A Simple guide to Transaction Processing

<table>
<thead>
<tr>
<th>Technology Demographic Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
<tr>
<td><strong>Platform/OS</strong></td>
</tr>
<tr>
<td><strong>Industry</strong></td>
</tr>
<tr>
<td><strong>Application</strong></td>
</tr>
<tr>
<td><strong>Database</strong></td>
</tr>
<tr>
<td><strong>Keywords</strong></td>
</tr>
<tr>
<td><strong>Date</strong></td>
</tr>
<tr>
<td>OneWorld</td>
</tr>
<tr>
<td>B733.x</td>
</tr>
<tr>
<td>All</td>
</tr>
<tr>
<td>All</td>
</tr>
<tr>
<td>All</td>
</tr>
<tr>
<td>All</td>
</tr>
<tr>
<td>Strategic, OneWorld, B9</td>
</tr>
<tr>
<td>July 2000</td>
</tr>
</tbody>
</table>
Disclaimer

All information contained in this document should be treated as a hypothetical project plan. It is often the case that with upgrades to production and development objects, that there are issues that will be raised. This will therefore dramatically increase project timelines.

None of the entries in this document are in any way a replacement for the JDEdwards OneWorld Xe Upgrade Guide – instead, this document should be treated as a complement.

Overview

This document is release independent - ie, I will attempt to describe Transaction Processing. The latter parts of this email are SAR posts directly affecting pre-B733 and post-B733 (including Xe) versions - and lastly I attach a list of applications in B7.3.3.1 that were Transaction Processing activated.

Note that Transaction Processing is now called "Lock Manager" (since B7.3.3) and has absolutely NOTHING to do with row-level database locking!

Chapter 1 - What is it?

Definitions and Abbreviations used throughout this document

- **TMS (Transaction Management Server)** - the server that is used to serve as a central "checkpoint" for all database transactions.

- **TP (Transaction Processing) Monitor** - the process that runs on the TMS that is capable of timestamping database record changes and notifying the user if he/she has retrieved a record (and left it unchanged) that has since been changed by another user.

- **TS (Time Stamp)** - A method of comparing multiple copies of a database record to determine if what the user currently sees reflects the most recently updated version of the record. If it is not the most current, a message is sent to the user with the obsolete copy of the record that they need to fetch the record again (to get a more current version). The TMS runs TP, which provides the time stamp (TS). The client or workstation requests the time stamp, if specified by the jde.ini file.

- **JDEIPC (JDE Inter-Process Connect)** - A setting in the enterprise server's jde.ini that tells the system which value (process ID) to start with when JDE processes are started. When startIPCKeyValue=xxxx is commented out, the processes start with some pre-determined, hard-coded number (usually 5000 or 5001). If the number(s) conflict with something else running on the server (competing for PIDs), the process(es) will not run properly (usually, they don't start at all). If startIPCKeyValue=xxxx is commented out, and the user starts any JDE processes on the server that seem to "vanish into thin air" or not start at all, it is a good idea to activate the value (remove the ;) and set the number (xxxx) to a different value such as 7001.
What is Transaction Processing?

1. A transaction is a logical unit of work, comprised of one or more SQL statements performed on any number of databases.

2. All database operations in the transaction are executed or none are executed at all.

3. This guarantees that the contents of the database always remain in a consistent state (data integrity).

4. In effect, Transaction Processing requires that a BUSINESS DOCUMENT (i.e., a Sales Order) be fulfilled start to finish - if a problem occurs in the document - then NONE of the data is committed.

5. Note that Transaction Processing can also be extended to BUSINESS FUNCTIONS - a classic functional example is the Sales Order Entry process.


How does OneWorld use Transaction Processing?

7. Built into the Database Middleware is a Transaction Monitor.
   a. Commit - To finalize the transaction against business data.
   b. Rollback - To "erase" the transaction - i.e., no data is affected.
   c. Two Phase Commit - used when the database is stored in multiple locations (i.e., multiple database servers).

   a. Auto commit - a transaction is finalized automatically.
   b. Manual commit - No transactions are performed until a manual "commit" command is sent.

