

---

# Support Technical Exchange: TPM - TSAM - ISDM Workflow Troubleshooting

Scott Berens, TPM Level 2 Support Engineer

[berenss@us.ibm.com](mailto:berenss@us.ibm.com)

07 December 2012



# Introduction

- **Abstract:**
  - Tivoli Provisioning Manager (TPM) workflows can be difficult to troubleshoot. With some simple approaches you will be able to quickly find the error and take steps towards resolution.
  
- **Objectives:**
  - Learn how to export the workflow log XML file (UI and CLI).
  - Use appropriate tools to view the XML data file.
  - Understand the significant elements of the XML.
  - Compare the workflow log XML output to the source workflow.
  - Understand the java stack trace.
  
- The intended audience is targeted at technical professionals responsible for deploying and supporting Tivoli products at customer or Business Partner locations.

# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- Reference Materials
- Questions

# Obtain the workflow log XML data - UI

- From TPM UI, Start Center → Provisioning Workflow Status

The screenshot shows the 'Provisioning Workflow Status' page in the TPM UI. The page title is 'Provisioning Workflow Status' and it includes navigation links like 'Bullets: (0)', 'Go To', 'Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below the navigation bar, there is a search bar with 'Find:' and a 'Select Action' dropdown. The main content area has tabs for 'List' and 'Execution Logs'. There are also options for 'Advanced Search', 'Save Query', and 'Bookmarks'. The primary data is presented in a table titled 'Deployment Requests' with columns for 'Deployment Request', 'Workflow Name', 'Submit Date', and 'Status'. The table shows several rows of data, with the row for '13.720 MyFirstWorkflow' highlighted in yellow.

Deployment Request	Workflow Name	Submit Date	Status
14.000	No_operation	12/4/12 1:48 PM	Success
13.780	No_operation	11/29/12 5:24 PM	Success
13.762	No_operation	11/28/12 3:24 PM	Success
13.761	test123	11/28/12 1:49 PM	Success
13.760	test123	11/28/12 1:48 PM	Success
13.720	MyFirstWorkflow	11/28/12 12:37 PM	Success
13.740	No_operation	11/28/12 12:28 PM	Success

# Obtain the workflow log XML data - UI

- Select the Deployment Request ID from the list and when viewing the execution logs, Select Action → Export:

The screenshot shows the 'Provisioning Workflow Status' interface. At the top, there are navigation links: 'Bulletins: (0)', 'Go To', 'Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below this is a search bar and a 'Select Action' dropdown menu. The 'Select Action' menu is open, showing options: 'Refresh Page', 'Add to Bookmarks', 'Delete Workflow Execution', 'Stop Execution', 'Force Cancellation', 'Export' (highlighted in yellow), and 'Run Workflow Again'. A red 'no' symbol is overlaid on the 'Download' button to the right of the menu.

The main content area shows 'Workflow Name: MyFirstWorkflow' and 'Status: Success'. Below this is a 'Parameters' section with a 'Filter' dropdown and a table with columns 'Parameters', 'Input Values', and 'Output Values'. The table has one row with 'Number' as the parameter and '1' as the input value. The result is 'This number is less than 10.'.

At the bottom, there is an 'Execution Logs' section with a 'Filter' dropdown and a table with columns 'Submit Date', 'Seconds.Milliseconds', 'Text', and 'Log Level'. The table has two rows of logs:

Submit Date	Seconds.Milliseconds	Text	Log Level
11/28/12 12:37 PM	28.773	Start workflow: 'MyFirstWorkflow'	debug
11/28/12 12:37 PM	28.980	End workflow: 'MyFirstWorkflow'	debug

## Obtain the workflow log XML data - UI

- Your browser should allow pop-ups from the TPM server site.
- Any warning messages which restrict the download of the XML file should be accepted.
- If you have a problem try exporting from a different browser.
- Browser security settings may interfere with your ability to receive the pop-up and save file options.
- Once you have saved the XML file you can open it for review.
- Details from the online documentation:
  - [http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf\\_hisexp.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_hisexp.html)

## Obtain the workflow log XML data - CLI

- Scenarios in which CLI method may be preferred:
  - You have a large quantity of workflow log export XML data to obtain.
  - You want to capture ALL XML logs of a specific workflow.
  - You want the export to contain decrypted information.
  - You have very long (large) workflow logs that need to be exported.
- As `tioadmin`, from the `TIO_HOME` directory:
  - `workflowLogExport.cmd|.sh`
    - n `workflow_name` - The name of the provisioning workflow. One of -n, -r or -i is needed.
    - r `request_id` - The ID generated by the deployment engine specific to the execution of the provisioning workflow.
    - i `input_file_name` - The file must contain provisioning workflow names and each provisioning workflow must be on its own line.
    - f `output_file_name` - The name of the output file. The default output file is `workflowLogExport.xml`.
    - `[-d username password]` - Optional. This parameter turns on decryption for all the commands run by the `Device.ExecuteCommand` workflow. Only members of the Tivoli® Provisioning Manager security group can use this parameter. Use the appropriate username and password combination.
- Details from the online documentation:
  - [http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf\\_hisexpall.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_hisexpall.html)
  - [http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.support.tpm.doc/commands/rcmd\\_workflowLogExport.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.support.tpm.doc/commands/rcmd_workflowLogExport.html)

