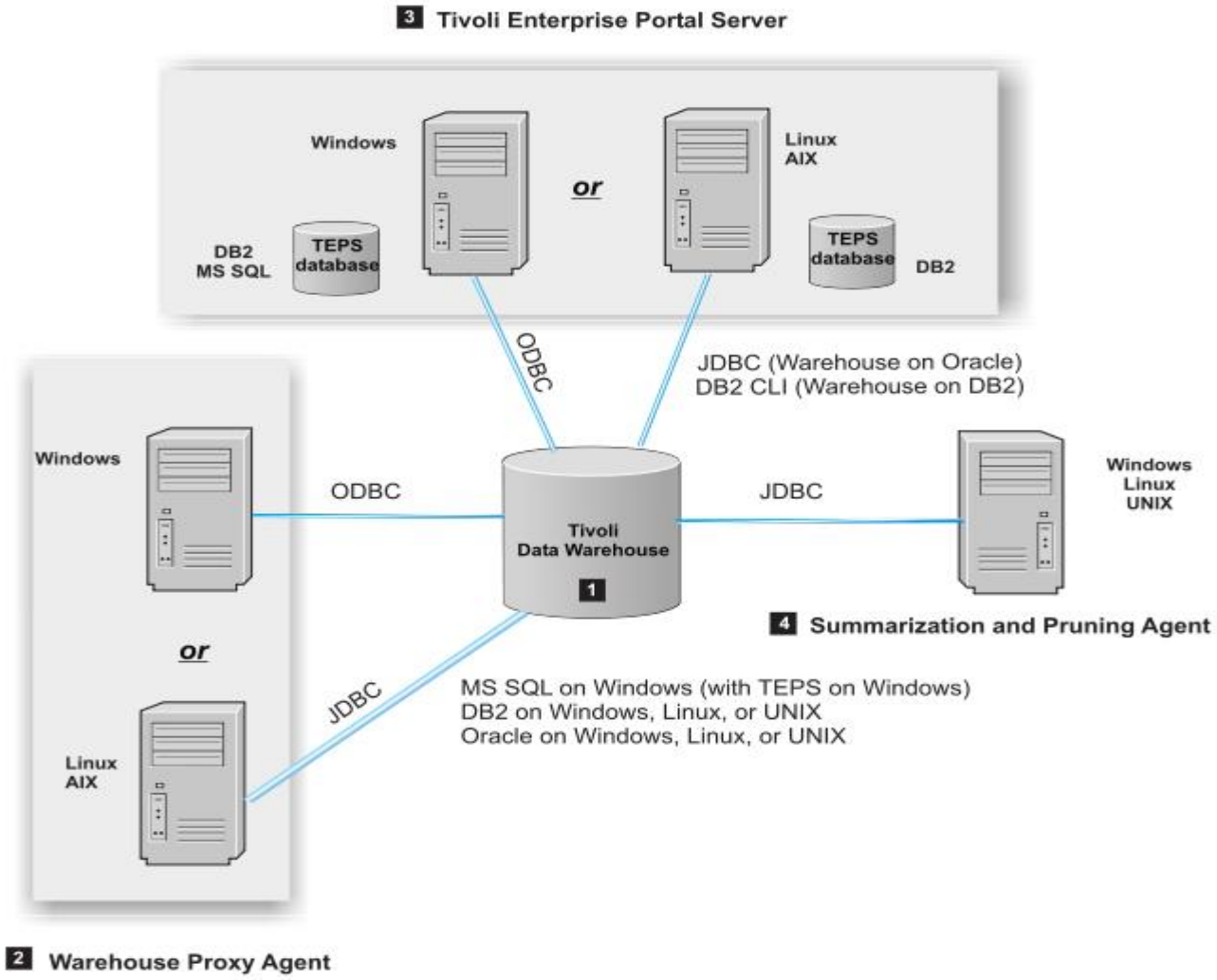
ITM 6.x Problem Determination and Troubleshooting

WPA - Overview

Tivoli software

ON DEMAND BUSINESS

Warehouse Solution- TDW



Historical Data Background

- ITM 6 provides two types of historical data
  - Short term - less than 24 hours old
    - Stored in a binary file on the agent system or on the TEMS system depending upon how user configured – Agent system is recommended
    - One binary file per attribute group
    - Binary file managed by TEMA framework
    - Persistent Data Store (PDS) used instead of file on z/OS
  - Long term – more than 24 hours old
    - Placed into TDW by WPA (Warehouse Proxy Agent) at interval specified by user – Warehouse Interval
  - Caveat's:
    - Historical data configuration allows data less than 24 hours old to be placed into TDW (Warehouse Interval = 1 hour)
    - Binary files will be pruned once data has been successfully inserted into TDW and it's greater than 24 hours old. If Warehouse Proxy does not run, data needs to be pruned by the user. Not true for PDS.

Historical Data Background (cont.)