Process flow for Transaction Processing

When a user starts OneWorld, the local JDE.INI is checked to see if the user is requesting any services from the TMS. If the user is NOT requesting any services from the TMS, nothing happens. However, if the user is requesting services, the client machine sends a message to the TMS that it is requesting a TS service. The TMS then checks its JDE.INI (on the server) to see if it has any services available. If it does, it sends a message back to the client machine indicating that it is ready to accept TS requests from the client. If the TMS has no services available, it sends a message back to the client machine indicating that the TMS is not accepting any TS requests from clients. Thus, from that point on the client will NOT communicate with the TMS server at all.

If the server is setup such that there AvailableServices=NONE, this means that the TMS server is "turned off." This means that there will be no further communication between the client and...
the TMS server. If the client is set up such that there are no services requested, this means that
the client will not attempt to make any communication with the TMS at all.

**IMPORTANT NOTE**

It is important to almost TOTALLY ignore transaction processing until one has almost totally
completed the conference room pilot stage of their implementation. This is extremely vital
since if TP is turned ON at the beginning of an implementation - the changes made to
applications throughout the implementation will adversely affect how transactions are created.
Therefore, ensure that TP is a stage of the pilot towards the very end (ie - prior to "stress test"
or "load testing" environment prior to go-live).

**Chapter 2 - Configuring it**

The following are the JDE.INI file settings for the client and the server respectively:

**JDE.INI - Workstation (Pre B733)**

```ini
[TP MONITOR ENVIRONMENT]
Status=ON
LogPath=\b7\data
LogStatements=NO
LogBufferSize=1000
RequestedService=TS *
Server=hp9000a *
ServerTimeout=60 *

(* Specific to the TimeStamp function of the TMS.)
```

The following is an explanation for these settings:

- **Status** - Shows whether the TP Monitor is "ON" or "OFF". This setting is obsolete as of
  B73.2.1. (no matter what this setting is, TP is always ON!)

- **LogPath** - Where should TP write its log (generally you will want it to go to wherever
  the jde.log and jdedebug.log go)

- **Log Statements** - Options are YES or NO. If you choose YES, debug-type statements are
  written to the TP/TMS log. Such statements are difficult to interpret for anyone besides
  the TP/TMS programmers/designers. Therefore, it is not usually necessary to have it on.

- **LogBufferSize** - Default is usually 1000. It is only necessary to increase this when you
  choose YES for LogStatements, as your performance will likely be affected.

- **RequestedService** - This is the service that the client is requesting from the server. The
  applicable values are:

  - **TS** - this means that the client is requesting timestamps from the server.
  - **NONE** - this means that the client is not requesting any services from the TMS. This
    means that the client will not even attempt to communicate with the TMS.

- **Server** - This is the machine on which the TMS is located. Please note that it does not
  necessarily have to be the same machine that other OneWorld services are running on.
  If there are multiple platforms and/or Enterprise Servers in use, the customer must
  choose one on which to run TMS, and have all jde.ini files for all clients and servers
  point to that one server. This parameter only refers to the server the TMS is running on.
  The only way to ensure TS will work properly is to make sure all transactions are going
  through one "central checkpoint".
ServerTimeout - This is the time after which to stop attempting to connect to the TMS server. If no response is received after this time, the client will give up trying to connect to the TMS. The client first tries to communicate with the TMS. If no response is obtained within 10 seconds, a message box is displayed which says "Waiting for transaction Server". It then uses the value in this ServerTimeout setting to wait for the TMS to respond. If the server does not respond within the given Server Timeout period, the user cannot log into OneWorld. If this happens, check that the machine specified in the Server setting is valid. If the machine is valid, check that the port that you are connecting to is the same port that the TMS is running on.

JDE.INI - Server (Pre B733)

[TP MONITOR ENVIRONMENT]
Status=ON
LogPath=/usr/oneworld/b732
LogStatements=YES
LogBufferSize=1000
RequestedService=NONE (for performance reasons!!)
Server=hp9000a
ServerTimeout=60
AvailableService=TS
RegistryCleanupInterval=90
RegistryRecordLifeSpan=60
LogServices=1

The RequestedService, Server, and ServerTimeout lines of this TMS section are similar to those on the client machine. This is because the same machine hosting the TMS can have a process that is running on it that is a "client" to the TMS. This is the case when you launch UBEs on the server. Therefore before that job executes, it looks at the first 3 lines of this JDE.INI just like workstations do when a user starts OneWorld. Please refer to the above section JDE.INI - Workstation for a detailed description of these first three lines.

