

IBM Tivoli Monitoring V6.2.3

Troubleshooting guidelines in the missing or incorrect real-time data scenario



© 2012 IBM Corporation

The main purpose of this presentation is to provide you with steps for proper and effective troubleshooting in a specific case: the missing or incorrect real-time data scenario, where the Tivoli® Monitoring agent returns no or incorrect data in Tivoli Enterprise Portal workspaces.

Assumptions

Before you proceed, the module designer assumes that you have these skills and knowledge:

- Tivoli Monitoring general architecture
- The troubleshooting approach based on the RAS1 logs

To proceed this module, you need to know the general architecture of IBM Tivoli Monitoring family, and to be familiar with the troubleshooting approach based on the RAS1 tracing and logging.

Objectives

When you complete this module, you can properly troubleshoot the monitoring product when the agent returns no data or incorrect data. This problem results in an incorrect or empty outcome in the Tivoli Enterprise Portal workspaces

When you complete this module, you can properly troubleshoot the scenario where the monitoring agent returns no data or incorrect data in one or more Tivoli Enterprise Portal workspaces.

Outline

- General information:
 - Product architecture
 - Configuration files and their location
 - RAS1 log files and their location
- Troubleshooting strategy:
 - Suggested trace settings
- Logical flow diagram:
 - Overview
 - Step1. Tivoli Enterprise Portal Server issues the SQL request
 - Step2. Tivoli Enterprise Monitoring Server manages the request
 - Step3. Collection is started at on the agent side
 - Step4. Data is sent back and the collection completed
 - Step5. Tivoli Enterprise Portal workspace is filled with data

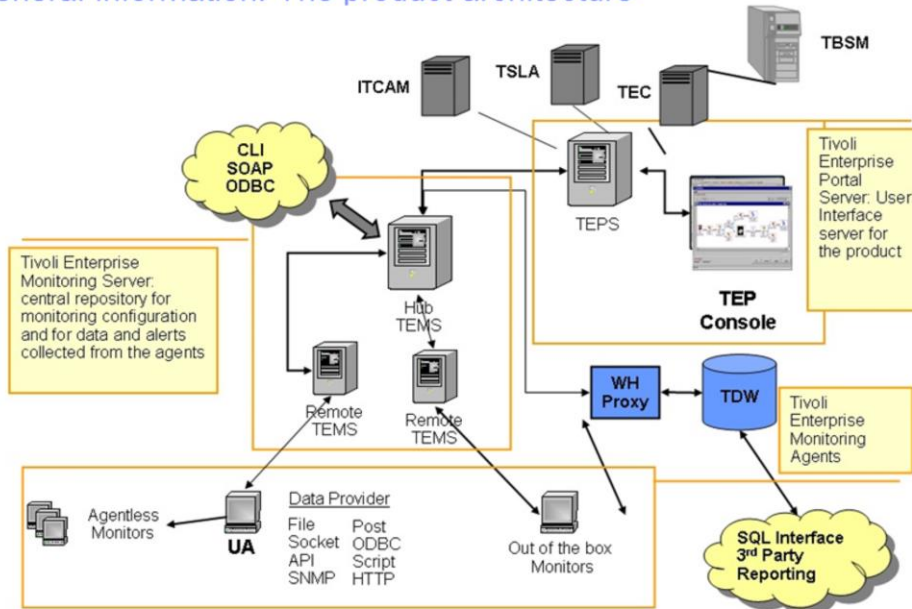
Here is the module outline.

In the first section, General information, you find highlights on the monitoring architecture, along with some practical information about the physical location of the configuration files and of the RAS1 log files.

The second part, Troubleshooting strategy, describes which level of traces you have to set on each of the involved components to properly investigate problems.

The third part, Logical flow diagram, is devoted to the logical flow of the communication messages exchanged among the components. This information is key to an effective approach to the analysis.

General Information: The product architecture



5

Troubleshooting guidelines in the missing or incorrect real-time data scenario

© 2012 IBM Corporation

You might be familiar with the Tivoli Monitoring architecture and its components. The core of the architecture is the Tivoli Enterprise Monitoring Server component, named Tivoli Enterprise Monitoring Server, which can be a hub or optionally (for scalability reasons in large environments) remote. The Tivoli Enterprise Monitoring Server manages the communication with the agents and owns the tables where the performance data (coming from the agents) is stored.

Tivoli Enterprise Monitoring Server is also responsible for the situation and policy management.