# Obtain the workflow log XML data - CLI

## ■ Example 1:

- `./workflowLogExport.sh -n Tpmfospd_Hardware_Discovery`  
2012-12-04 15:21:51,874 INFO COPDEX175I The workflow logs are extracted in:  
"/usr/ibm/tivoli/common/COP/logs/workflowLogExport.xml".  
Setting 600 permissions on /usr/ibm/tivoli/common/COP/logs/workflowLogExport.xml
- Review of the output file indicates 2 separate deployment requests have been stored for the "Tpmfospd\_Hardware\_Discovery" workflow:

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow-execution-history>
  <workflow id="3589" name="Tpmfospd_Hardware_Discovery">
    <deployment-request id="13686">
    </deployment-request>
    <deployment-request id="13685">
    </deployment-request>
  </workflow>
</workflow-execution-history>
```



# Obtain the workflow log XML data - CLI

## ■ Example 2:

- `./workflowLogExport.sh ./workflowLogExport.sh -n Device.ExecuteCommand -d maxadmin ibmtpm71`  
 2012-12-04 15:34:07,667 INFO COPDEX175I The workflow logs are extracted in:  
`"/usr/ibm/tivoli/common/COP/logs/workflowLogExport.xml"`.  
 Setting 600 permissions on `/usr/ibm/tivoli/common/COP/logs/workflowLogExport.xml`
- Review of the output file indicates 12 separate deployment requests have been stored for the "Device.ExecuteCommand" workflow
- With the "-d username password" specified we can see the command being executed:
 

```
<workflow id="3801" name="Device.ExecuteCommand">
  <deployment-request id="13609">
    <execution-log workflow-name="Device.ExecuteCommand" id="11228" date="Aug 16, 2012 2:32:35 PM"
    position="108" call-stack-level="0" log-text="Start workflow:
    'Device$ExecuteCommand'">
      <log-details position="0" name="DeviceId">11300</log-details>
      <log-details position="1" name="ExecuteCommand">./TSM.exe</log-details>
      <log-details position="2" name="WorkingDirectory">c:\92980</log-details>
      <log-details position="3" name="CredentialsKey" />
      <log-details position="4" name="TimeoutInSeconds" />
      <log-details position="5" name="TreatTimeoutAs" />
    </execution-log>
```

# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- Reference Materials
- Questions

## Review the workflow log XML data

- Any XML editor, text editor and even web browser will work.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<workflow-execution-history>
  <deployment-request id="13720" workflow-name="MyFirstWorkflow" create-username="MAXADMIN" status="success">
    <execution-log workflow-name="MyFirstWorkflow" id="11480" date="Nov 28, 2012 12:37:28 PM" position="0" call-stack-level="0" log-text="Start workflow:
    &apos;MyFirstWorkflow&apos;">...</execution-log>
    <execution-log workflow-name="MyFirstWorkflow" id="11481" date="Nov 28, 2012 12:37:28 PM" position="1" call-stack-level="0" log-text="End workflow:
    &apos;MyFirstWorkflow&apos;">...</execution-log>
  </deployment-request>
</workflow-execution-history>

```

- List of preferred editors which all use Syntax Highlighting:
  - Notepad++, Visual SlickEdit, UltraEdit, Gedit, Vim, XML Marker
  - XML Notepad:** <http://www.microsoft.com/en-us/download/details.aspx?id=7973>

```

<?xml version="1.0" encoding="UTF-8"?>
<workflow-execution-history>
  <deployment-request id="13720" workflow-name="MyFirstWorkflow" create-username="MAXADMIN" status="success">
    <execution-log workflow-name="MyFirstWorkflow" id="11480" date="Nov 28, 2012 12:37:28 PM" position="0" call-stack-level="0" log-text="Start workflow:
    &apos;MyFirstWorkflow&apos;">
      <log-details position="0" name="Number">1</log-details>
    </execution-log>
    <execution-log workflow-name="MyFirstWorkflow" id="11481" date="Nov 28, 2012 12:37:28 PM" position="1" call-stack-level="0" log-text="End workflow:
    &apos;MyFirstWorkflow&apos;">
      <log-details position="0" name="Result">This number is less than 10.</log-details>
    </execution-log>
  </deployment-request>
</workflow-execution-history>

```

## Review the workflow log XML data

- Most browsers and editors which handle Syntax Highlighting can also handle the proper conversion of ASCII and HTML character encoding.
- If you find additional HTML character codes which are disruptive to your review, use find/replace:
  - `&#xA;` → carriage return
  - `&#x9;` → line feed
  - `&amp;` → ampersand &
  - `&apos;` → apostrophe '
  - `&quot;` → double quotes “

# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- Reference Materials
- Questions