The last four lines of this server JDE.INI section are relevant to the TM server (TMS) only. The following is an explanation for the meanings of these settings:

AvailableService - this is the service that the server is providing to the clients. The possible values are:

TS - this means that the server is providing Timestamps to the clients (when the client requests the service).
NONE - this means that the server is not providing any services to the client (or to the server, if applicable) at all. When this is set, the client will communicate with the server once, then will be informed that the server is not providing any services at all.
RegistryCleanupInterval - This is the time (in minutes) after which the TMS cleans up its internal caches. This is done so that the memory does not get committed with data that is no longer relevant. This setting should not be an issue once the TMS starts running. Issues with regard to this setting have to be dealt with by the system administrator as they involve the use of memory on the machine.
RegistryRecordLifeSpan - This is the time (in minutes) which a record should stay in the TMS caches before it is purged. This means that at the time indicated by the value for RegistryCleanupInterval all records in the TMS internal caches for a time that is longer
than the value indicated for the RegistryRecordLifeSpan setting will be removed from the cache. This setting should not be really an issue once the TMS starts running. Issues with regard to this setting have to be dealt with by the system administrator as they involve the use of memory on the machine.

LogServices - This is a special setting that turns on the Trace Log for the TMS. This log is used to supplement the JDE.LOG. It is more detailed than the JDE.LOG, but is a complicated log that requires a strong familiarity with the internal workings of the TMS. Thus, it is recommended that this setting be used only when all possible debugging methods have been exhausted. It should be shipped out with a value of 0.

The possible values are:

- this means that tracing for TMS is turned ON
- this means that tracing for TMS is turned OFF

Changes to JDE.INI post B.733

The Transaction Processing settings in the TP MONITOR ENVIRONMENT section of the JDE.INI have often been perceived as very confusing. The majority of the settings are now obsolete and can be eliminated.

These changes are effective with OneWorld version B733.

TP Monitor replaced with Lock Manager

The TP MONITOR ENVIRONMENT section will be renamed to LOCK MANAGER.

A number of settings will be eliminated from the LOCK MANAGER section. These settings will either be assigned default values internally or become obsolete.

<table>
<thead>
<tr>
<th>Existing Setting</th>
<th>How Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Obsolete, always ON</td>
</tr>
<tr>
<td>LogPath</td>
<td>Assigned base directory from JDE.ini</td>
</tr>
<tr>
<td>LogStatements</td>
<td>Obsolete, always LOG</td>
</tr>
<tr>
<td>LogBuggerSize</td>
<td>Internal default</td>
</tr>
<tr>
<td>DisplayServerErrorMsg</td>
<td>Always display server error messages</td>
</tr>
<tr>
<td>ServerRetryInterval</td>
<td>Internal default</td>
</tr>
<tr>
<td>RegistryCleanupInterval</td>
<td>Internal default</td>
</tr>
<tr>
<td>RegistryRecordLifeSpan</td>
<td>Internal default</td>
</tr>
<tr>
<td>Server Timeout</td>
<td>Internal default</td>
</tr>
</tbody>
</table>

Three settings are preserved from the old TP MONITOR ENVIRONMENT with simplified values.

RequestedService

This setting indicates the type of service that the client requests from the server. The service that is currently being provided by servers is time stampsing (TS) only.

AvailableService
This setting indicates the service that the lock manager server is offering. It is also used to indicate whether the lock manager server is on or off. This setting is now either TS or NONE.

Server

This is a client only setting which indicates the lock manager server to be used to process records. The value for this setting is the same as the existing TP MONITOR ENVIRONMENT - Server setting, the name of the server acting as the lock manager.

NOTE : TP Monitor Environment concurrently supported

The TP MONITOR ENVIRONMENT section setting will also be supported for a transition period of undetermined time. If the new LOCK MANAGER section does not exist, a check is made for settings in the old TP MONITOR ENVIRONMENT section. Failure only occurs if both do not exist. If both exist, LOCK MANAGER settings have priority.

Chapter 3 - Using it

The next step is to correctly identify what, if any, transactions require to have transaction processing turned on. TP is application-level dependant - ie, the developer of an application can CHOOSE whether or not to use Transaction Processing. Of course, a number of standard OW applications have TP built into them (please see 3rd part of this email).