Tivoli Enterprise Portal Server is the presentation component of the collected data to the user. The Tivoli Enterprise Portal Server configuration data (that is, workspaces definitions, user permissions, and so on) are stored in a database that can be based on DB2®, Oracle, or SQLSERVER.

The agents, supporting most of the known platforms, are responsible for the data collection.

There is also a component devoted to the historical data collection. It is made up of a Warehouse Proxy agent and a Summarization and Pruning agent.

General information: The configuration files and their location

- Tivoli Enterprise Portal Server:
 - On Windows®: <install path>\CNPS\KFWENV
 - On UNIX® or Linux®: <install path>/config/cq.ini
- Tivoli Enterprise Monitoring Server:
 - On Windows: <install path>\CMS\KBBENV
 - On UNIX or Linux: <install path>/config/<tems name>_ms_<tems name>.config
 - On zOS: hlq.miq.RKANPAR(KDSENV)
- Tivoli Enterprise Managed Agent:
 - On Windows: <install path>\TMAITM6\<pc>env
 - On UNIX or Linux: <install path>/config/<pc>.ini
 - On zOS: hlq.miq.RKANPAR(K<pc>ENV)

This slide shows the physical location on the main supported operating system (OS) platforms, of the Tivoli Monitoring configuration files.

Details are provided for the three main components: Tivoli Enterprise Portal Server, Tivoli Enterprise Monitoring Server, and agent.

General information: The RAS1 log files and their location

- AGENTS:
 - On Windows: <install path>\TMAITM6\logs\ - On Unix or Linux: <install path>/logs/<hostname>_pc_<hextimestamp>-nn
 - On zOS: Rklvlog/Rkpdlog
- Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server:
 - On Windows: <install path>\logs\ - On UNIX or Linux: <install path>/logs/<hostname>_pc_<hextimestamp>-nn
 - On zOS: Rklvlog/Rkpdlog (Tivoli Enterprise Monitoring Server only)
- Tivoli Enterprise Portal Browser:
 - On Windows: c:\Documents and Settings\Administrator\ApplicationData\IBM\Java\Deployment\log\Plugin142.trace
 - On Unix or Linux: None
- Tivoli Enterprise Portal Desktop:
 - On Windows: <install path>\cnp\logs\kcjras1.log
 - On Unix or Linux: <installpath>/logs/kcjras1.log

This slide shows the physical location on the main supported OS platforms of the RAS1 log files.

Details are provided for the three main components: Tivoli Enterprise Portal Server, Tivoli Enterprise Monitoring Server, and agent.

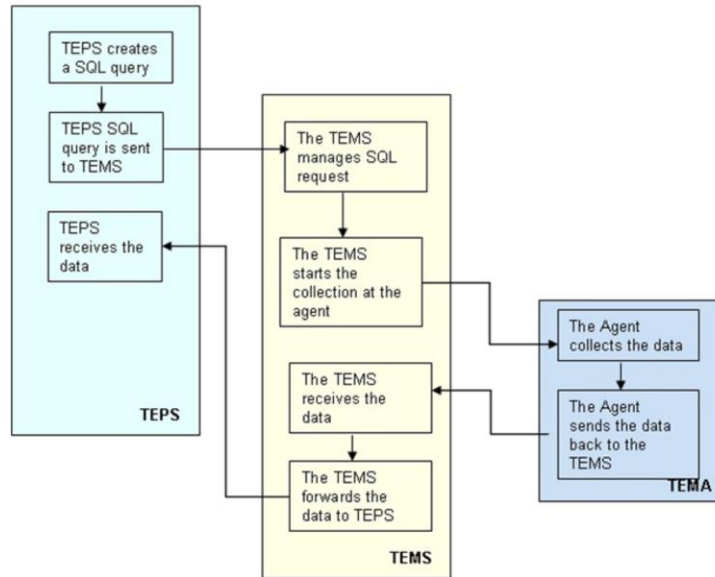
Troubleshooting strategy: Suggested trace settings

- Tivoli Enterprise Portal Server:
 - KBB_RAS1=ERROR (UNIT:ctsql IN,ER)
- Tivoli Enterprise Monitoring Server:
 - KBB_RAS1=ERROR (UNIT:kpxreq ALL) (UNIT:kdssqprs INPUT ERROR) (UNIT:kpxrprcq ALL)
- AGENT:
 - KBB_RAS1=ERROR (UNIT:bra ALL)
- In most cases, more specific traces are available on the agent, such as:
 - UNIT:k_pc_nn ALL (pc: product code, nn: progressive number that is, UNIT:kmq67 ALL)

The error scenario involves all of the three main components: the Tivoli Enterprise Portal Server, which places the data request; the Tivoli Enterprise Monitoring Server, which manages and drives such requests; and the agent, which performs the data collection.