# Understand the workflow log XML data

```

<workflow-execution-history>
  <deployment-request id="17862" error-details="SSH_Execute_Command72(line:44) SSH_Execute_Command(line:253)
    ServiceAccessPoint$ExecuteCommand(line:29) Default_Device_Execute_Command(line:30) Device$ExecuteCommand(line:29)
    NIM_Install_Clone_Image(line:1169) NIM_Install_BaseImage(line:15) BootServer$InstallGoldenMasterImage(line:148) Image_Install(line:21)
    Cloud_PPC_AIX_Install(line:25) SoftwareInstallable$Install(line:66) Cloud_OS_Module_Install(line:59) Cloud_Add_Server(line:70)
    Cluster$AddServer(line:48) RP$ClusterProvision(line:98) com.ibm.tivoli.ldo.runtime.scriptlet.ssh.SSHExecuteCommandException:
    COPCOM123E A shell command error occurred: Exit code=2, Error stream="", Output stream="prompting_for_data_at_console Timeout
    occured. Cstate_result = success Cstate = Base Operating System installation is being performed Mstate = in the process of booting". at
    com.ibm.tivoli.ldo.runtime.scriptlet.ssh.SshExec.executeCommand(SshExec.java:163) at
    com.ibm.tivoli.ldo.runtime.scriptlet.ssh.SshExec.exec(SshExec.java:189) at sun.reflect.GeneratedMethodAccessor5512.invoke(Unknown
    Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke
    (Method.java:618) at com.ibm.tivoli.tpm.common.reflect.ReflectionHelper.evaluate(ReflectionHelper.java:158) at
    com.ibm.tivoli.tpm.common.reflect.ReflectionHelper.evaluate(ReflectionHelper.java:126) at
    com.ibm.tivoli.orchestrator.de.util.ReflectionHelper.invoke(ReflectionHelper.java:60) at
    com.ibm.tivoli.tpm.wkf.core.SSH_Execute_Command72.execute(SSH_Execute_Command72.java:119) at
    com.ibm.tivoli.tpm.wkf.core.SSH_Execute_Command.execute(SSH_Execute_Command.java:119) at
    sun.reflect.GeneratedMethodAccessor5510.invoke(Unknown Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:618) at
    com.ibm.tivoli.ldo.runtime.ServiceBase.invoke(ServiceBase.java:90) at
    com.ibm.tivoli.tpm.wkf.core.ServiceAccessPoint$ExecuteCommand.execute(ServiceAccessPoint$ExecuteCommand.java:70) at
    com.ibm.tivoli.tpm.wkf.default_device_model.Default_Device_Execute_Command.execute(Default_Device_Execute_Command.java:92) at
    sun.reflect.GeneratedMethodAccessor5507.invoke(Unknown Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:618) at
    com.ibm.tivoli.ldo.runtime.ServiceBase.invoke(ServiceBase.java:90) at com.ibm.tivoli.tpm.wkf.core.Device$ExecuteCommand.execute
    (Device$ExecuteCommand.java:70) at com.ibm.tivoli.tpm.wkf.NIM.NIM_Install_Clone_Image.execute(NIM_Install_Clone_Image.java:2097)
    at com.ibm.tivoli.tpm.wkf.NIM.NIM_Install_BaseImage.execute(NIM_Install_BaseImage.java:55) at
    sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke
    (NativeMethodAccessorImpl.java:79) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at
    java.lang.reflect.Method.invoke(Method.java:618) at com.ibm.tivoli.ldo.runtime.ServiceBase.invoke(ServiceBase.java:90) at
    com.ibm.tivoli.tpm.wkf.core.BootServer$InstallGoldenMasterImage.execute(BootServer$InstallGoldenMasterImage.java:272) at
    com.ibm.tivoli.tpm.wkf.image.Image_Install.execute(Image_Install.java:61) at com.ibm.tivoli.tpm.wkf.Cloud.Cloud_PPC_AIX_Install.execute
    (Cloud_PPC_AIX_Install.java:73) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at
    sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79) at sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:618) at
    com.ibm.tivoli.ldo.runtime.ServiceBase.invoke(ServiceBase.java:90) at com.ibm.tivoli.tpm.wkf.core.SoftwareInstallable$Install.execute
    (SoftwareInstallable$Install.java:125) at com.ibm.tivoli.tpm.wkf.Cloud.Cloud_OS_Module_Install.execute
    (Cloud_OS_Module_Install.java:127) at com.ibm.tivoli.tpm.wkf.Cloud.Cloud_Add_Server.execute(Cloud_Add_Server.java:121) at
    sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke
    (NativeMethodAccessorImpl.java:79) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at
    java.lang.reflect.Method.invoke(Method.java:618) at com.ibm.tivoli.ldo.runtime.ServiceBase.invoke(ServiceBase.java:90) at
    com.ibm.tivoli.tpm.wkf.core.Cluster$AddServer.execute(Cluster$AddServer.java:93) at
    com.ibm.tivoli.tpm.wkf.Cloud.RP$ClusterProvision.execute(RP$ClusterProvision.java:208) at sun.reflect.NativeMethodAccessorImpl.invoke0
  
```

■ **WHAT IS ALL OF THIS ????????**

## Understand the workflow log XML data

- 3 quick steps to understand the workflow failure:
  - 1) call-stack-level
  - 2) error-message
  - 3) error stream
- Use the above search keys for quick navigation in the XML log file.
- For extra credit we will review a java stack trace exception and provide a simple approach to understanding a java failure without having to review java source code nor having any knowledge of java coding techniques.

## Understand the workflow log XML data – Call stack

- Call stack level (0,1,2,3,4,etc) indicates the depth of the current workflow as referenced from the parent workflow.
  - A trail of 'breadcrumbs', tracking deeper into sub-level workflows.
  - Provides an awareness of **where** you are at.
  - Example call stack level flow from a TSAM VM provision log:

```

workflow-name="RP$ClusterProvision" date="Nov 7, 2012 10:16:52 AM" call-stack-level="0"
workflow-name="RP$CreateVirtualServer" date="Nov 7, 2012 10:16:53 AM" call-stack-level="1"
workflow-name="CloudHostPlatform_PreNetworkConfiguration" date="Nov 7, 2012 10:16:53 AM" call-stack-level="2"
workflow-name="RP$Get_SubnetAndIPAddressFromPool" date="Nov 7, 2012 10:16:53 AM" call-stack-level="3"
workflow-name="RP.RM_Allocate_Ip_Address" date="Nov 7, 2012 10:16:53 AM" call-stack-level="4"
  
```

- After a sub-level workflow completes it will wrap up to it's parent workflow and call-stack-level will decrease (back towards 0).

```

workflow-name="RP.RM_Allocate_Ip_Address" date="Nov 7, 2012 10:16:54 AM" call-stack-level="4"
workflow-name="RP$Get_SubnetAndIPAddressFromPool" date="Nov 7, 2012 10:16:54 AM" call-stack-level="3"
.
.
.
  
