



ITM 6.x – Scenario-based troubleshooting

Scenario #1 : Missing real data on TEP console

Yew Hoong Ng – Global Response Team Asia Pacific

Tivoli software

ON DEMAND BUSINESS™

Scenario #1

- User cannot get performance data on TEP console.
- User gets wrong performance data on TEP console.
- Slow response time to get performance data from agent.

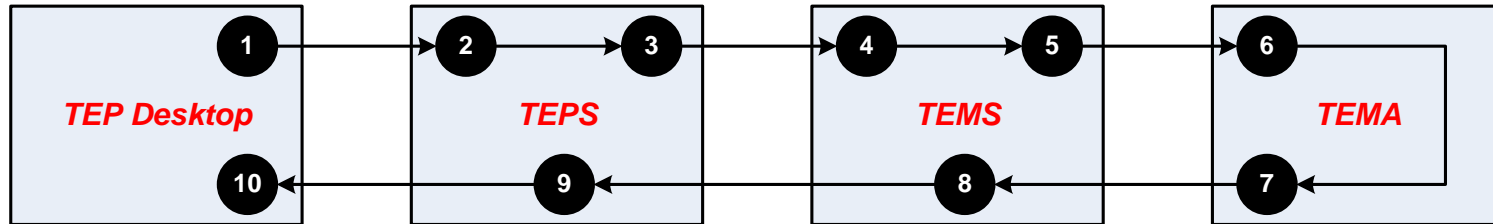


Requirements

- Increase trace level of TEPS to ERROR (UNIT:ctsql INPUT,ERROR).
- Increase trace level of TEMS to ERROR (UNIT:kdssqprs INPUT) (UNIT:kpx ALL).
- Increase trace level of TEMA to ERROR (UNIT:kra ALL).
- Simulate the problem and collect RAS1 log files from all components.
- All log files' timestamp must be synchronized in time.



Diagram and flow description



- 1 User clicks on a Workspace
- 2 TEPS receives the request (unit:ctdatabus)
- 3 SQL statement is created with **query id** as a unique identifier (unit:ctsqlstatement)
- 4 TEPS' SQL request is parsed (unit:kdssqprs) and TEMS' SQL request is created (unit:kpxreqds)
A unique **request handle** is created
- 5 RPC request to TEMA is created with **request handle** as identifier (unit:kpxrpcrq)
- 6 A request is received (unit:kraafmgr) and data collection is started (unit:kraafira)
The **request handle** is used as a reference
- 7 The result is sent back to TEMS using **request handle** as a reference (unit:kraadspt)
- 8 TEMS receives the result using **request handle** as identifier (unit:kpxrpcrq)
- 9 TEPS receives the result using **query id** as identifier (unit:ctsqlaccesssql1)
- 10 User sees the result on a Workspace

Example (TEPS)

- SQL statement is created with **query id** as a unique identifier

```
(4BD16F49.0055-11:ctsqlstatement.cpp,212,"SQLStatement::SQLStatement")  
TEMS(74): SELECT UNIXDISK.ORIGINNODE, UNIXDISK.UMOUNTPT,  
UNIXDISK.UDSKNAME, UNIXDISK.DSKSIZE, UNIXDISK.SPCAVAIL, UNIXDISK.SPCUSED,  
UNIXDISK.PCTSPCUSED, UNIXDISK.INODESIZE, UNIXDISK.INODEFREE,  
UNIXDISK.INODEUSED, UNIXDISK.PCTINDUSED, UNIXDISK.TIMESTAMP FROM  
OMUNX.UNIXDISK AT( 'TEMS' ) WHERE (UNIXDISK.ORIGINNODE = ?) AND  
SYSTEM.PARMA("TIMEOUT","600",3)
```

```
(4BD16F49.0056-11:ctsqlstatement.cpp,237,"SQLStatement::SQLStatement")  
TEMS(74): Values: 'IS11:KUX'
```

Example (TEMS)