The next stage is to turn TP on both on the Server and on the Workstation. Under B7.32 this had a varying number of JDE.INI settings - that were simplified under B7.33 onwards as the [LOCK MANAGER] setting. Appendix 1 contains the settings for Transaction Monitor (pre-B733) and Appendix 2 contains settings for Lock Manager (post B733)

Simple Record Changed

To ensure that 2 users do not update the same piece of data - a Record Change Detection setting exists in the JDE.INI. This is normally turned ON as standard:

1. The change from the first user will update the record
2. When the 2nd user attempts to make a change to the same record, he will be notified in a simple message and be forced to reinquire on the record. All changes that they have made to the record will be lost
## Transaction Processing Scenario's

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Form1</th>
<th>Form2</th>
<th>Interconnect</th>
<th>Transaction Processing type used</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TP Off</td>
<td>TP Off</td>
<td>TP Off</td>
<td>All forms use &quot;Auto-commit&quot; ie commits whenever</td>
</tr>
<tr>
<td>B</td>
<td>TP Off</td>
<td>TP Off</td>
<td>TP On</td>
<td>Because neither form uses Manual Commit - the interconnect is ignored. Auto-commit is used</td>
</tr>
<tr>
<td>C</td>
<td>TP On</td>
<td>TP Off</td>
<td>TP Off</td>
<td>Form1 uses Manual Commit and Form2 uses Autocommit. The transaction boundary is not extended to include Form2</td>
</tr>
<tr>
<td>D</td>
<td>TP On</td>
<td>TP Off</td>
<td>TP On</td>
<td>Even though the TP setting is OFF in Form2, the interconnect extends the transaction boundary to include Form2. Therefore Form2’s TP off is overridden</td>
</tr>
<tr>
<td>E</td>
<td>TP Off</td>
<td>TP On</td>
<td>TP Off</td>
<td>Form1 and Form2 operate independently - Form1 operates using Auto-commit and Form2 operates as a transaction boundary</td>
</tr>
<tr>
<td>F</td>
<td>TP Off</td>
<td>TP On</td>
<td>TP On</td>
<td>Form1 and Form2 still operate independently - the interconnect is ignored and Form1 runs Autocommit and Form2 operates in its own transaction boundary</td>
</tr>
<tr>
<td>G</td>
<td>TP On</td>
<td>TP On</td>
<td>TP Off</td>
<td>Each form operates in Manual commit mode - and a commit is sent between Form1 and Form2. Therefore, 2 distinct transaction boundaries can be seen</td>
</tr>
<tr>
<td>H</td>
<td>TP On</td>
<td>TP On</td>
<td>TP On</td>
<td>Both Form1 and Form2 are treated as a single transaction boundary - only after exiting Form2 is the transaction processed</td>
</tr>
</tbody>
</table>

Because transaction processing often covers more than a single form (a single form, after all, is extremely simple to configure) - a system is in place with JD Edwards to allow developers to change the boundary to suit their purposes. The above table depicts 2 forms - and how the form interconnect (or business function interconnect) would operate.
Chapter 4 - Forms and functions supporting TP

Transaction processing is available to the following form types (note - one may have to update me if I missed any under B733.x):

- Fix/Inspect
- Header/Detail
- Headerless Detail

The following events also can be assigned Transaction Processing functions (again - any updates welcome!)

- OK Button Clicked
- OK Post Button Clicked
- Add Record to DB Before
- Add Record to DB after
- Update Record to DB before
- Update Record to DB after
- Add Grid Record to DB Before
- Add Grid Record to DB after
- All Grid Records added to DB
- Update Grid Record to DB before
- Update Grid Record to DB after
- All Grid Records updated to DB
- Delete grid record from DB before
- Delete grid record from DB after
- All grid records deleted from DB
### Appendix 1: Transaction Processing and Record Locking completed at B73.3.1 level