Each of these components needs to be instrumented with specific traces, as shown in this slide.

Logical flow diagram: Overview



In this slide, you can see the general picture of the whole data flow behind this scenario. This flow is covered in depth in the following few slides.

Logical flow diagram: Step 1 (Tivoli Enterprise Portal Server issues the SQL request)

- Tivoli Enterprise Portal Server > SQL query is created
(2010.036 12.15.18-778:ctsqlstatement.cpp,212,"SQLStatement::SQLStatement")
HUB_LAB232157(39): SELECT WTLOGCLDSK.ORIGINNODE,
WTLOGCLDSK.INSTCNAME, WTLOGCLDSK.FREEMGBTES, WTLOGCLDSK.PCUSED,
WTLOGCLDSK.PCFREE, WTLOGCLDSK.TOTALSIZE, WTLOGCLDSK.PCTDSKRDTM,
WTLOGCLDSK.PCTDSKWRTM, WTLOGCLDSK.PCTDSKTIME,
WTLOGCLDSK.DSKQUELEN, WTLOGCLDSK.DSKPCTFRSP,
WTLOGCLDSK.TIMESTAMP FROM KNT.WTLOGCLDSK AT('HUB_LAB232157') WHERE
(WTLOGCLDSK.ORIGINNODE = ?) AND SYSTEM.PARMA("time out","600",3)
- Tivoli Enterprise Portal Server > SQL query variables are resolved with values
(2010.036 12.15.18-778:ctsqlstatement.cpp,237,"SQLStatement::SQLStatement")
HUB_LAB232157(39): Values: 'Primary:LAB232157:NT'

The workflow starts whenever a Tivoli Enterprise Portal user opens a navigator line item in order to see the data of interest.

The first part of the workflow is initiated by Tivoli Enterprise Portal Server: there is an underlying SQL query generated on Tivoli Enterprise Portal Server behind the data request. The slide shows what the Tivoli Enterprise Portal Server query looks like:

- The number in blue between brackets (39 in this case) is the query identifier.
- The AT clause in green specifies the hub or remote Tivoli Enterprise Monitoring Server that manages the query.
- The trace turned on the Tivoli Enterprise Portal Server allows you to have the full details of the query, and, as a consequence, of the columns and of the table. Each of the columns refers to a single attribute, while the table represents the attribute group.
- The **originnode** value in red is resolved with the name of the Managed System providing the data.

Logical flow diagram: Step 2 (Tivoli Enterprise Monitoring Server manages request) 1 of 2

- Tivoli Enterprise Monitoring Server > SQL Request is parsed
 - (2010.036 12.15.18-C40:kdssqprs.c,588,"PRS_ParseSql") SQL to be parsed is ... Ascii 0x00550000.
 - (2010.036 12.15.18-C40:kdssqprs.c,624,"PRS_ParseSql") SELECT WTLOGCLDSK.ORIGINNODE, WTLOGCLDSK.INSTCNAME, WTLOGCLDSK.FREEMGBTE
 -
 -
 - (2010.036 12.15.18-C40:kdssqprs.c,624,"PRS_ParseSql") SKQUELEN, WTLOGCLDSK.DSKPCTFRSP, WTLOGCLDSK.TIMESTAMP FROM KNT.WTLOGCLDS
 - (2010.036 12.15.18-C40:kdssqprs.c,624,"PRS_ParseSql") K AT('HUB_LAB232157') WHERE (WTLOGCLDSK.ORIGINNODE = ?) AND SYSTEM.PAR
 - (2010.036 12.15.18-C40:kdssqprs.c,624,"PRS_ParseSql") MA("TIMEOUT","600",3)
- Tivoli Enterprise Monitoring Server -> Request is created
 - (2010.036 12.15.18-1728:kpxreq.cpp,398,"AddTarget") Adding target <Primary:LAB232157:NT> to req ".WTLOGCLDSK <4161801971>
 - (2010.036 12.15.18-1728:kpxreqi.cpp,134,"RequestImp_constr") RequestImp RES1 Create handle 4161801970, owner 4161801971, obj 4C0E250, node "Primary:LAB232157:NT"

11

Troubleshooting guidelines in the missing or incorrect real-time data scenario

© 2012 IBM Corporation

The following part of the workflow takes place inside the Tivoli Enterprise Monitoring Server that:

- Picks up the request
- Parses the request itself

Any SQL syntax or logical errors are revealed at this time.