```

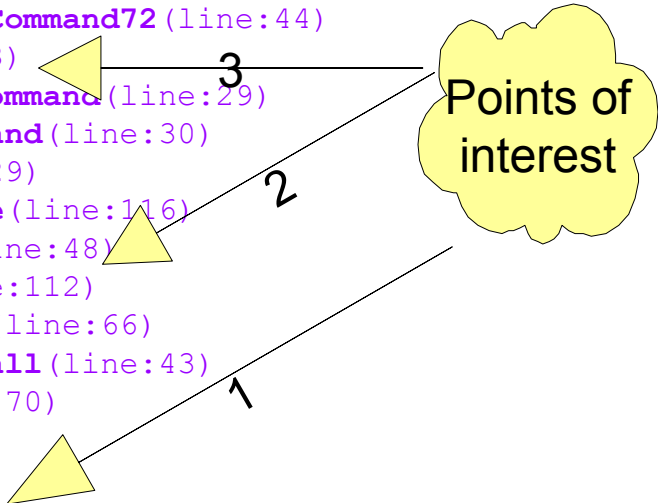


# Understand the workflow log XML data – Call stack

- When you consolidate a stack of workflow calls it could look quite complicated.
- Read from the bottom up to understand the progression (taken from error-details of the same TSAM provision failure):

```

error-details="SSH_Execute_Command72 (line:44)
SSH_Execute_Command (line:253)
ServiceAccessPoint$ExecuteCommand (line:29)
Default_Device_Execute_Command (line:30)
Device$ExecuteCommand (line:29)
Cloud_Repository_Mount_Share (line:116)
FileRepository$MountShare (line:48)
Cloud_ITM_Agent_Install (line:112)
SoftwareInstallable$Install (line:66)
Default_SoftwareModule_Install (line:43)
SoftwareModule$Install (line:70)
Cloud_Add_Server (line:142)
Cluster$AddServer (line:48)
RP$ClusterProvision (line:98)
com.ibm.tivoli.ldo.runtime.scriptlet.ssh.SSHExecuteCommandException: COPCOM123E A shell
command error occurred: Exit code=1, Error stream="mount: 1831-008 giving up on:
10.5.10.5:/repository
vmount: Operation not permitted.", Output stream="".
    
```



## Understand the workflow log XML data – Error-message

- In a failed workflow log, the error-message string will provide the quickest summary about what error has occurred.

```
error-message="COPCOM123E A shell command error occurred: Exit code=1, Error stream="
mount: 1831-008 giving up on: 10.5.10.5:/repository vmount: Operation not permitted.
";, Output stream=&""." workflow-name="RP.ClusterProvision" create-username="maxadmin"
error-code="COPCOM123EshellCommandError" status="failed">
```

- error-message="COPCOM123E" Error stream=  
"mount: 1831-008 giving up on:  
10.5.10.5:/repository  
vmount: Operation not permitted.",
- workflow-name="RP.ClusterProvision"
- create-username="maxadmin"
- error-code="COPCOM123EshellCommandError"
- status="failed"

## Understand the workflow log XML data – Error stream

- The “error stream” generally will provide a high level of information as to the cause of the error.
- Use it as a search key to pinpoint the exact section of log in which the error has occurred.
- Once you have found the exact error browse up and down (~20 lines) to understand what has been completed, what has just been called, and what occurred (if anything) after the error was thrown.
- Usually the error will be near the end of the log but it can be the case that an error will be caught by the workflow code (by design) and a post error or cleanup workflow will continue. Thus, the error might not always be at the end of the file.