- TEPS' SQL request is parsed (unit:*kdssqprs*) and TEMS' SQL request is created (unit:*kpxreqds*); A unique **request handle** is created


```
(4BD16F49.002C-E:kpxreq.cpp,400,"AddTarget") Entry
(4BD16F49.002D-E:kpxreq.cpp,408,"AddTarget") Adding target <IS11:KUX> to req
*.UNIXDISK <294655258>
(4BD16F49.002E-E:kpxreqi.cpp,134,"RequestImp_constr") RequestImp RES1 Create handle
294655257, owner 294655258, obj 34F32758, node "IS11:KUX"
(4BD16F49.004B-E:kpxreqds.cpp,482,"Update") Request <294655257> to node IS11:KUX now
has status 1
(4BD16F49.004F-E:kpxreqi.cpp,778,"UseRequestImp") RequestImp RES1 Use handle
294655257: 34F32758
(4BD16F49.0051-E:kpxreqmg.cpp,310,"QueueTask") Queing START PRIO req 0x34F32758
handle 294655257 node IS11:KUX
```
- RPC request to TEMA is created with **request handle** as identifier (unit:*kpxrpcrq*)


```
(4BD16F49.0057-F:kpxreqi.cpp,835,"startRequestTask") Starting req @0x34F32758
<294655257>
(4BD16F49.0059-F:kpxrpcrq.cpp,365,"PrintSelf") RPC request <294655257> to node
IS11:KUX address ip.pipe:#172.16.202.24[6015]
(4BD16F49.005A-F:kpxreqi.cpp,721,"RequestImp_PrintSelf") Request to IS11:KUX,
<294655257,0> obj: 0x34F32758, retries:0, flags:0x98000000, status:1,
owner:0x3503FF88
(4BD16F49.0063-F:kpxreqds.cpp,482,"Update") Request <294655257> to node IS11:KUX now
has status 3
(4BD16F49.0069-F:kpxreqi.cpp,794,"BaseDrop") RequestImp RES1 Drop handle 294655257:
34F32758
```

Example (TEMA)

- A request is received (unit:*kraafmgr*) and data collection is started (unit:*kraafira*); The **request handle** is used as a reference.

```
(4BD16F49.000C-1:kraafmgr.cpp,590,"Start") Start received
<294655257,0> on *.UNIXDISK
```

```
(4BD16F49.0027-1:kraafmgr.cpp,622,"Start") Start complete
<294655257,428868573> on *.UNIXDISK, status = 0
```

```
(4BD16F49.002D-5:kraafira.cpp,568,"DriveDataCollection")
OMUNX.UNIXDISK, <294655257,428868573> expired.
```

```
(4BD16F49.00BA-5:kraux02b.cpp,292,"PrintSelf")
omunx_unixdisk_base @1107A0390 <294655257,428868573>
```

- The result is sent back to TEMS using **request handle** as a reference (unit:*kraadspt*)

```
(4BD16F49.0244-5:kraadspt.cpp,270,"sendDataToProxy") Sending 14
rows for OMUNX.UNIXDISK, <294655257,428868573>
```



Example (TEMS)

- TEMS receives the result using **request handle** as identifier (unit: *kpxrpcrq*)
 - (4BD16F49.0128-55:kpxreqi.cpp,778,"UseRequestImp") RequestImp RES1 Use handle **294655257**: 34F32758
 - (4BD16F49.0129-55:kpxreqi.cpp,794,"BaseDrop") RequestImp RES1 Drop handle **294655257**: 34F32758
 - (4BD16F49.012B-55:kpxreqi.cpp,778,"UseRequestImp") RequestImp RES1 Use handle **294655257**: 34F32758
 - (4BD16F49.012C-55:kpxrpcrq.cpp,743,"IRA NCS Sample") Rcvd 14 rows sz 1088 tbl *.UNIXDISK req <**294655257**,428868573> node <IS11:KUX>
 - (4BD16F49.012E-55:kpxreqds.cpp,482,"Update") Request <**294655257**> to node IS11:KUX now has status 7
 - (4BD16F49.013C-55:kpxreqi.cpp,794,"BaseDrop") RequestImp RES1 Drop handle **294655257**: 34F32758
 - (4BD16F49.0150-E:kpxreqi.cpp,814,"BaseDestroy") RequestImp RES1 Destroy handle **294655257**: 34F32758
 - (4BD16F49.0151-E:kpxreqi.cpp,794,"BaseDrop") RequestImp RES1 Drop handle **294655257**: 34F32758
 - (4BD16F49.0155-E:kpxreqi.cpp,144,"RequestImp_destr") RequestImp RES1 Delete handle **294655257**, owner 294655258, obj@34F32758
 - (4BD16F49.0156-E:kpxreqds.cpp,482,"Update") Request <**294655257**> to node IS11:KUX now has status 8

Example (TEPS)

- TEPS receives the result using **query id** as identifier (unit: *ctsql/accesssql1*)

4BD16F49.005F-

```
C:ctsql/accesssql1.cpp,1001,"CTSQLEvaluatorSQL1_i::AccessElement:  
:pullSequenceWithTimeout") TEMS(74): Rows returned: 14
```

Reference – TEMS Proxy request table

- 0 = CTIRA_init
- 1 = CTIRA_connect_requested
- 2 = CTIRA_disconnect_requested
- 3 = CTIRA_connected
- 4 = CTIRA_connect_failure
- 5 = CTIRA_disconnected
- 6 = CTIRA_disconnect_failure
- 7 = CTIRA_data_arrived
- 8 = CTIRA_deleted
- 9 = CTIRA_persist_connect_requested
- 10 = CTIRA_persist_connected
- 11 = CTIRA_persist_connect_failed