<table>
<thead>
<tr>
<th>App/BSFN</th>
<th>Ver.</th>
<th>Description of Completed Work</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/L Post (B0000122)</td>
<td>B73.1</td>
<td>F0902 records are individually locked and released while adjusting balances.</td>
<td>Future - Need to consider commit/rollback of F0911 updating.</td>
</tr>
<tr>
<td>Regular Next Numbers (X0010)</td>
<td>B73.2</td>
<td>F0002 and F00021 records are locked and released with each update.</td>
<td></td>
</tr>
<tr>
<td>Unique ID Next Numbers (X00022)</td>
<td>B73.2</td>
<td>F00022 records are locked and released with each update.</td>
<td></td>
</tr>
<tr>
<td>Journal Entries (P0911)</td>
<td>B73.2</td>
<td>Commit/rollback of F0911 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Voucher Entry (P0411)</td>
<td>B73.2</td>
<td>Commit/rollback of both F0411 and F0911 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Inventory Transactions: Inventory Adjustments (P4114), Inventory Issues (P4112), Inventory Transfers (P4113), Item Reclassifications (P4116)</td>
<td>B73.2</td>
<td>Used normal T/P for primary parts of the transactions - F4111, F0911, etc. Used record locking with immediate commit per line item on updates to the F41021 table using business function Update F41021 Location (XF41021).</td>
<td>Future - Figure out some way through application design, to rollback updates to the F41021 table when the primary transaction fails.</td>
</tr>
<tr>
<td>Sales Order Entry (P4210)</td>
<td>B73.2</td>
<td>Used record locking with immediate commit per line item on updates to the F41021 table for inventory commitments. Roll back inventory commitments if user presses Cancel, not if technical error occurs.</td>
<td>Future - T/P for primary transaction. Complete rollback for all types of failures.</td>
</tr>
<tr>
<td>Multi Company Vouchers (P041016)</td>
<td>B73.3</td>
<td>Commit/rollback of both F0411 and F0911 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Multi Voucher (P041017)</td>
<td>B73.3</td>
<td>Commit/rollback of both F0411 and F0911 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Store &amp; Forward Voucher (P0411Z1)</td>
<td>B73.3</td>
<td>Commit/rollback of both F0411Z1 and F0911Z1 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Speed Voucher Entry (P0411SV)</td>
<td>B73.3</td>
<td>Commit/rollback of both F0411 and F0911 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Manual Payments (P0413M)</td>
<td>B73.3</td>
<td>Commit/rollback of both F0413 and F0414 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Payroll Transactions - Update Tax History (B0700045) Update Payment History (B0700047) Update Time Entry History (B070005) Update DBA History (B0700055) Update Employee Master (B0700056)</td>
<td>B73.3</td>
<td>Transaction processing is in each of the business functions called by the Payroll Final Update program.</td>
<td>Each function updates a different file so they are set around the file updates. One employee at a time is run through the Final Update UBE so at any one time, only a few records should be locked at a time.</td>
</tr>
<tr>
<td>Landed Cost - P43291</td>
<td>B73.3.1</td>
<td>Commit/rollback of F43121, F0911, and F4111 records for each application of landed costs.</td>
<td>If landed costs are applied at receipt, the landed costs are also part of the TP boundary for PO Receipts.</td>
</tr>
<tr>
<td>PO Receipts (P4312)</td>
<td>B73.3.1</td>
<td>Commit/rollback of F43121, F0911, and F4111 records for each receipt.</td>
<td></td>
</tr>
<tr>
<td>Voucher Match (P4314)</td>
<td>B73.3.1</td>
<td>Commit/rollback of F43121, F0411, and F0911 records on a per document basis.</td>
<td></td>
</tr>
<tr>
<td>Batch Journal Entries (R09110Z)</td>
<td>B73.3.1</td>
<td>Commit/rollback of F0911 and F0911Z1 records on a per document basis. The rollback of the F0911Z1 is that the update to the processed will not occur if the F0911 is not committed.</td>
<td></td>
</tr>
<tr>
<td>Store &amp; Forward Journal Entries (R09110Z)</td>
<td>B73.3.1</td>
<td>Commit/rollback of F0911 and F0911Z1 records on a per document basis. The rollback of the F0911Z1 is that the update to the processed field will not occur if the F0911 is not committed.</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 2: General Troubleshooting FAQ on TP

Question 1 - I am getting the error: “JDB990900 - Failure to communicate with OneWorld Transaction Management Server”?