# Understand the workflow log XML data – Error stream

- Example:

```

<execution-log workflow-name="Cloud_Repository_Mount_Share" id="168867" date="Nov 7, 2012
10:43:28 AM" position="57010" call-stack-level="8" log-text="Failed workflow:
&apos;Cloud_Repository_Mount_Share&apos;" />
<execution-log workflow-name="FileRepository.MountShare" id="168868" date="Nov 7, 2012 10:43:28
AM" position="57011" call-stack-level="7" log-text="Failed workflow:
&apos;FileRepository.MountShare&apos;" />
<execution-log workflow-name=
"com.thinkdynamics.kanaha.de.javaplugin.datacentermodel.UnlockDCMObject" id="168869" date="Nov 7,
2012 10:43:28 AM" position="57012" call-stack-level="6" log-text="Start JavaPlugin:
&apos;com.thinkdynamics.kanaha.de.javaplugin.datacentermodel.UnlockDCMObject&apos;" />
<execution-log workflow-name=
"com.thinkdynamics.kanaha.de.javaplugin.datacentermodel.UnlockDCMObject" id="168870" date="Nov 7,
2012 10:43:28 AM" position="57013" call-stack-level="6" log-text="End JavaPlugin:
&apos;com.thinkdynamics.kanaha.de.javaplugin.datacentermodel.UnlockDCMObject&apos;" />
<execution-log workflow-name="Cloud_ITM_Agent_Install" id="168871" date="Nov 7, 2012 10:43:28 AM"
position="57014" call-stack-level="6" log-text="Failed workflow:
&apos;Cloud_ITM_Agent_Install&apos;" />
<execution-log workflow-name="SoftwareInstallable$Install" id="168872" date="Nov 7, 2012 10:43:28
AM" position="57015" call-stack-level="5" log-text="" />
<execution-log workflow-name="SoftwareInstallable.Install" id="168873" date="Nov 7, 2012 10:43:28
AM" position="57016" call-stack-level="5" log-text="Failed workflow:
&apos;SoftwareInstallable.Install&apos;" />
<execution-log workflow-name="Default_SoftwareModule_Install" id="168874" date="Nov 7, 2012
10:43:28 AM" position="57017" call-stack-level="4" log-text="Failed workflow:
&apos;Default_SoftwareModule_Install&apos;" />
<execution-log workflow-name="SoftwareModule.Install" id="168875" date="Nov 7, 2012 10:43:28 AM"
position="57018" call-stack-level="3" log-text="Failed workflow:
&apos;SoftwareModule.Install&apos;" />
<execution-log workflow-name="Cloud_Add_Server" id="168876" date="Nov 7, 2012 10:43:28 AM"
position="57019" call-stack-level="2" log-text="Error Installing IBM Tivoli Monitoring Agent" />
<execution-log workflow-name="Cloud_Add_Server" id="168877" date="Nov 7, 2012 10:43:28 AM"
position="57020" call-stack-level="2" log-text="COPCOM123E A shell command error occurred: Exit
code=1, Error stream=&quot;mount: 1831-008 giving up
n: &#xA;10.5.10.5;/repository&#xA;vmount: Operation not permitted.&#xA;&quot;;, Output
stream=&quot;&quot;.&#xA;&lt;tpm_returncode&gt;1&lt;/tpm_returncode&gt;&#xA;
&lt;tpm_returnerrorstring&gt;mount: 1831-008 giving up
on:&#xA;10.5.10.5;/repository&#xA;vmount: Operation not
permitted.&#xA;&lt;/tpm_returnerrorstring&gt;&#xA;&lt;tpm_returnresult&gt;&lt;/
tpm_returnresult&gt;" />

```

Points of interest

## Understand the workflow log XML data – Java stack Trace

- “Reading” a java stack trace, move from the bottom up.
- Check the java paths and look for method names as a clue.

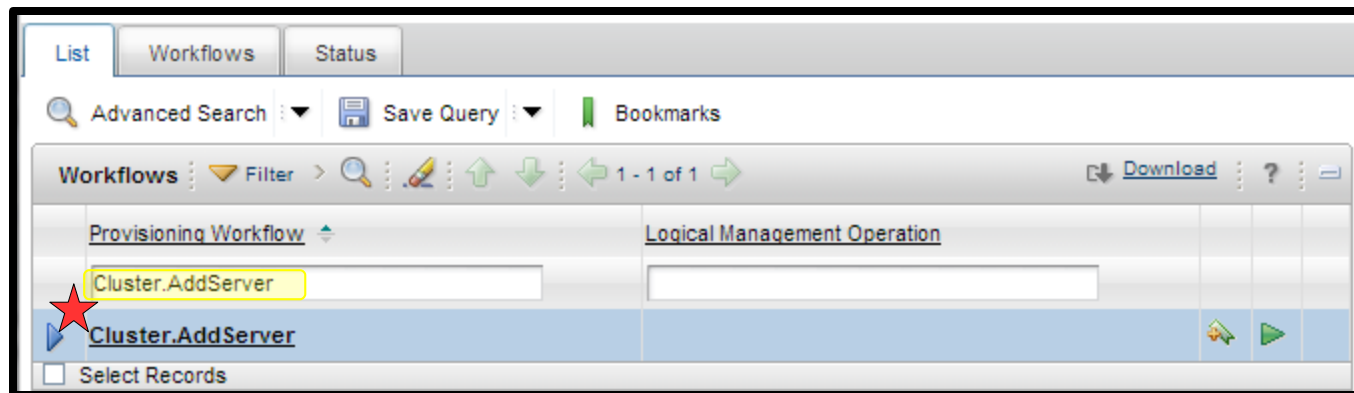
```
at java.net.SocketInputStream.socketRead0(Native Method)
...
at com.ibm.db2.jcc.t4.z.b(z.java:199)
...
at com.ibm.db2.jcc.am.jn.executeQuery(jn.java:663)
...
at psdi.mbo.MboSet.fetchNext(MboSet.java:2515)
...
at com.ibm.tivoli.orchestrator.de.dto.WorkflowExecutionLog.delete(WorkflowExecutionLog.java:208)
...
at com.ibm.tivoli.orchestrator.de.dto.DeploymentRequest.delete(DeploymentRequest.java:266)
at com.ibm.tivoli.orchestrator.de.dto.maximo.DeploymentRequestMbo.domainObjectDelete(DeploymentRequestMbo.java:46)
at com.ibm.tivoli.tpm.maximo.mbo.AbstractBaseDomainObjectMboSet.deleteMbo(AbstractBaseDomainObjectMboSet.java:124)
at psdi.mbo.MboSet.saveTransaction(MboSet.java:6849)
...
at psdi.mbo.MboSet.save(MboSet.java:6682)
at com.ibm.tivoli.tpm.maximo.mbo.AbstractBaseDomainObjectMboSet.save(AbstractBaseDomainObjectMboSet.java:284)
at psdi.mbo.MboSet.save(MboSet.java:6623)
at psdi.webclient.system.beans.ResultsBean.save(ResultsBean.java:129)
at com.ibm.tivoli.tpm.automation.DeploymentRequestAppBean.DELETE(DeploymentRequestAppBean.java:120)
...
at psdi.webclient.system.session.WebClientSession.handleRequest(WebClientSession.java:700)
...
```

# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- Reference Materials
- Questions

## Compare workflow code against workflow log

- Workflows can be easily viewed from the TPM UI, by accessing Start Center → Provisioning Workflows.
- TPM UI is also referred to as the “maximo ui, TSAM Admin console, and TPAE UI.”
- Viewing the workflow log export XML file you might be interested in the actions that a particular workflow is taking.
- Search the workflow name:



## Compare workflow code against workflow log

- Open the workflow in the browser and compare to your workflow log XML.
- Search for log info statements in the workflow which can be used as trace-points when reviewing the workflow log XML.

pSeries\_ExecuteCommandOnHMC.wkf

```
var locale = Java[PseriesHelper#getProvisioningServerLocale()]
var localePrefix = "export LANG=" + locale + "; "

ExecuteCommand = localePrefix + sdmcPrefix + ExecuteCommand
log debug "HMC command: " + ExecuteCommand

Device.ExecuteCommand(DeviceId, ExecuteCommand, WorkingDirectory,
```

pSeries\_ExecuteCommandOnHMC-workflowLog.XML

```
<execution-log workflow-name="pSeries_ExecuteCommandOnHMC" id="167493"
date="Nov 7, 2012 10:18:22 AM" position="55644" call-stack-level="6"
log-text="HMC command: export LANG=en_US.UTF-8; viosvr cmd -m
&quot;cdwrp710&quot; -p psvio -c &quot;lsvg rootvg -field
freepps -fmt , &quot;" />
```



## Compare workflow code against workflow log

- Open the workflow in your preferred editor or the APDE and compare the code against the workflow log XML.

test123 workflow in APDE

```
test123.wkf X
workflow test123 (in Number, out Result) LocaleInsensitive
#invoke jython in a conditional statement to evaluate input
  if Jython[int(Number) < 10] then
    Result = "This number is less than 10."
  else
    Result = "This number is greater than 10."
  endif
```

test123 workflow log XML in notepad++

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow-execution-history>
  <deployment-request id="13761" workflow-name="test123" status="success">
    <execution-log workflow-name="test123" id="11486" date="Nov 28, 2012
1:50:34 PM" position="6" call-stack-level="0" log-text="Start
workflow: 'test123'">
      <log-details position="0" name="Number">5</log-details>
    </execution-log>
    <execution-log workflow-name="test123" id="11487" date="Nov 28, 2012
1:50:34 PM" position="7" call-stack-level="0" log-text="End workflow:
'test123'">
      <log-details position="0" name="Result">This number is less than 10.
    </log-details>
    </execution-log>
  </deployment-request>
</workflow-execution-history>
```

## Compare workflow code against workflow log

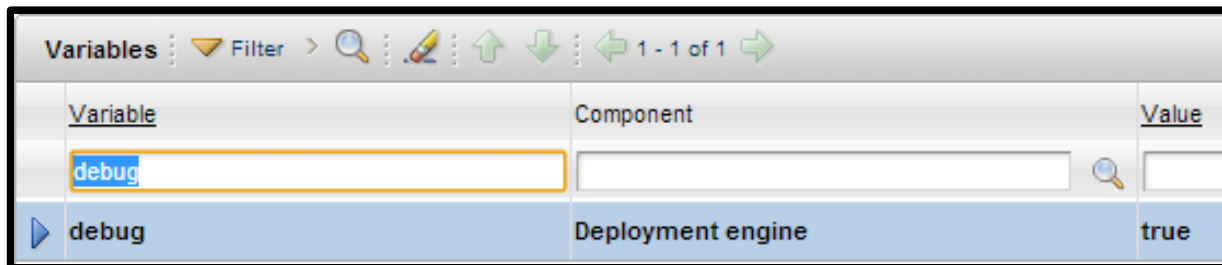
- One important note about workflow code:
  - Most of the time it is relatively easy to read the jython code and understand what is happening.
  - You will encounter situations when a java method gets called directly.
  - When java gets called you may not find much additional java logging inside the workflow log XML.
  - This is because the java code is being run directly in the TPM engines and the log output will be stored in the TIO\_LOGS/console.log.
  - Correlate time and date stamps from the workflow log XML to understand what the java code is outputting into the console.log.

# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- Reference Materials
- Questions

## Troubleshooting Tips and Techniques

- Workflow debug logging can help to show more information about the function of the workflow.
- Default logging at Info level will show only the highest level of information available.
  - 1) Go To → Administration → Provisioning → Provisioning Global Settings.
  - 2) Click the Variables tab.
  - 3) Click New Row..
  - 4) Name the variable debug with component Deployment engine, and value true.



Variable	Component	Value
debug		
▶ debug	Deployment engine	true

- [http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/wkf\\_troubleshooting/trwkwf\\_trblogs.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/wkf_troubleshooting/trwkwf_trblogs.html)

# Troubleshooting Tips and Techniques

- **Make the workflow editable.**

- 1) Log on to the provisioning server as tioadmin.
- 2) Open the DB2 command window (db2 / db2cmd).
- 3) Connect to the TPM database