- TDW storage requirement based on:
  - number of attribute groups configured
  - size of each attribute group
  - number of agents (instances of agents)
  - historical collection cycle (5, 15, 30, 60 min)
  - pruning and aggregation parameters
- Each agent install guide should have a documentation of the attribute group size
- Use the ITM/TDW Warehouse Load Projections spreadsheet on ISM website for estimating database requirements

https://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10TM1Y

Overview of Warehouse Proxy Agent

- Warehouse Proxy Agent (WPA) is responsible for offloading historical data collected by ITM 6 into the Tivoli Data Warehouse (TDW)
- WPA runs on:
  - Windows 2000, 2003 and 2008
  - AIX 5.2, 5.3 and 6.1
  - Linux Redhat and SUSE
  - Solaris V9, V10
- TDW supports the following Relational Databases:
  - DB2 V9.1, 9.5 and 9.7
  - Oracle 9.2
  - Oracle 10g Release 1 and 2
  - Oracle 11g Release 1 and 2
  - MS SQL Server 2000, 2005 and 2008

Note: For the exact list of platforms see product documentation and/or the certification matrix
http://www-306.ibm.com/software/sysmgmt/products/support/Tivoli_Supported_Platforms.html

Overview of Warehouse Proxy Agent (cont.)

- TDW database can be local or remote
- Can use multiple WPA's to achieve scalability
- Can use "Use Batch" feature to improve performance of WPA, particularly beneficial when TDW is remote.

ITM 6.x Problem Determination and Troubleshooting

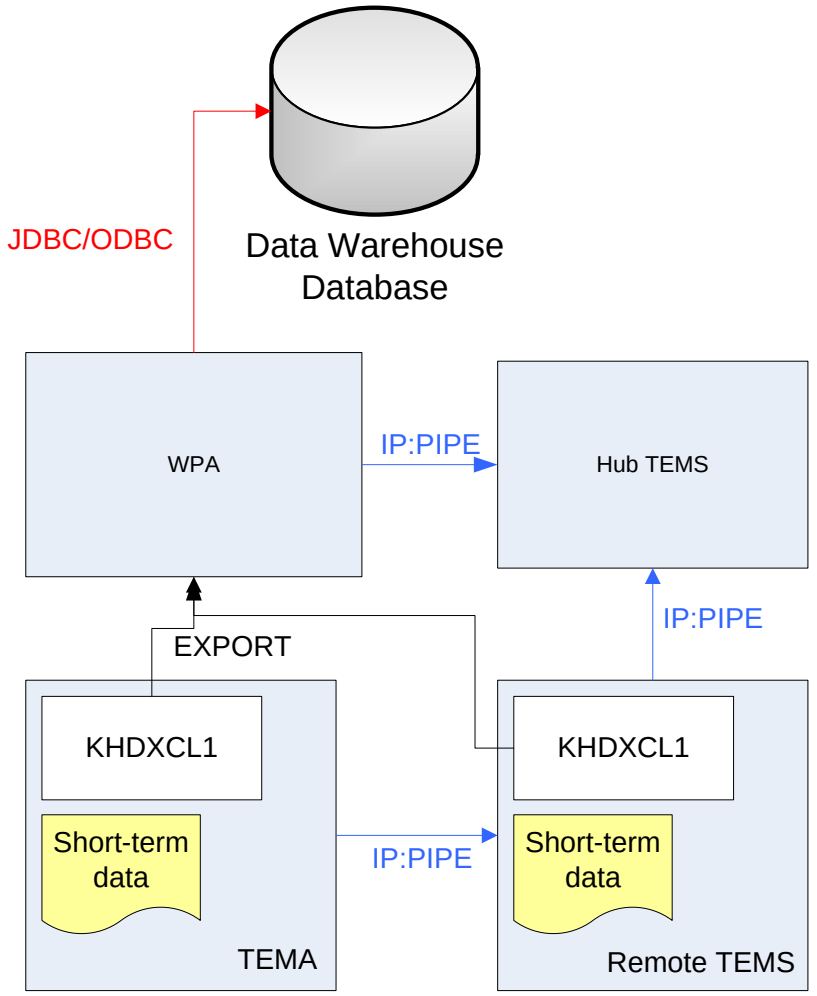
WPA – Data Flow

Tivoli software

ON DEMAND BUSINESS

WPA Data Flow

- When WPA starts, it will register its network address with Hub TEMS GLB.
- Collection on:
  - **TEMS:** check GLB during export time
  - **TEMA:** TEMS checks GLB every 60 mins then distributes it to all on-line TEMAs.
- Use `KPX_WAREHOUSE_REGCHK` variable to set the checking interval.
- Every collection time, new data is collected and stored in binary files.
- Every warehousing time, the export service (khdxcl1) use worker threads to export per batch 1000 rows (max) from each binary file.
- Use `KHD_EXPORT_THREADS` variable to set the number of worker threads. Default: 10.



WPA Data Flow (cont.)

- **WPA**
  - The export worker threads are then placed on a working queue. The maximum size of the queue is set to 1000: `KHD_QUEUE_LENGTH=1000`.
  - Each worker thread request an ODBC/JDBC connection from the connection pool. The connection pool is initialized when the Warehouse Proxy Agent is started. The default number of connection in the pool is set to 10: `KHD_CNX_POOL_SIZE=10`.
  - It is important to set the number of worker threads greater or equal to the number of ODBC/JDBC connections.
    `KHD_EXPORT_THREADS >= KHD_CNX_POOL_SIZE`.
  - In addition to the configurable environment variables mentioned above, the standard agent framework provides some control over the Warehouse Proxy Agent's scalability and performance profile. When the Warehouse Proxy Agent starts up, it initializes a number of RPC listeners .`CTIRA_NCSLISTEN=10`.

WPA Data Flow (cont.)

- **WPA**
  - If the export request has been successful meaning that all the data from this export has been inserted successfully in the database before the server timeout, a successful status is sent to the export service on the agent or TEMS.
  - If this is not the case, an unsuccessful status (with a status number indicating the reason of failure) is sent to the export service.
- **Agent or TEMS**
  - When receiving a successful export status, an entry is created in the marker file `khdexp.cfg` for the corresponding table and with the last WRITETIME timestamp inserted in the database.
  - If no status has been received before the client timeout, a new export request is sent.
  - Every successful export, all the data older than 24 hours in the binary file is pruned if the corresponding data has been successfully inserted in the database using the marker file `khdexp.cfg`.

Client/Server Timeout

- Client Time out
  - An export request will be resent if a status has not been received from the Warehouse Proxy before this timeout

    `KHD_STATUSTIMEOUT = 900s (15 min)`
- Server Time out
  - `KHD_SRV_STATUSTIMEOUT = 600s (10 min)`
  - An export will be rejected by the Warehouse Proxy Agent in 4 stages:
    - stage `END_QUEUE`
    - stage `START EXPORT`
    - stage `START SAMPLE`
    - stage `COMMIT EXPORT`
  - `KHD_SRV_STATUSTIMEOUT` should always be less than `KHD_STATUSTIMEOUT` by at least 60s.

Server Timeout Stages

- Stage `END_QUEUE`
  - if the time between the export was sent to the work queue and the time before it is extracted from the queue has exceeded the server timeout, the export request is rejected.
- Stage `START EXPORT`
  - if the time between the export was sent to the work queue and the time before it starts to do some existence checking in the database has exceeded the server timeout, the export request is rejected.
- Stage `START SAMPLE`
  - if the time between the export was sent to the work queue and the time before it fetches all the row in the sample has exceeded the server timeout, the export request is rejected.
- Stage `COMMIT EXPORT`
  - if the time between the export was sent to the work queue and the time before committing the rows in the database has exceeded server timeout, the export request is rejected.

Export Status

- Examples of export status (`khdxstat.h`)
  - 20 : ODBC Error
  - 26: Metafile Not Found
  - 27: Metafile IO Error
  - 30: History File Not Found
  - 49: RPC Error
  - 53: Server Died
  - 60: Init JVM Error
  - 61: Java Error
  - 62: JDBC Error
  - 73: Warehouse Proxy Not Registered
  - 216: Server Timeout
  - 218: Database Error

TMZDIFF, WRITETIME, EXPORTTIME

- **TMZDIFF**: it is the difference between the time zone where the agent is installed and the Universal Time (GMT), if the data is collected at the TEMA. The TMZDIFF is the difference between the time zone of the TEMS and the Universal Time (GMT). This value is shown in seconds, if the data is collected at the TEMS. This value is shown in seconds
  - Example CT is –6h from GMT , TMZDIFF is 6 hours, so in seconds: 3600*6*-1 = 21600 (a positive number).
  - If TEMA in France which is +1h from GMT, TMZDIFF= 3600* 1-1=-3600 (a negative number)
- **WRITETIME**: the time the record was written in the **binary file**. The format of this timestamp is a 16-character value in the format *cyymmddhhmmssttt*, where:
  - – c = century
  - – yymmdd = year, month, day
  - – hhmmssttt = hours, minutes, seconds, milliseconds
- **EXPORTTIME:** (column in the `WAREHOUSELOG` table) is the time the export was done for a batch of rows. The WPA does not keep a record of every row , only at the batch level.

WPA on Non-Windows Internals

- The differences of the WPA on Linux/Unix comparing to the WPA on Windows are only at the Database calls. The WPA on windows is using ODBC calls whereas the WPA on Unix is using **JDBC calls**.
- A new C++ class has been created: **CTJDBC** that is a child of the Database Class: CTDBCBase. This class contains all the JNI calls (C code calling JAVA code).
- A JAVA class called **KHDXJDBC** contains all the JDBC calls. This class is part of the package `khdxjdbc.jar`**.**
- At initialization the WPA on LINUX starts the JVM.Then each threads that starts an export will attach to this JVM and will connect to the database. Once the 10 threads (the default) have been successfully connected to the database, the connections will be reused. The WPA on LINUX uses a connection pool as well as the WPA on Windows.
- To run the WPA on linux on debug mode :

```
khdxjdbc server
```

- The WPA unix library is called : `libkhdxsrj1.so` (or `.a`).
- A new configuration panel has been created. The code is in khd.jar and is using the library `libkszutil.so` (or `.a`).

Warehouse Tables

- WAREHOUSEID
  - This table only exists if the database is DB2 or ORACLE. It contains all the table names or column names that have been converted from a long name to a short name to follow the database vendor specifications.

RECTYPE	TABLENAME	OBJECTNAME	COLUMNNAME	ATTRNAME
COL	WEBSVC	Web_Service	MSRAIOBUSG	Measured_Async_I/O_Bandwidth_Usage
COL	WEBSVC	Web_Service	TOTAAIOREQ	Total_Allowed_Async_I/O_Requests

  - Database Vendor restrictions

	MSSQL	Oracle	DB2
Max table length	128	30	128
Max column length	128	30	30

- UTF8TEST
  - This table only exists if the database is DB2 or ORACLE. It contains 2 characters (a French e with accent and a Japanese character). This table is used to check that the database has been created with a UTF8 encoding. If this is not the case, the initial connection to the database will fail

Warehouse Tables (cont.)