Answer: The client failed to establish communication with the TMS. Check the following:

- Check to see if RequestedService=TS in the JDE.INI on the client machine. If RequestedService=NONE, the user should not get this error.
- Check the Server setting in the JDE.INI on the client machine. The machine specified here has to be a valid machine. If the Requested Service on the client is TS, then the Available Service on the Server must be TS. Contact the person in charge of the machine that is indicated in Server to see if the server is running.
  *In most cases, the OneWorld Servers are down when this message is received.
- Check the ports in [JDENET] section. This has to be a port on which the TMS is running. To check the port on which the TMS is running, contact the person in charge of the machine that is indicated in Server.
- Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and escalate to JDE OneWorld Technical Customer Support if necessary.

Question 2 - What does the message "Waiting for Transaction Server" indicate?

Answer: This message means that client has sent a request to the server, it has waited 10 seconds and has not yet received a response. The client will then wait again for the ServerTimeout period specified in the JDE.INI. This message is for informational purpose only. It does not necessarily mean that there is an error.

Question 3 - I am getting the error "JDB9909109 - Failed to get available transaction services from the transaction manager"?

Answer: The client has failed to get the transaction services that the TMS is providing. Check the following:

- This error is usually a result of a communication problem between the client and the server. If on the client, check the logs for JDENET errors. If on the server, check the jdedebug.log for JDENET errors.
- On the client this message is received after the "Waiting for Transaction Server" message. Basically, the client has timed out while waiting for the server to respond. This could be the result of a slow network. To verify this, go to the JDE.INI and increase the ServerTimeout period for the client.
- Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and escalate to JDE OneWorld Technical Customer Support if necessary.
Question 4 - What does the message "JDB9909110 - The OneWorld Transaction Manager is not offering the requested transaction service(s)" indicate?

Answer: This message means that the RequestedServices in the JDE.INI are not offered by the TMS. This means that the client's RequestedServices does not correspond to the server's AvailableServices. Check the following:

- Check that the RequestedServices value is typed correctly. It should be NONE, if no services are requested.
- If you wish to request a service, contact with the person responsible for the machine hosting the server to check what the AvailableServices for the TMS are.

Question 5 - What do the following messages (generated by TimeStamp process) mean?

- "<Table Name> table has been changed by another user. Please re-select your data."
- "<Table Name> is the name of the table that you are trying to update"
- "JDB9909124 - The record that you are trying to update has changed after you fetched it. Please re-inquire."

Answer: This means that between the time that you fetched the data from the table and the time that you are trying to update the data in the table, another user updated that data. This essentially means that you are attempting to update based on outdated data. You should re-inquire on your data before completing the update.

Question 6 - What does the message "JDB9909139 - OneWorld TMS Error (Table has no index)" indicate?

Answer: This means that the table that you are trying to update, has no specified index. This error is sent to the client from the server.

Contact the person responsible for the machine hosting the TMS to check whether the Global Table Specs on that server are correct. This person should also check whether the table exists in the specs at all. All the tables should be in the specs and should have at least a primary index defined for them. Only OneWorld specs need to be looked at. The index that is mentioned in this error is not the database index.

Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and contact JDE OneWorld Technical Support if necessary.

Question 7 - What does the error "JDB9909149 - OneWorld TMS Internal Error (Failure to read the table specs on the server)" indicate?

Answer: This means that TMS has failed to read the table specs for the specified table on the server.

Contact the person responsible for the machine hosting the TMS to check whether the Global Table Specs on that server are correct. This person should also check whether the table exists in the specs at all. It may be necessary to do a Server Package Install of the table to get the table specs to the server. You can also easily delete the 2 glbltbl files on the server and then try the request again to see whether they are the problem.

Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and contact JDE OW Technical Support if necessary.
Question 8 - What does the error "JDB<XXXXXXX> - OneWorld TMS Internal Error" indicate?

Answer: This means that a TMS internal error has occurred. However this error is also received if the server has gone down after the client has connected. <XXXXXXX> is the specific error code.

Contact the person responsible for the machine hosting the TMS to verify whether the server is up.

If <XXXXXXX> is any one of the following:

9909140
9909141
9909154

this means that the TMS has failed to allocate some memory on the server.

Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and contact JDE OneWorld Technical Support if necessary.

Question 9 - What does the error "JDB9909160 - OneWorld TMS Internal Error (General Error)" indicate?

Answer: This means that a TMS internal error has occurred. However this error is also received if the server has gone down after the client has connected.

Contact the person responsible for the machine hosting the TMS to verify whether the server is up.

Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and contact JDE OneWorld Technical Support if necessary.

Question 10 - What does the error "JDB9909162 - