

# ITM 6.x – Scenario-based troubleshooting

*Scenario #2 : Missing short-term historical data on TEP console*

***Yew Hoong Ng – Global Response Team Asia Pacific***



## Scenario #2a

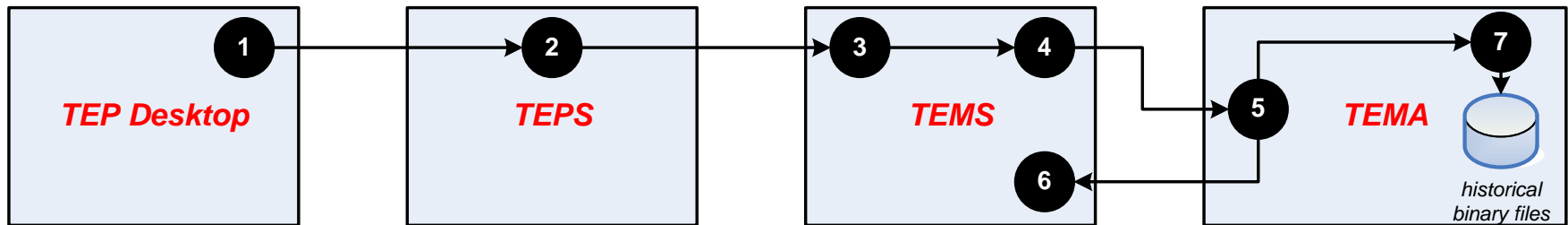
- Historical data collection has been configured but not started



## 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 configures historical collection
- 2 TEPS sends a request to update table O4SRV.TSITDESC for a new historical Situation (UADVISOR\_\*) (unit: *ctsqlstatement*)
- 3 TEPS request is received and TEMS' SQL request to start historical collection is created (unit: *kpxreqds*); a unique **request handle** is created
- 4 RPC request to TEMA is created with the **request handle** as identifier (unit: *kpxrpcrq*).
- 5 A request is received (unit: *kraafmgr*) and historical data collection is started (unit: *kraafmgr*). The **request handle** is used as a reference.
- 6 TEMS request is confirmed (unit: *kpxrpcrq*). The **request handle** is used as a reference.
- 7 Historical collection starts (unit: *kraahist*) and data is written to binary files (unit: *kraahbin*).

## Example (TEPS)

- TEPS sends a request to update table O4SRV.TSITDESC for a new historical Situation (UADVISOR\_\*) (unit: *ctsqlstatement*)

```
(4C762406.001C-14F0:ctsqlstatement.cpp,213,"SQLStatement::SQLStatement")
HUB_ITMDVD26(161): INSERT INTO O4SRV.TSITDESC (SITNAME, TEXT, AFFINITIES, PDT,
REEV_TIME, REEV_DAYS, ALERTLIST, AUTOSTART, SENDMSGQ, LSTDATE, LSTUSRPRF,
LSTRELEASE, LSTCCSID, PRNAMES, CMD, AUTOSOFT, ADVISE, SITINFO, QIBSCOPE, QIBCLASSID
) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
(4C762406.001D-14F0:ctsqlstatement.cpp,238,"SQLStatement::SQLStatement")
HUB_ITMDVD26(161): Values: 'UADVISOR_KNT_WTLOGCLDSK' '' '%IBM.STATIC021
00000000010000000000' '*IF *HISTRULE SELECT
ORIGINNODE,SAMPLENO,ROWNO,TIMESTAMP,INSTCNAME,PCTDSKRDTM,PCTDSKTIME,PCTDSKWRTM,DSKP
CTFRSP,DSKBTSSEC,DSKQUELEN,DSKRDBTSEC,DSKRDSEC,DSKTRNSSEC,DSKWRTSEC,DSKWRTBYTS,FREE
MGBTES,TOTALSIZE,PCUSED,PCFREE,PHYDISKNUM,AVGQLEN,AVGRDQLEN,AVGWRQLEN,DSKBTSSEC8,DS
KRDBTSE8,DSKWRTBYT8 FROM KNT.WTLOGCLDSK WHERE SYSTEM.PARMA("HISTORY","IRA",3) AND
SYSTEM.PARMA("SITNAME","UADVISOR_KNT_WTLOGCLDSK",23) AND
SYSTEM.PARMA("INTERVAL","300000",10)' '000500' '0' '' '*SYN' '*NONE'
'1100826042127000' 'SYSADMIN' 'V100' '65535' '' '*NONE' 'NNN' '*NONE'
'TFWD=N;OV=N;~' 'E' '5140'
```

## Example (TEMS)

- TEPS request is received and TEMS' SQL request to start historical collection is created (unit:*kpxreqds*); a unique **request handle** is created

```
(4C7624EC.000F-E94:kpxreqds.cpp,325,"InitializeRequest") Request:
  UADVISOR_KNT_WTLOGCLDSK <4226812561> *.WTLOGCLDSK
(4C7624EC.003D-E94:kpxreq.cpp,408,"AddTarget") Adding target <Primary:ITMDVD26:NT>
  to req UADVISOR_KNT_WTLOGCLDSK *.WTLOGCLDSK <4226812561>
(4C7624EC.003E-E94:kpxreqi.cpp,134,"RequestImp_constr") RequestImp RES1 Create
  handle 4224715408, owner 4226812561, obj 707DBD8, node "Primary:ITMDVD26:NT"
```

- RPC request to TEMA is created with **request handle** as identifier (unit:*kpxrpcrq*)

```
(4C7624EC.004C-E94:kpxreq.cpp,754,"Start") Starting request UADVISOR_KNT_WTLOGCLDSK
  *.WTLOGCLDSK <4226812561>.
(4C7624EC.004F-E94:kpxreqds.cpp,519,"Update") Request <4224715408> to node
  Primary:ITMDVD26:NT now has status 1
(4C7624EC.0059-E94:kpxrpcrq.cpp,365,"PrintSelf") RPC request <4224715408> to node
  Primary:ITMDVD26:NT address ip:#10.100.100.89[38213]
(4C7624EC.0065-DE4:kpxrpcrq.cpp,173,"requestStart") StartAgent instruction UTF8: 0,
  len: 31[HIST()LSTDATE(1100826042517000)]
(4C7624EC.0069-E94:kpxcloc.cpp,2145,"KPXLOC_TakeSample") TakeSample for
  UADVISOR_KNT_WTLOGCLDSK <4226812561> exit status 77, defer_once <1>.
(4C7624EC.006A-DE4:kpxreqds.cpp,519,"Update") Request <4224715408> to node
  Primary:ITMDVD26:NT now has status 3
(4C7624EC.006E-DE4:kpxrpcrq.cpp,269,"requestStart") RPC StartAgent completed.
  (4C7624EC.006F-DE4:kpxrpcrq.cpp,277,"requestStart") Exit
```

## Example (TEMA)

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

```
(4C7624EC.0010-177C:kraafmgr.cpp,619,"Start") Start received
UADVISOR_KNT_WTLOGCLDSK <4224715408,0> on *.WTLOGCLDSK

(4C7624EC.0011-177C:kraafmgr.cpp,623,"Start") Value of instruction
<HIST() LSTDATE(1100826042517000)>

(4C7624EC.002D-177C:kraafira.cpp,308,"ctira") Creating request @0x5aac3d0
<4224715408,4277143388> KNT.WTLOGCLDSK, UADVISOR_KNT_WTLOGCLDSK

(4C7624EC.002E-177C:kraaevx.cpp,479,"CreateSituationEvent") *EV-INFO:
Input event: obj=0x5AAC3D0, type=1, excep=0, numbRow=0, rowData=0xNULL,
status=0, sitname="UADVISOR_KNT_WTLOGCLDSK"

(4C7624EC.0041-177C:kraafmgr.cpp,651,"Start") Start complete
UADVISOR_KNT_WTLOGCLDSK <4224715408,4277143388> on *.WTLOGCLDSK, status
= 0
```



## Example (TEMA)

- Historical collection starts (unit:*kraahist*) and data is written to binary files (unit:*kraahbin*).

```
(4C762618.0093-F90:kraahist.cpp,135,"writeData") Entry
(4C762618.0098-F90:kraahbin.cpp,232,"WriteMetaFile") Metafile name is
<C:\IBM\ITM\TMAITM6\logs\WTLOGCLDSK.hdr>
(4C762618.009A-F90:kraahbin.cpp,337,"WriteRow")  _buffer_size: 408.
Application data size: 380
(4C762618.009B-F90:kraahbin.cpp,378,"WriteRow") Using timestamp of
1100826043017
(4C762618.009C-F90:krabufwk.cpp,306,"IRA_GetTimeZoneOffset") Active RAS1
Classes: EVERYT EVERYE EVERYU
(4C762618.009D-F90:kraahbin.cpp,399,"WriteRow") *INFO: History Record
TimeZone Offset +14400
(4C762618.009E-F90:kraahbin.cpp,499,"WriteRow") Samples = 10, timestamp =
1100826043017
(4C762618.009F-F90:kraahbin.cpp,499,"WriteRow") Samples = 11, timestamp =
1100826043017
(4C762618.00A0-F90:kraahbin.cpp,531,"WriteRow") Wrote 2 rows history data,
UADVISOR_KNT_WTLOGCLDSK KNT.WTLOGCLDSK, <4224715408,4277143388>.
(4C762618.00A2-F90:kraahist.cpp,144,"writeData") Exit
```

- TEMA operational log

```
1100826162517781KRAIRA000 Starting Enterprise situation
UADVISOR_KNT_WTLOGCLDSK <4224715408,4277143388> for KNT.WTLOGCLDSK.
IRA Constructor
```



## More analysis

- TEPS' query id is used to track an SQL request to TEMS. In this case, TEPS' query id = 161.
- TEPS updates Situation table in EIB (O4SRV.TSITDESC) on TEMS.
- Request handle is very important because this one piece of information can glue information between TEMS and TEMA. In our example, TEMS request handle = 4224715408.
- Metafile for historical binary files will be created once the interval is expired.
- Different stage of TEMS Proxy request is identified by its status. For example after request is successfully started TEMS Proxy request status is set to 3, which means CTIRA is connected. List of Proxy request status is shown in the reference slide.

## Scenario #2b

- User cannot get short-term historical data on TEP console.
- Slow response time to get short-term historical data from agent.

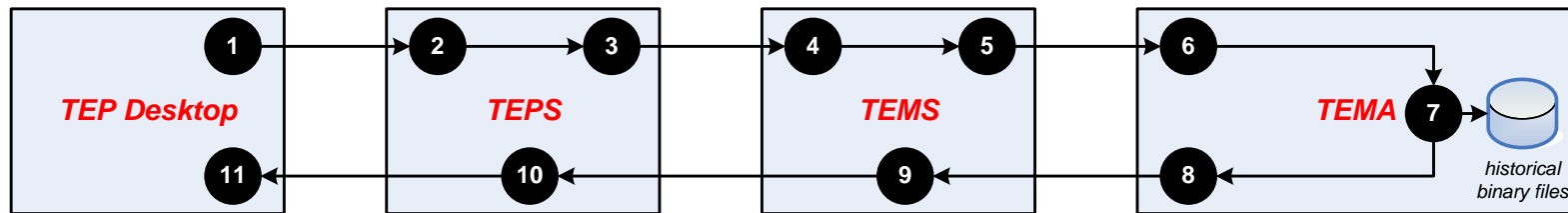


## Requirements

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



## Diagram & flow description



- 1 User clicks on a Workspace
- 2 TEPS receives the request (unit:ctdatabus)
- 3 A HISTORY SQL statement is created with **query id** as a unique identifier (unit:ctsqlstatement)
- 4 TEPS' SQL request is parsed (*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); the request consists of **HISTREAD** for historical data request.
- 6 A request is received (unit:kraafmgr) and an instruction is created with HISTREAD for historical data request; the request handle is used as reference
- 7 Data is read from historical binary files (unit:krarhagt)
- 8 The result is sent back to TEMS using **request handle** as a reference (unit:kraadspst)
- 9 TEMS receives the result using **request handle** as identifier (unit:kpxrpcrq)
- 10 TEPS receives the result using **query id** as identifier (unit:ctsqlaccesssql1)
- 11 User sees the result on the Workspace

## Example (TEPS)

- A HISTORY SQL statement is created with **query id** as a unique identifier (unit: *ctsqlstatement*)

```
(4C76A9F9.0002-
AD4:ctsqlstatement.cpp,213,"SQLStatement::SQLStatement")
HUB_ITMDVD26(1023): SELECT WTLOGCLDSK.TIMESTAMP,
WTLOGCLDSK.ORIGINNODE, WTLOGCLDSK.INSTCNAME,
WTLOGCLDSK.PHYDISKNUM, WTLOGCLDSK.TOTALSIZE,
WTLOGCLDSK.PCUSED, WTLOGCLDSK.FREEMGBTES, WTLOGCLDSK.PCFREE,
WTLOGCLDSK.DSKQUELEN, WTLOGCLDSK.PCTDSKTIME,
WTLOGCLDSK.PCTDSKRD TM, WTLOGCLDSK.PCTDSKWRTM,
WTLOGCLDSK.DSKPCTFRSP FROM KNT.WTLOGCLDSK AT( 'HUB_ITMDVD26'
) HISTORY() WHERE ( (WTLOGCLDSK.TIMESTAMP >= ?) AND
(WTLOGCLDSK.TIMESTAMP <= ?) ) AND ( (WTLOGCLDSK.ORIGINNODE =
?) AND SYSTEM.PARMA("TIMEOUT","600",3) )
(4C76A9F9.0003-
AD4:ctsqlstatement.cpp,238,"SQLStatement::SQLStatement")
HUB_ITMDVD26(1023): Values: '1100830214557000'
'1100831024557000' 'Primary:ITMDVD26:NT'
```

## Example (TEMS)

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

```
(4C76A9F9.0128-D9C:kpxreq.cpp,408,"AddTarget") Adding target <Primary:ITMDVD26:NT> to req *.WTLOGCLDSK <15731389>
```

```
(4C76A9F9.0129-D9C:kpxreqi.cpp,134,"RequestImp_constr") RequestImp RES1 Create handle 4277144246, owner 15731389, obj 6DC67A8, node "Primary:ITMDVD26:NT"
```

```
(4C76A9F9.0140-D9C:kpxreq.cpp,1779,"AddInstructions") Add instruction 220 <HISTREAD(TABLE(WTLOGCLDSK)APPL(KNT)COLS(DSKPCTFRSP:180,PCTDSKWRTM:176,PCTDSKRDTM:172,PCTDSKTIME:168,DSKQUELEN:164,PCFREE:160,FREEMGBTES:156,PCUSED:152,TOTALSIZE:148,PHYDISKNUM:144,INSTCNAME:80,ORIGINNODE:16,TIMESTAMP:0))>
```
- RPC request to TEMA is created with **request handle** as identifier (unit:*kpxrpcrq*); the request consists of **HISTREAD** for historical data request.
 

```
(4C76A9F9.0144-D9C:kpxreq.cpp,754,"Start") Starting request *.READHIST <15731389>.
```

```
(4C76A9F9.0147-D9C:kpxreqds.cpp,519,"Update") Request <4277144246> to node Primary:ITMDVD26:NT now has status 1
```

```
(4C76A9F9.014B-D9C:kpxreqi.cpp,823,"UseRequestImp") RequestImp RES1 Use handle 4277144246: 6DC67A8
```

```
(4C76A9F9.0157-DE8:kpxrpcrq.cpp,365,"PrintSelf") RPC request <4277144246> to node Primary:ITMDVD26:NT address ip:#10.100.100.89[38213]
```

```
(4C76A9F9.0158-DE8:kpxreqi.cpp,766,"RequestImp_PrintSelf") Request to Primary:ITMDVD26:NT, <4277144246,0> obj: 0x6DC67A8, retries:0, flags:0x98000000, status:1, owner:0x96C2008
```

```
(4C76A9F9.015F-DE8:kpxrpcrq.cpp,173,"requestStart") StartAgent instruction UTF8: 0, len: 220 [HISTREAD(TABLE(WTLOGCLDSK)APPL(KNT)COLS(DSKPCTFRSP:180,PCTDSKWRTM:176,PCTDSKRDTM:172,PCTDSKTIME:168,DSKQUELEN:164,PCFREE:160,FREEMGBTES:156,PCUSED:152,TOTALSIZE:148,PHYDISKNUM:144,INSTCNAME:80,ORIGINNODE:16,TIMESTAMP:0))] ]
```

```
(4C76A9F9.016D-DE8:kpxreqds.cpp,519,"Update") Request <4277144246> to node Primary:ITMDVD26:NT now has status 3
```

```
(4C76A9F9.0171-DE8:kpxrpcrq.cpp,269,"requestStart") RPC StartAgent completed.
```