- WAREHOUSELOG
  - This table allows you to know how many exports succeeded, and how many failed because of an ODBC/JDBC error or a TIMEOUT issue.
  - It also gives you the timestamp at each important stage of the export.
  - Running this query :

```
SELECT * FROM WAREHOUSELOG where ROWINSERTED = 0;
```

Will tell you which agent (originnode) , which attribute group (Object) at what time (EXPORTTIME) got an error (ERROR MSG)

ORIGINNODE	OBJECT	STARTQUEUE	END QUEUE	START EXPORT	EXPORT TIME	ROWS INSERTED	ROWS RECEIVED	ROWS SKIPPED	START TIME	EXPORT TIME	ERROR MSG
Primary::box1: NT	NT_System	105082011500 00000	10508201 151000000	10508201152 000000	10508201153 000000	1000	1000	0	105081509 20000000	105081509200 00000	
Primary::box1: NT	NT_System	105082011540 00000	10508201 155000000	10508201156 2000000	10508201157 000000	0	1000	0	105081509 20000000	105081509200 00000	Sample data rejected for timeout reason at stage COMMIT EXPORT

Warehouse Tables (cont.)

- WAREHOUSELOG
- From ITM 6.2.3, Statements from the Warehouse Proxy Agent into the WAREHOUSELOG table are now disabled by default.
- Tivoli Data Warehouse log tables can grow very large and require regular pruning.
- The self-monitoring workspaces provided by the Warehouse Proxy Agent provide sufficient information to determine if agent is operating correctly
- To restore the old behavior, you must edit the Warehouse Proxy Agent configuration file and change the KHD_WHLOG_ENABLE variable.

ITM 6.x Problem Determination and Troubleshooting

WPA – Troubleshooting Techniques

Tivoli software

ON DEMAND BUSINESS

Troubleshooting

- Use TEP to see if agent had any problems
  - Check the Warehouse Proxy workspace which contains the Error table – situation will fire by default on any error
    - Some errors are environmental and will auto-correct, like data not exporting due some temporary problem or timeout
    - Database problems may require attention, especially inadequate database log size
- Check logs if not enough information in TEP
  - For Warehouse Proxy:
    - <hostname>_hd_*.log (and <hostname>_hd_java_*.log if running on UNIX/Linux)

Log and Trace files

- **WPA RAS1 log**
  - Name:
    - Windows

      `<hostname>_hd_nnnnnnnnnn.log`
    - Non-Windows

      `<hostname>_hd_nnnnnnnnnn.log`

      `<hostname>_hd_java_nnnnnnnnnn.log`
  - Locations:
    - Windows

      `%CANDLE_HOME%\logs`
    - Non-Windows

      `$CANDLEHOME/logs`
  - Contains trace statements from execution of WPA agent

Log and Trace files (cont.)

- **Warehouse Proxy Operation Log**
  - Name:
    - Windows

      `<hostname>_LG0` - most current log

      `<hostname>_LG1` - previous log
    - Non-Windows

      `<hostname>:Warehouse.LG0` - most current log

      `<hostname>:Warehouse.LG1` - previous log
  - Locations :
    - Windows

      `%CANDLE_HOME%\TMAITM6\logs`
    - Non-Windows

      `$CANDLEHOME/logs`
  - Contains an audit trail for each export written to the warehouse database

    ```
    1051025082559000KHD001 Inserted 212 rows from Primary:CCOOK1:NT into NT_Memory
    ```

General Trace Options

Warehouse Proxy functionality can be verified by enabling the trace setting (UNIT:khdxdbex OUTPUT) on the WPA. Not overly expensive, only writes 1 line per export

- Tracing with the option ALL is expensive and should only be enabled on request by support to diagnose problems:

```
ERROR (UNIT:khdx ALL) (UNIT:khdjava1 ALL)
```

- `(UNIT: khdx ALL)` will trace all the C++ functions.
- `(UNIT: khdjava1 ALL)` will trace all the Java calls, including all the binding values for each column and each row of each export.
- The maximum number of trace files is set to 5. The log files wrap after that.

Warehousing RAS1 Trace

- `ERROR (UNIT:khdx ST ERR)`
- `ERROR (UNIT:khdx ALL)`

- **Tracing warehouse export issues from TEMS or agents**:
  - Using common library KHDXCL1 in TEMS and agents for history warehouse exporting. Displays process of reading history files and transmitting history data to the Warehouse Proxy Agent.

- **Tracing Warehouse Proxy agent issues**:
  - Displays status and processing of all history data being inserted into the history warehouse database. Includes the SQL CREATE TABLE, INSERT INTO, and ALTER TABLE statements that are being used.

RAS1 Trace for binary file issues

- Short term history reports can be issued from TEP. A short-term history report is, by default, any time span less than 24 hours. This time interval can be modified in TEP.

- All short-term history reports are directed to the ITM history files (or binary file) at either the TEMS or the Agent, depending on how the history situation has been configured.