```
db2 connect to maxdb71 user maximo using PaSsWoRd.
```

- 4) Enter the following command for your 'workflow\_name':

```
update workflow4 set IS_EDITABLE='Y' WHERE WORKFLOW_NAME='workflow_name'  
DB20000I The SQL command completed successfully.
```

– **Note:** You do not need to restart TPM.

- **You can now modify and re-compile the workflow to test a certain command or logging statement.**

- [http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf\\_cmpview.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v45r1/topic/com.ibm.tivoli.tpm.wkf.doc/workflows/twkf_cmpview.html)

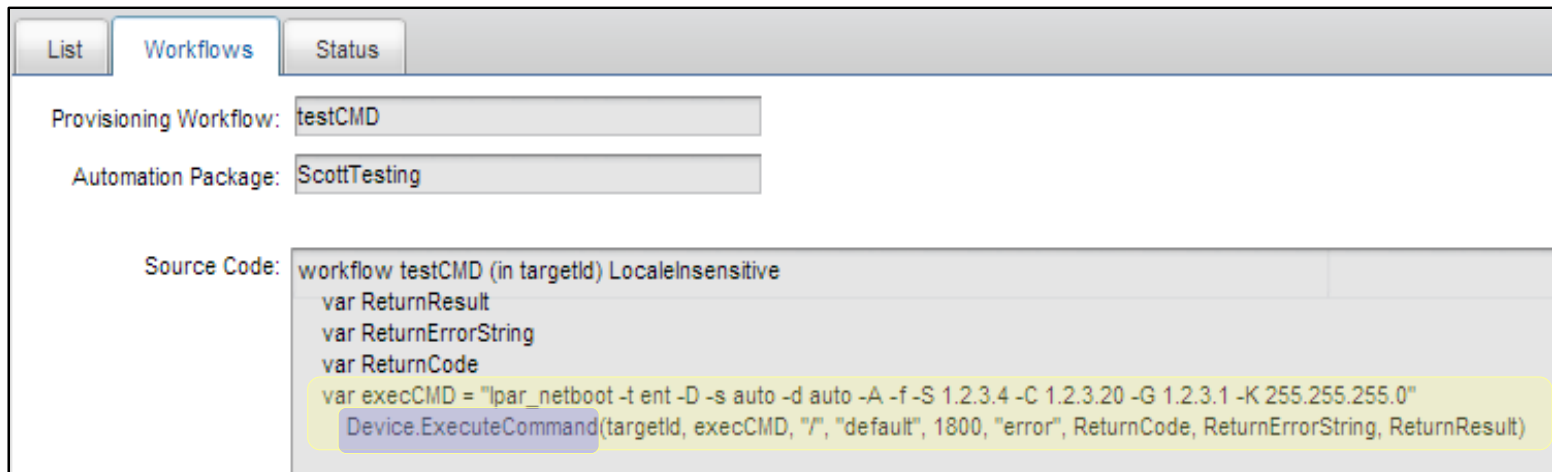
# Troubleshooting Tips and Techniques

- Lets say you find that a particular Device.ExecuteCommand is failing against a target system:

```
<execution-log workflow-name="Device.ExecuteCommand" id="168121" date="Nov 7, 2012 10:21:04 AM" position="56276"
call-stack-level="10" log-text="End workflow: &apos;Device.ExecuteCommand&apos;">
  <log-details position="0" name="ReturnCode">0</log-details>
  <log-details position="1" name="ReturnErrorString" />
  <log-details position="2" name="ReturnResult">lpar_netboot: Error : Close command sent/bin/stty: standard input:
  Invalid argument
# Connecting to pchmura0
# Connected
# Checking for power off.
# Power off complete.
# Power on pchmura0 to Open Firmware.
# Power on complete.
# Client IP address is 10.2.10.20.
# Server IP address is 10.2.10.12.
# Gateway IP address is 10.2.10.1.
# Subnetmask IP address is 255.255.255.0.
# Getting adapter location codes.
# /vdevice/l-lan@30000002 ping successful.
# Network booting install adapter.
# bootp sent over network.
# Network boot proceeding, lpar_netboot is exiting.
# Finished.</log-details>
```

# Troubleshooting Tips and Techniques

- continued...
- Rather than re-run the failing workflow to get the same point of failure, create your own test workflow and run the `Device.ExecuteCommand` directly against the target.



The screenshot shows a web interface with three tabs: "List", "Workflows", and "Status". The "Workflows" tab is active. Below the tabs, there are two input fields: "Provisioning Workflow:" with the value "testCMD" and "Automation Package:" with the value "ScottTesting". Below these fields is a "Source Code:" section containing the following code:

```
workflow testCMD (in targetId) LocaleInSensitive
var ReturnResult
var ReturnErrorString
var ReturnCode
var execCMD = "lpar_netboot -t ent -D -s auto -d auto -A -f -S 1.2.3.4 -C 1.2.3.20 -G 1.2.3.1 -K 255.255.255.0"
Device.ExecuteCommand(targetId, execCMD, "/", "default", 1800, "error", ReturnCode, ReturnErrorString, ReturnResult)
```

## Troubleshooting Tips and Techniques

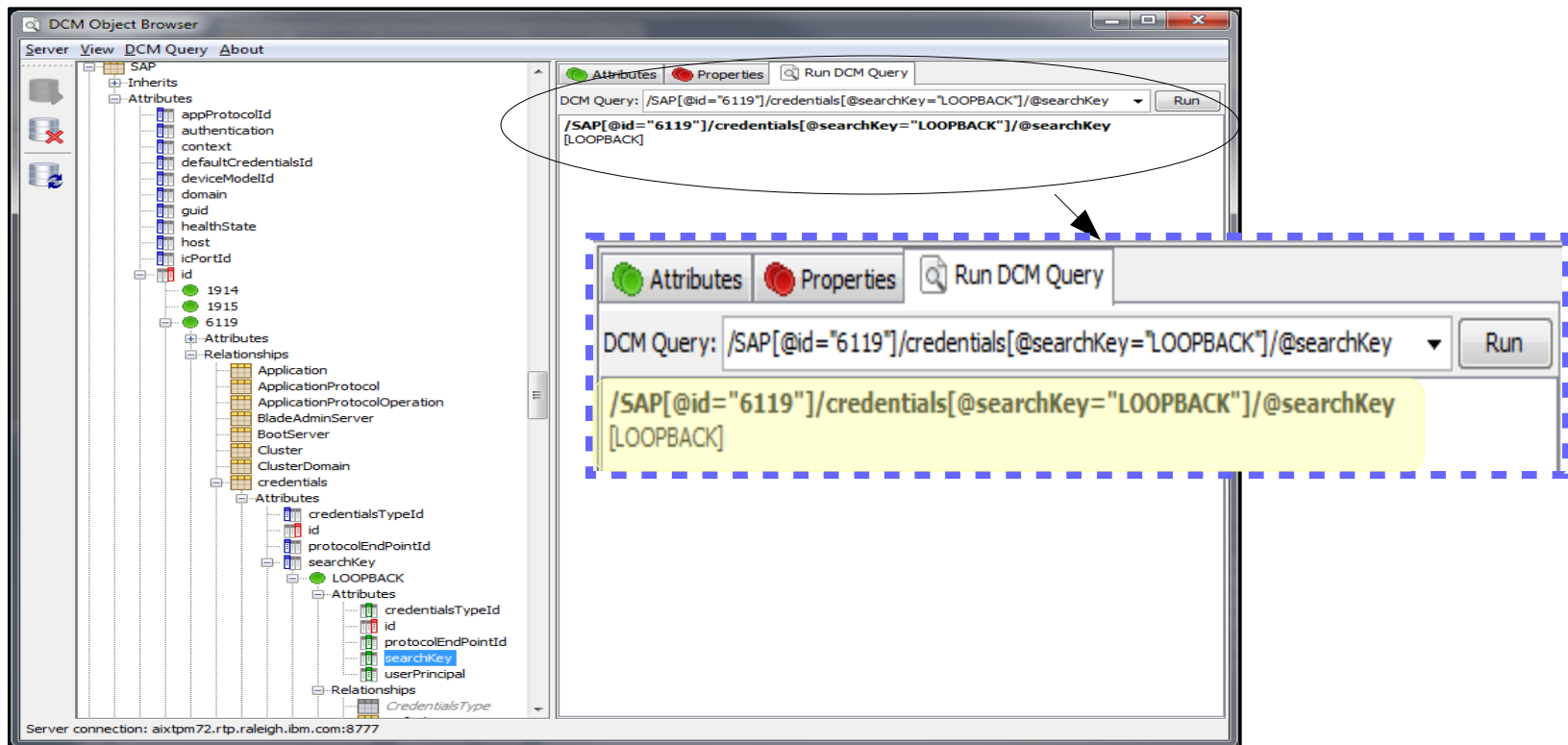
- continued...
- If that test workflow still fails, run the command directly on the system and verify that your syntax is correct.

```
-bash-3.2# lpar_netboot -t ent -D -s auto -d auto -A -f -S 1.2.3.4 -C 1.2.3.20 -G 1.2.3.1 -K 255.255.255.0  
-bash: lpar_netboot: command not found  
-bash-3.2#
```



# Troubleshooting Tips and Techniques

- DCM Object Browser tool available from the ISM library:
- <https://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW101099>
- Useful in troubleshooting DCMQuery relationships.



# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- [Reference Materials](#)
- Questions

## Reference Materials

- **TPM 7210: Performance Cookbook**
  - <http://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10109B>
- **TPM and TSAM Version 7: Database Configuration and Hygiene**
  - <http://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW101088>
- **TPM and TSAM Version 7: A DBMS Movement Solution**
  - <http://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW101082>
- **TPM 721 Online Documentation:**
  - <http://publib.boulder.ibm.com/infocenter/tivihelp/v45r1/index.jsp>
- **Service Management Connect - Data Center Automation Blog:**
  - <https://www.ibm.com/developerworks/mydeveloperworks/blogs/d6a38b59-943a-434b-a473-b408ed64847d>
- **TPM Support Site:**
  - [http://www-947.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli\\_Provisioning\\_Manager](http://www-947.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_Provisioning_Manager)
- **IBM Service Management videos:**
  - <http://www.youtube.com/user/ibmservicemanagement>

# Agenda

- Obtain the workflow log XML data
- Review the workflow log XML data
- Understand the workflow log XML data
- Compare workflow code against workflow log
- Troubleshooting Tips and Techniques
- Reference Materials
- Questions

# Q & A