At this point, the workflow decides whether Tivoli Enterprise Monitoring Server is able to recognize and accept the Tivoli Enterprise Portal Server-provided query.

When the parsing process successfully completes, Tivoli Enterprise Monitoring Server is able to add the target for such request.

Primary:Lab232157:NT, in red, is the Tivoli Monitoring Agent for Windows acting as the target of the request issued here.

Handler and Owner numbers, in blue, represent the Tivoli Enterprise Monitoring Server, which handles the request and the agent being the target for the request.

Logical flow diagram: Step 2 (Tivoli Enterprise Monitoring Server manages request) 2 of 2

- Tivoli Enterprise Monitoring Server > Request is started
(2010.036 12.15.18-1728:kpxreq.cpp,737,"Start") Starting request
*.WTLOGCLDSK <4161801971>.
- Tivoli Enterprise Monitoring Server > CTIRA_connection request is managed
(2010.036 12.15.18-1728:kpxreqds.cpp,474,"Update") Request <4161801970> to
node Primary:LAB232157:NT **now has status 1**

(2010.036 12.15.18-1138:kpxrpcrq.cpp,365,"PrintSelf") RPC request
<4161801970> to node Primary:LAB232157:NT address
ip.pipe:#9.168.114.50[6015]

(2010.036 12.15.18-1138:kpxreqds.cpp,474,"Update") Request <4161801970> to
node Primary:LAB232157:NT **now has status 3**

(2010.036 12.15.18-1138:kpxrpcrq.cpp,269,"requestStart") RPC StartAgent
completed.

This slide still represents inside the Tivoli Enterprise Monitoring Server here. You see here, how the request is managed further.

First of all, Tivoli Enterprise Monitoring Server starts the request, clearly visible from the "Starting request" string that is associated to the number just highlighted.

Starting a request means that Tivoli Enterprise Monitoring Server started an RPC (remote procedure call) to the Node that is expected to return the data. Status 1 means that Tivoli Enterprise Monitoring Server is issuing a connection request to the agent, which manages the origin node.

When that RPC request is completed, the status changes from 1 to 3, meaning that the connection to IRA successfully completed.

This part of the logs and this information are key in case you suspect agent status or communication between the agent and Tivoli Enterprise Monitoring Server.

Logical flow diagram: Step 3 (collection started at agent side)

- Tivoli Enterprise Monitoring Agent > Collection is started at the agent
(2010.036 12.15.18-1258:kraafmgr.cpp,642,"Start") Start received <4161801970,0> on *.WTLOGCLDSK
- Tivoli Enterprise Monitoring Agent > Data is collected at the agent
(2010.036 12.15.18-1760:krant06b.cpp,303,"PrintSelf") _data {
(2010.036 12.15.18-1760:krant06b.cpp,304,"PrintSelf") count=4, allocated=20,
allocSize=20
(2010.036 12.15.18-1760:krant06b.cpp,309,"PrintSelf") {
(2010.036 12.15.18-1760:krant06b.cpp,310,"PrintSelf") originnode="Primary:LAB232157:NT "
(2010.036 12.15.18-1760:krant06b.cpp,311,"PrintSelf") sampleno=24
(2010.036 12.15.18-1760:krant06b.cpp,312,"PrintSelf") rowno=1
(2010.036 12.15.18-1760:krant06b.cpp,313,"PrintSelf") timestamp="1100205121518921"
(2010.036 12.15.18-1760:krant06b.cpp,316,"PrintSelf") instcname="C: "
(2010.036 12.15.18-1760:krant06b.cpp,317,"PrintSelf") pctdskrdtm=0
(2010.036 12.15.18-1760:krant06b.cpp,318,"PrintSelf") pctdsktime=0
(2010.036 12.15.18-1760:krant06b.cpp,319,"PrintSelf") pctdskwrtm=0

Now, you see the view from the agent side. The agent receives the request holding the same identifier as in the previous slide. The 4161801970 number, in red, indicates to the agent which Tivoli Enterprise Monitoring Server originated the received request.

The agent is able to collect the data, and from the sequence of "PrintSelf" in this slide, you can understand what are the values returned by the agent for each column of the query.

At this point, you realize whether the agent is returning incorrect data, or if you need to troubleshoot further the agent data provider (not covered in this module).