- When STH kept on TEMS:

```
ERROR (UNIT:kpxhsloc ALL) (UNIT:krabhsco ALL)
```

- When STH kept on Agents:

```
ERROR (UNIT:kra ALL)
```

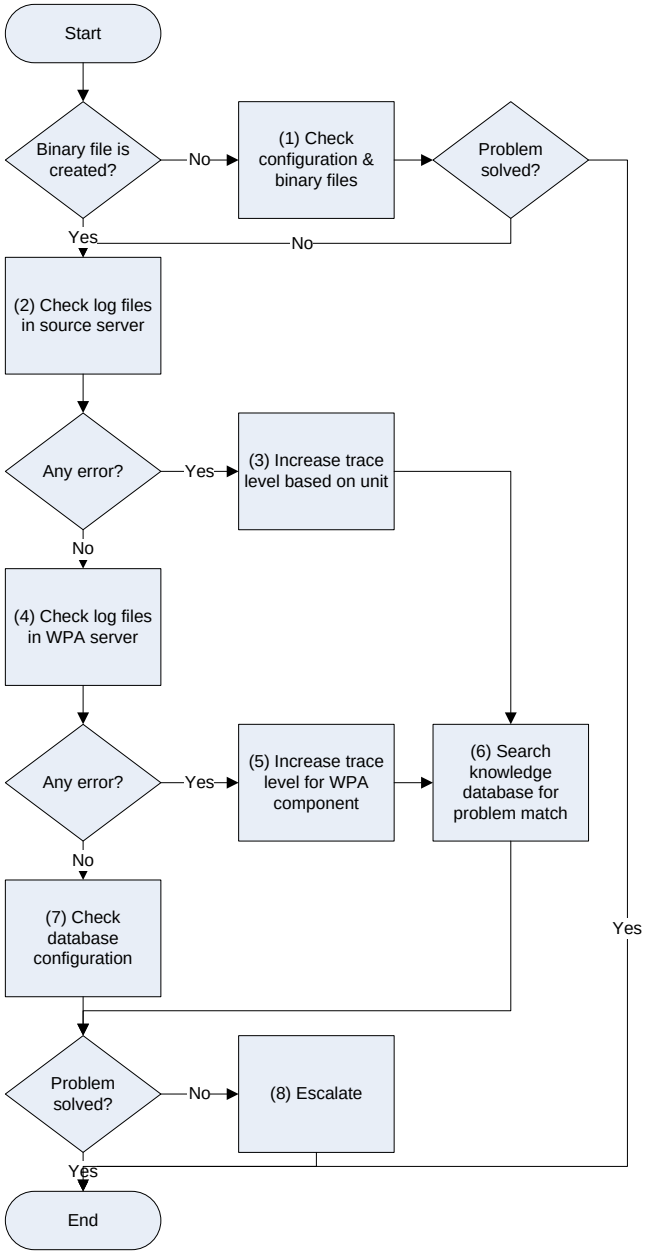
Environment variable to speed up export

- **KHD_EXPORT_DEBUG= Y**
  - Set this variable for any agent for which you want the export to occur at the collection interval (such as every 5 minutes) instead of at the warehouse load interval (such as every hour)
  - If you collect the data at the TEMS, set the variable at the TEMS

- **KPX_WAREHOUSE_REGCHK=5**
  - To allow the TEMS to check every 5 minutes the network address of the Warehouse Proxy instead of waiting at least 1 hour.

WPA Problem Determination Flow

- The purpose is to provide a logical flow in order to find out where the problem really is.
- There are 4 main components:
  - TEMA
  - Collection Server (TEMA or TEMS)
  - WPA
  - Database
- Each of component has different way to troubleshoot.
- It includes the network communication among those components.



Check Configuration & Binary Files

- Check whether new binary files have been created.
  - Windows:

    `%CANDLE_HOME%\tmaitm6\logs`
  - Non-Windows:

    `%CANDLEHOME/logs`
- If not, then step (1) is to check out the Warehouse configuration again.
- Alternatively, check **TSITDESC** table whether the corresponding **UADVISOR_*<name>*** has been correctly defined and started.
- If everything is okay, trace the historical collection process.
  - Collection on:
    - **TEMS:** `ERROR (UNIT:kpxhsloc ALL) (UNIT:krabhsco ALL)`
    - **TEMA:** `ERROR (UNIT:kra ALL)`
- Most of the time, the problem can be resolved by delete the existing binary files and redo the configuration.

Check Configuration & Binary Files (cont.)

- Sometimes the binary files get corrupted or one of the columns shows corrupted value. Use `krarloff` program to convert the binary file to flat file and read the content.
  - Command syntax (example):

    ```
    krarloff -h -d <delimiter> -o <output file> <binary file>
    ```

  - Example:

    ```
    krarloff -h -d ',' -o memory.txt wtmemory
    ```

  - `-h` will print the header. The header identifies the attribute column name.
- If the collection server is TEMS, check whether the network communication between TEMA and TEMS is okay.
- **Note:** Data in the binary file will not be lost if the Warehousing process fails. The binary file will just continue to grow.

Check Collection Server

- If the binary files have been created, step (2) is to check any error log in the collection server (TEMS or TEMA).
- Step (3) is to increase the trace level to find more information:

  ```
  ERROR (UNIT:khdx ST ERR)
  ```

- Using common library KHDXCL1 in TEMS and agents for history warehouse exporting. Displays process of reading history files and transmitting history data to the Warehouse Proxy Agent.
- `KHDEXP.CFG` is the file that the TEMA or TEMS uses to track what records in the binary file have been warehoused.
- `KHD_EXPORT_DEBUG = Y`
  - Set this variable for any agent for which you want the export to occur at the collection interval (such as every 5 minutes) instead of at the warehouse load interval (such as every hour)
  - If you collect the data at the TEMS, set the variable at the TEMS
- `KPX_WAREHOUSE_REGCHK=5`
  - To allow the TEMS to check every 5 minutes the network address of the Warehouse Proxy instead of waiting at least 1 hour.

Check Collection Server (cont.)

