892: Information Server 8.5 parallel jobs that use the Netezza stage show a status of Running even though they have finished. (Windows)
  http://www.ibm.com/support/docview.wss?uid=swg1JR39892
  - Fix available in 8.5 FP2
  - 8.7 and later
- JR44676: Netezza connector server canvas jobs hang intermittently on windows platform when they process zero rows. (Windows)
  http://www.ibm.com/support/docview.wss?rs=14&uid=swg1JR44676
  - Fix available in 9.1.2
- JR48857: Provision to set load unload timeout in Netezza Connector
  http://www.ibm.com/support/docview.wss?uid=swg1JR48857
  - Fix available in 11.3 contact IBM Support
- JR44156: Netezza connect jobs hang intermittently trying to open the pipe second time
  http://www.ibm.com/support/docview.wss?rs=14&uid=swg1JR44156
  - Fix available in 8.7 Fix Pack 2
  - 9.1 FP1

Additional Information - APARs

Connectivity Cont'd

- JR44156: Netezza connect jobs hang intermittently trying to open the pipe second time
  http://www.ibm.com/support/docview.wss?rs=14&uid=swg1JR44156
  - Fix available in 8.7 Fix Pack 2
  - 9.1 FP1
- JR37958: Job does not finish, if non-existing table name is specified to "exception table name" in DB2 connector
  http://www.ibm.com/support/docview.wss?rs=14&uid=swg1JR37958
  - Fix available in 8.5 Fix Pack 1
  - 8.7 and later
- JR46530: DB2 connector in db2 for z/os (a) hangs/slow loading partitioned table, (b) does not read partitioned table in parallel
  http://www.ibm.com/support/docview.wss?uid=swg1JR46530
  - Fix available in 9.1.2
- JR50856: Greenplum connector parallel unload intermittently hangs
  http://www.ibm.com/support/docview.wss?uid=swg1JR50856
  - Patch available for 11.3 contact IBM Support

Additional Information - Technotes

Design Issues

- InfoSphere DataStage Parallel Job hangs when updating a table using a multi node configuration
  http://www.ibm.com/support/docview.wss?uid=swg21430643
- DataStage Job hangs while performing Oracle Update (Upsert) in Parallel
  http://www.ibm.com/support/docview.wss?uid=swg21430589
- DataStage Job with MQ stage hangs after end-of-data message put on queue
  http://www.ibm.com/support/docview.wss?uid=swg21600044

Additional Information - Technotes

DataStage Parallel Engine

- Information Server DataStage Jobs intermittently Hang on GRID
  http://www.ibm.com/support/docview.wss?uid=swg21590910
- DataStage sequence jobs hang with no error when using Grid Toolkit
  http://www.ibm.com/support/docview.wss?uid=swg21654563
- DataStage jobs hung in a project in IBM InfoSphere DataStage
  http://www.ibm.com/support/docview.wss?uid=swg21390366
- DataStage jobs running SQL Server Enterprise stage in parallel on multiple nodes may hang during bulk write if primary key for table defined
  http://www.ibm.com/support/docview.wss?uid=swg21504442

Additional Information - Technotes

Windows

- Teradata connector bulk access jobs fail using Teradata Client v13.10 on Windows
  http://www.ibm.com/support/docview.wss?uid=swg21567580
- Information Server DataStage Parallel Jobs on Windows are hanging or showing faults with KERNELBASE.dll in Event Log
  http://www.ibm.com/support/docview.wss?uid=swg21567108
- DataStage Parallel Jobs on Windows fail to start or show random failures
  http://www.ibm.com/support/docview.wss?uid=swg21669421
- General recommendations to run InfoSphere Information Server DataStage parallel jobs on Windows Platforms
  http://www.ibm.com/support/docview.wss?uid=swg21684610

Additional Information – Use debugger to collect stack trace (1 of 3)

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

45

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 listed above. Then confirm "which osh" returns the expected location.

Additional Information – Use debugger to collect stack trace (2 of 3)

- dbx example

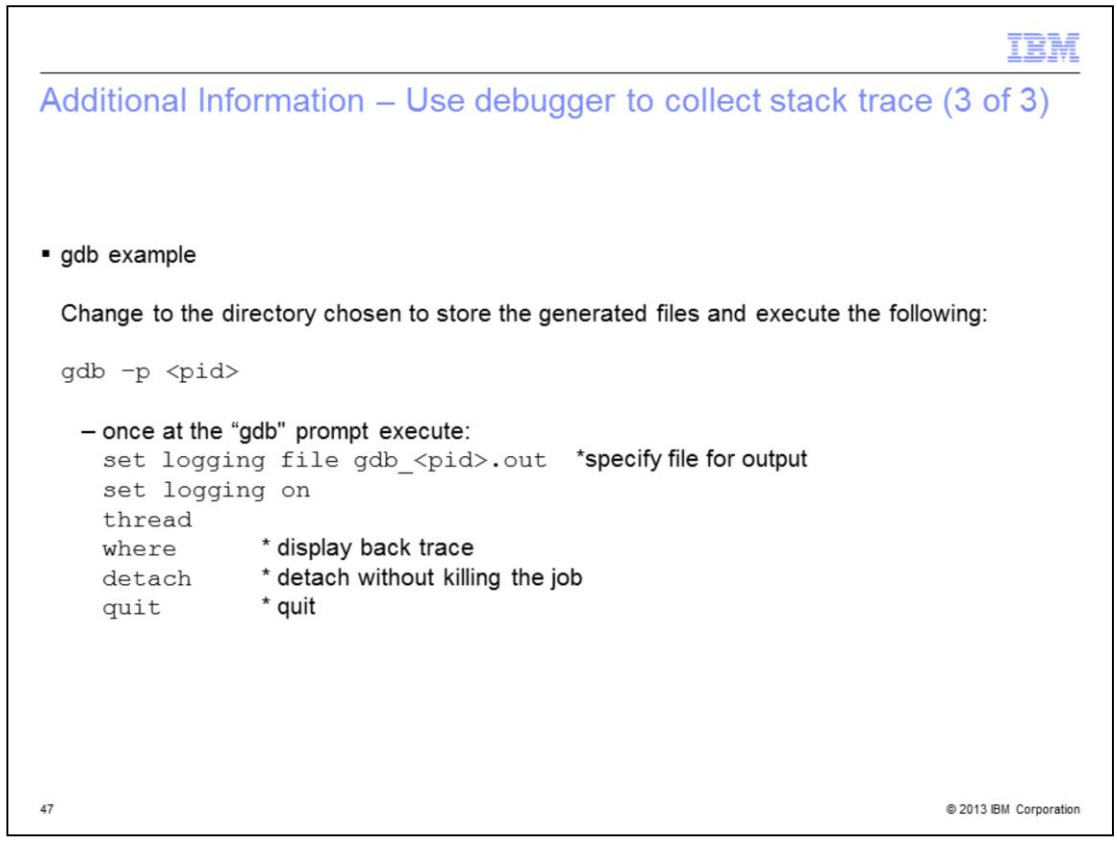
  Change to the directory chosen to store the generated files and execute the following:

```
dbx -a <pid> $APT_ORCHHOME/bin/osh
```

  – once at the "dbx" prompt run the following commands:

```
thread
where > dbx_<pid>.out      * Results will be displayed and sent to the file
detach      * Use detach to exit the dbx command shell without stopping the job
```


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.



This slide has 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.