## Example (TEMA)

- A request is received (unit:*kraafmgr*) and an instruction is created with HISTREAD for historical data request; the **request handle** is used as reference
 

```
(4C76A9FA.0013-17E4:kraafmgr.cpp,619,"Start") Start received <4277144246,0> on *.READHIST
(4C76A9FA.0014-17E4:kraafmgr.cpp,623,"Start") Value of instruction
<HISTREAD(TABLE (WTLOGCLDSK) APPL (KNT) COLS (DSKPCTFRSP:180,PCTDSKWRTM:176,PCTDSKRDtm:172,PCTDSKTIME:168,D
SKQUELEN:164,PCFREE:160,FREEMGBTES:156,PCUSED:152,TOTALSIZE:148,PHYDISKNUM:144,INSTCNAME:80,ORIGINNODE
:16,TIMESTAMP:0))>
(4C76A9FA.001D-17E4:kraafira.cpp,308,"ctira") Creating request @0x5bdc5e8 <4277144246,133171048>
KPX.READHIST,
(4C76A9FA.0026-17E4:krarhagt.cpp,937,"DetermineTableName") Table name requested is <WTLOGCLDSK>.
```
- Data is read from historical binary files (unit:*krarhagt*)
 

```
(4C76A9FA.0038-17E4:krarhagt.cpp,227,"SetFileNames") Setting Datafile
<C:\IBM\ITM\TMAITM6\logs\WTLOGCLDSK>, Metafile <C:\IBM\ITM\TMAITM6\logs\WTLOGCLDSK.hdr>
(4C76A9FA.0061-17E4:krabhsco.cpp,470,"KRA_parseMetaBuffer") History data record length set to <408>
(4C76A9FA.0074-17E4:krarhagt.cpp,1100,"TakeSampleConstructor") Row size set to 184, table WTLOGCLDSK
(4C76A9FA.0075-17E4:krarhagt.cpp,1115,"TakeSampleConstructor") Allocating history read buffer of 408
bytes, table WTLOGCLDSK.
(4C76A9FA.0077-17E4:krabhsco.cpp,93,"Print KRA HistoryData") KRA HistoryData struct @5BDC6DC
columnList=5BD0FC0, tablename=WTLOGCLDSK, SourceFileName=C:\IBM\ITM\TMAITM6\logs\WTLOGCLDSK,
MetaFileName=C:\IBM\ITM\TMAITM6\logs\WTLOGCLDSK.hdr
(4C76A9FA.008A-F90:krabhsco.cpp,633,"KRA_OpenHistoryFiles") Opened source file
C:\IBM\ITM\TMAITM6\logs\WTLOGCLDSK
(4C76A9FA.0763-F90:kraafira.cpp,993,"InsertRow") rowsize = 184, newsize = 5, newbytes = 920, _allocated
= 0, _allocSize = 5
```
- The result is sent back to TEMS using **request handle** as a reference (unit:*kraadspt*)
 

```
(4C76A9FA.07C3-F90:kraadspt.cpp,706,"sendDataToProxy") Sending 8 rows for KPX.READHIST,
<4277144246,133171048>.
(4C76A9FA.07CA-17E4:kraafmgr.cpp,651,"Start") Start complete <4277144246,133171048> on *.READHIST,
status = 0
```



## Example (TEMS)

- TEMS receives the result using **request handle** as identifier (unit: *kpxrpcrq*)

```
(4C76A9F9.0177-5D4:kpxreqi.cpp,823,"UseRequestImp")
```

```
RequestImp RES1 Use handle 4277144246: 6DC67A8
```

```
(4C76A9F9.0178-5D4:kpxrpcrq.cpp,749,"IRA_NCS_Sample") Rcvd 8  
rows sz 184 tbl *.READHIST req <4277144246,133171048> node  
<Primary:ITMDVD26:NT>
```

```
(4C76A9F9.0179-5D4:kpxreqdt.cpp,68,"SetData") SetData number  
of rows=<8>
```

```
(4C76A9F9.017A-5D4:kpxreqds.cpp,519,"Update") Request  
<4277144246> to node Primary:ITMDVD26:NT now has status 7
```





## Example (TEPS)

- TEPS receives the result using **query id** as identifier  
(unit:*ctsqlaccesssql1*)

```
(4C76A9FA.0000-  
14B4:ctsqlaccesssql1.cpp,1015,"CTSQLEvaluatorSQL1_i::AccessElement::pullSequenceWithTimeout")  
HUB_ITMDVD26(1023): Rows returned: 8
```

## More analysis

- TEPS' query id is used to track an SQL request to TEMS. In this case, TEPS' query id = **1023**.
- Request handle is very important because this one piece of information can glue information between TEMS and TEMA. In our example, TEMS request handle = **4277144246**.
- For short-term historical data query, TEMA gets data from the respective binary file. In this case, the short-term historical binary file is **WTLOGCLDSK**.
- Different stage of TEMS Proxy request is identified by its status. For example after data is sent by TEMA, TEMS Proxy request status is set to 7, which means data has arrived (`CTIRA_data_arrived`). List of Proxy request status is shown in reference slide.



## 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