- Multiple WPA can be used. To find out about which WPA that TEMA or TEMS should connect to, increase the trace level: (`UNIT:kprwhpx STATE`). This will print to the TEMS RAS1 log the WPA that is being used.
- Check also whether the network communication between TEMA or TEMS to WPA is okay.
- Once the error message is identified, step (6) is to look for the same message in problem database.

Sample Messages in the Exporter Logs

- `(3EDCC183.231B6D40-198:khdxbase.cpp,257,"setError") Error "Unable to locate Candle Export Server"`

  could not establish the current WHP agent location or address

- `(3EDCC183.231B6D40-198:khdxdacl.cpp,601,"resolveServerAddress") Warehouse proxy not registered`

  could not establish the current WHP agent location or address

- `(3EDCC183.231B6D40-198:khdxdacl.cpp,445,"exportData") Export for object <TCP_Statistics>  failed, Status = 73`

  could not establish the current WHP agent location or address

  73 : Warehouse Proxy not registered

- `Export Request  for table <NT_Process>  failed with status 53`

  The WHP agent died

- `Export Request  for table <NT_Memory>  failed with status 216`

  The WHP agent timeout during an export (see foil on Server Timeout)

Check WPA

- If no error in collection server, move on to WPA and do step (4) to check whether any error in the WPA log files.
- Step (5) is to increase the trace level to find more information:

  `ERROR (UNIT:khdx ST ERR)`

- Displays status and processing of all history data being inserted into the history warehouse database. Includes the SQL CREATE TABLE, INSERT INTO, and ALTER TABLE statements that are being used.
- Since FP2, multiple WPA can be used. To find which TEMS served by WPA, increase trace level to: (`UNIT:khdxrpcr STATE`). This will cause the value for `KHD_WAREHOUSE_TEMS_LIST` to be printed to the RAS1 log.
- Once the error message is identified, step (6) is to look for the same message in problem database.

Check Database

- Step (7) is to check the Data Warehouse database. It includes communication and configuration.
- For JDBC, increase the trace level:

```
(UNIT:khdjava1 ALL)
```

- It will trace all the Java calls, including all the binding values for each column and each row of each export.
- If the source the problem can be found and it still can't be resolved, the problem can now be escalated, step (8), with all the information.

Test when WPA Service is Started

- Check that the WPA can connect to the database.
- Check, if the database is Oracle or DB2, that the encoding is set to UTF8
- Check, if the database is DB2, that a buffer pool of page size 8k is created and if not create one as well as 3 new table spaces using the 8k buffer pool.
  - The buffer pool is called: ITMBUF8K
- The 3 table spaces are called: ITMREG8K, ITMSYS8K, ITMBUF8k
- Creates a database cache that contains a list of all the tables and columns that exist in the database
- If any one of these tests fails, a message will be written in the log if the class ERROR is set for the trace. Messages will also appear in the Event Viewer.

Test when WPA Service is Started (cont.)

- The tests to the database will be retried until success every 10 minutes. There are 2 environment variables that can change this default setup:
  - `KHD_CNX_WAIT_ENABLE` is set to Y by default. It permits to wait before a retry. Changing this variable to N will allow to not wait before retries. Be careful when setting this variable to N as you can easily generate a huge log file if the tests to the database fail at each retry.
  - `KHD_CNX_WAIT` is the time in minutes to wait before trying to reconnect. Default is 10 minutes.

ITM 6 Problem Determination and Troubleshooting

WPA – Special/Unique Situation

Tivoli software

ON DEMAND BUSINESS

Database connection issues - 1 : JDBC drivers

- Check the java trace file

```
== 1 t=main java.lang.ClassNotFoundException: com.ibm.db2.jcc.DB2Driver
    at java.net.URLClassLoader.findClass(URLClassLoader.java:375)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:562)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:442)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:494)
    at java.lang.Class.forName1(Native Method)
    at java.lang.Class.forName(Class.java:180)
    at com.tivoli.twh.khd.khdxjdbc.initJDBC(khdxjdbc.java:110)
== 2 t=main initJDBC : com.ibm.db2.jcc.DB2Driver
```

- This indicates that the “correct” JDBC drivers have not been added and saved successfully in the Configuration Panel. Most of the time the Add button has not been used.
- The added JDBC drivers can be seen in the `hd.ini` file in the `KHD_WAREHOUSE_JARS` and `KHD_CLASSPATH` variables.

```
KHD_WAREHOUSE_JARS=/opt/IBM/db2/V8.1/java/db2jcc.jar,/opt/IBM/db2/V8.1/java/d
   b2jcc_license_cu.jar
KHD_CLASSPATH=$CANDLEHOME$/$BINARCH$/bin/khdxjdbc.jar:/usr/opt/db2_08_01/java
   /db2jcc.jar:/usr/opt/db2_08_01/java/db2jcc_license_cu.jar:/opt/IBM/db2/V8.
   1/java/db2jcc.jar:/opt/IBM/db2/V8.1/java/db2jcc_license_cu.jar
```

Database connection issues - 2 : use another tool

- Test the database connection with another tool. For instance, Squirrel can be used to connect to any database vendor using any JDBC drivers.
  - *http://squirrel-sql.sourceforge.net/*
- Some security issues, user privileges can be found easily this way.

Database connection issues - 3 : socket errors

- Example of error that will happen even with Squirrel.
- Error in the C trace:

```
(448E63CD.0000-1:khdxjdbc.cpp,3851,"processJavaException") Exception message:
   com.tivoli.twh.khd.KHDException
+448E63CD.0000  at
   com.tivoli.twh.khd.khdxjdbc.getDbConnection(khdxjdbc.java:261)
(448E63CD.0001-1:khdxjdbc.cpp,3853,"processJavaException") Exception message:
   com.tivoli.twh.khd.KHDException
+448E63CD.0001  at
   com.tivoli.twh.khd.khdxjdbc.getDbConnection(khdxjdbc.java:261)
(448E63CD.0002-1:khdxdbb.cpp,599,"initializeDatabase") Connection with
   Datasource "jdbc:db2://localhost:60000/WAREHOUS" failed
```

- Error in the Configuration Panel/Agent Parameter, after clicking on the button "Test database connection“:

```
"Database connection failed. IO Exception Socket to server localhost on port
   60000.
Rmk: this error will be in the java trace file with defect 30987
```

Database connection issues - 4 : listener not active

- **Message:** `IO Exception Socket Error: the listener is not active`
- Solution for DB2:
  - `db2set -i <instance> DB2COMM=tcpip`

    If the instance name is db2inst1, then the command would be:

    `db2set -i db2inst1 DB2COMM=tcpip`
  - `db2 update dbm cfg using SVCENAME <port>`

    If the port is 60000, then the command would be:

    `db2 update dbm cfg using SVCENAME 60000`
  - Check the port used in the file `/etc/services` you should find:

    `DB2_db2inst1 60000/tcp`
  - `db2stop`
  - `db2start`

Database connection issues - 5 : getDbConnection

- `Database Connection failed.Error Opening socket to server localhost/127.0.0.1 on port 60000 with message: connection refused DB2Connection Correlator`

  Solution for DB2:
  DB2 is not started, use the following command:
  ```
  db2start
  ```

- `Database Connection failed. The application server rejected establishment of the connection. An attempt was made to access a database, WAREHOUS, which was not found:`

  Solution for DB2:
  The WAREHOUS database has not been created.
  Use the following command to create it:
  ```
  db2 create database Warehous using codeset utf-8 territory US
  ```

Database connection issues - 6 : getDbConnection

Possible Exceptions when using the Test Database Connection button in the Configuration panel:

- `Database Connection failed.Connection authorization failure occurred. Reason: password invalid`

Solution for DB2:

Check that the userid/pw selected has been created and can connect to the database.

```
db2 connect to WAREHOUS user <userid> using <pwd>
```

Database cnx issues - 7 : khdxjdbc.java

C trace:

```
(44903959.0002-1:khdxjdbc.cpp,2302,"SQL_Tables") Calling SQLTables
    using owner <CAT>
(44903959.0003-1:khdxjdbc.cpp,3851,"processJavaException") Exception
    message: com.tivoli.twh.khd.KHDException
+44903959.0003    at
    com.tivoli.twh.khd.khdxjdbc.execute(khdxjdbc.java:797)
(44903959.0004-1:khdxjdbc.cpp,3853,"processJavaException") Exception
    message: com.tivoli.twh.khd.KHDException
+44903959.0004    at
    com.tivoli.twh.khd.khdxjdbc.execute(khdxjdbc.java:797)
(44903959.0005-1:khdxdbb.cpp,599,"initializeDatabase") Connection
    with Datasource "jdbc:db2://localhost:60000/ITMDW4" failed
```

Database cnx issues - 8 : Cannot Create BufferPool

Java trace:

```
== 70 t=main for statement: CREATE BUFFERPOOL ITMBUF8K IMMEDIATE  SIZE 250
    PAGESIZE 8 K
== 71 t=main com.ibm.db2.jcc.a.SqlException: DB2 SQL error: SQLCODE: -552,
    SQLSTATE: 42502, SQLERRMC: CAT;CREATE BUFFERPOOL
 at com.ibm.db2.jcc.a.rf.d(rf.java:1396)
 at com.ibm.db2.jcc.b.jb.l(jb.java:356)
 at com.ibm.db2.jcc.b.jb.a(jb.java:64)
 at com.ibm.db2.jcc.b.w.a(w.java:48)
 at com.ibm.db2.jcc.b.dc.c(dc.java:312)
 at com.ibm.db2.jcc.a.sf.cb(sf.java:1695)
 at com.ibm.db2.jcc.a.sf.d(sf.java:2287)
 at com.ibm.db2.jcc.a.sf.Z(sf.java:1298)
 at com.ibm.db2.jcc.a.sf.execute(sf.java:1282)
 at com.tivoli.twh.khd.khdxjdbc.execute(khdxjdbc.java:791)
```

Solution for DB2

- The user that connects to the database must have privileges to create a buffer pool and 3 table spaces.
- You can add the user as a member of DB2ADMIN group.
- To find this group , you can run this command:

```
db2 get dbm cfg | grep SYSADM
```

Database cnx issues - 9 : Not UTF8 encoding

C trace errors:

```
(44904608.0000-1:khdxbase.cpp,250,"setError") Error 201/3/0(00000000)/0
(44904608.0001-1:khdxbase.cpp,266,"setError") Error "Warehouse Database
   server encoding is not UTF8.You need to create a UTF8 database.
+44904608.0001 "
(44904608.0002-1:khdxdbb.cpp,558,"initializeDatabase") Database server
   encoding is not UTF8.You need to create a UTF8 database.
(44904608.0003-1:khdxdbb.cpp,599,"initializeDatabase") Connection with
   Datasource "jdbc:db2://localhost:60000/ITMDW5" failed
(44904608.0004-1:khdxdbb.cpp,602,"initializeDatabase") disconnectDatasource
   failed
(44904608.0005-1:khdxsrvc.cpp,719,"testDatabaseConnection")
   testDatabaseConnection failed
(44904608.0006-1:khdxsrvc.cpp,650,"setupExportServer") A retry will be
   attempted in 10 minute(s).
(44904608.0007-1:khdxsrvc.cpp,651,"setupExportServer") Please check the
   troubleshooting guide for any error not described above.
```

Solution for DB2

Drop the warehouse database and recreate a new one with UTF8 encoding:

```
db2 create database Warehous using codeset utf-8 territory US
```

JDBC errors - 1 : Transaction Log Full

C trace errors:

```
06.153 03.22.41 {8}hdxdbex.cpp,392,"startSession") Connection with Datasource
    "ITM Warehouse" successful
(06.153 03.23.42 {81}hdxbase.cpp,250,"setError") Error 219/3/-964(FFFFFC3C)/0
    executing SQLExecute
(06.153 03.23.42 {81}hdxbase.cpp,266,"setError") Error "[IBM][CLI
    Driver][DB2/NT] SQL0964C  The transaction log for the database is full.
    SQLSTATE=57011
+447FE77E.0001 "
```

Solution for DB2

you need to increase the transaction log files.

For example:

- `update db cfg for warehous using LOGFILSIZ 8192;`
- `update db cfg for warehous using LOGPRIMARY 12;`

JDBC errors - 2 : Deadlock

Solution for DB2
You need to increase the lock list
For example:

```
update db cfg for warehous using LOCKLIST 1500;
```

Also prior to FP4, 2 indexes on the WAREHOUSEID were created separately , one by the WPA and one by the S&P, but both are needed. When the S&P was not started, we encountered deadlock due to a table scan on the WAREHOUSEID. The workaround prior to FP4 was to start the S&P agent that created the missing index.

Examples of error in the Event Log - 1

- **Status 31**

```
Exporter Error testDatabaseConnection failed.

Exporter Error 98/3/0(00000000)/0. Native Error Message "Userid
  decode failed. status = 31.  "
```

The WHP agent could not decode the entries in the registry

- **Error 20**

```
Exporter Error testDatabaseConnection failed. Exporter Error
20/3/0(00000000)/0. Exporter Error reported by khdxodbc.cpp at 4084
in ODBCError
```

The WHP agent could not connect to the database. An ODBC error occurred.

- **Error 49**

```
Exporter Error reported by khdxrpcs.cpp at 1650 in postStatus

Exporter Error 49/3/49(00000031)/0 executing KHD_PostStatus. Native
Error Message RPC Error to    node "Primary:CCOOK1:NT""
```

The WHP agent could not return the export status back to the client, agent or TEMS, that sent the data

Examples of error in the Event Log - 2

- **DB2 priviledges**

  Exporter Error testDatabaseConnection failed.

  ```
  Exporter Error 0/3/-552(FFFFFDD8)/0 executing SQLExecute. Native
  Error Message "[IBM][CLI Driver][DB2/NT] SQL0552N  "ITMUSER" does not
  have the privilege to perform operation "CREATE BUFFERPOOL".
  SQLSTATE=42502
  ```

  The WHP agent connected to the warehouse database cannot create a DB2 Bufferpool, and 3 DB2 tablespaces. The user (here ITMUser) does not have enough privileges.

  When the ODBC data source, DB2 database and DB2 user are created by the configuration panel, the user is normally created as a member of the Windows OS Administrator group. On some platform such as Windows XP, this may not happen. To solve this issue, add the OS ITMUser to the Administrator group or to the DB2ADMNS windows group.

  Also when the DB2 database is remote, the user on the remote system should be a member of DB2ADMIN group.

  To find this group , you can run this command:

  ```
  db2 get dbm cfg | grep SYSADM
  ```

Examples of error in the Event Log - 3

- **SQLTables error**

```
Exporter Error testDatabaseConnection failed.

Exporter Error 0/3/-552(FFFFFDD8)/0 executing SQLExecute. Error
20/3/-443(FFFFFE45)/0 executing SQLTables
```

An error may occur with the SQLTables if the database was created before applying IBM DB2 FixPack 10. You encounter an SQL0443N error if you run a DB2 Call Level Interface (CLI) catalog function (such as SQLTables(), SQLColumns(), or SQLStatistics().

Run the bind routine db2schema.bnd file against each database to resolve this error

The following examples shows the rebinding commands from the `c:\SQLLIB\bnd>DB2` directory:

```
db2connect to <warehouse database name>

c:\SQLLIB\bnd>DB2 bind db2schema.bnd blocking all grant public
```

Examples of error in the Event Log - 4

- **Bind error**

Oracle error:

```
[Oracle][ODBC][Ora]ORA-01461: can bind a LONG value only for insert into
a LONG column
```

DB2 error:

```
Native Error Message "[IBM][CLI Driver][DB2/NT] SQL0302N  The value of a
host variable in the EXECUTE or OPEN statement is too large for its
corresponding use.SQLSTATE=22001
```

An bind error may occur if the data that needs to be inserted is not converted correctly to UTF8 encoding. Most of the time this problem happen when we forgot to set the system environment variable to support the UTF8 encoding for Oracle and DB2.

- DB2: `DB2CODEPAGE=1208`
- ORACLE: `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`

As any system environment variable, you will need to reboot the machine to be sure it is valid.