Logical flow diagram: Step 4 (data sent back and collection completed)

- Tivoli Enterprise Monitoring Agent > Collection is completed
(2010.036 12.15.18-1258:kraafmgr.cpp,674,"Start") Start complete
<4161801970,3052406689> on *.WTLOGCLDSK, status = 0
- Tivoli Enterprise Monitoring Agent > Data is sent back to the monitoring server
(2010.036 12.15.18-1760:kraadspt.cpp,254,"sendDataToProxy")
Sending 4 rows for KNT.WTLOGCLDSK, <4161801970,3052406689>.
- Tivoli Enterprise Monitoring Server > Data arrives at the monitoring server
(2010.036 12.15.18-13BC:kpxrprq.cpp,743,"IRA_NCS_Sample") Rcvd 4 rows sz 356
tbl *.WTLOGCLDSK req <4161801970,3052406689> node <Primary:LAB232157:NT>
(2010.036 12.15.18-13BC:kpxreqds.cpp,474,"Update") Request <4161801970> to
node Primary:LAB232157:NT now has status 7
- Tivoli Enterprise Portal Server > TEPS QUERY is returned with data
(2010.036 12.15.18-
61C:ctsqlaccesssql1.cpp,1001,"CTSQLEvaluatorSQL1_i::AccessElement::pullSequenc
eWithTimeout") HUB_LAB232157(39): Rows returned: 4

This slide details the end of the workflow; the data-return phase.

At the agent side, the collection is complete and the agent can rest before another cycle of work. The collected data is sent back to Tivoli Enterprise Monitoring Server.

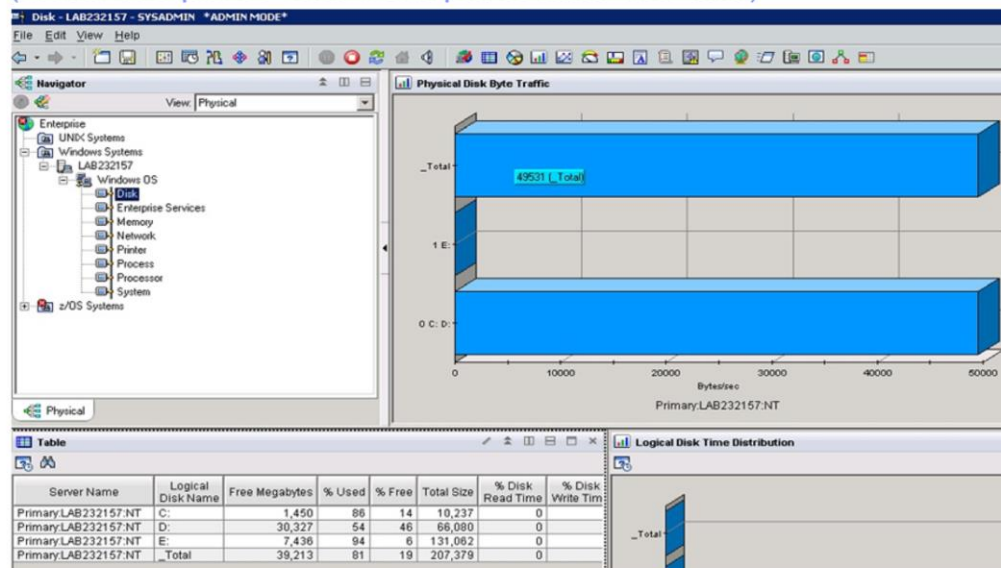
This activity is the meaning of the "sending 4 rows for" message.

Tivoli Enterprise Monitoring Server receives data and changes the status of request to 7: this value means that data arrived.

Status=7 is a good sign that you need to check in your Tivoli Enterprise Monitoring Server log to ensure that your request of data completed successfully.

The last step of the workflow is that the Tivoli Enterprise Portal Server SQL query is completed with the 4 rows of data being returned.

Logical flow diagram: Step 5 (Tivoli Enterprise Portal workspaces filled with data)



15

Troubleshooting guidelines in the missing or incorrect real-time data scenario

© 2012 IBM Corporation

This slide shows the four rows of data returned.

You can use the 'PrintSelf' statements highlighted in the previous slides to compare them against the values in the tables shown in the Tivoli Enterprise Portal interface. This matching ensures that the Tivoli Enterprise Portal Server is showing exactly the data that the Tivoli Enterprise Monitoring Server passed back.

Summary

Now that you have completed this module, you have a clearer understanding of how to troubleshoot missing or incorrect real-time data issues.

Now that you have completed this module, you know how to properly troubleshoot the scenario where the monitoring agent returns no data or incorrect data in one or more Tivoli Enterprise Portal workspaces.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.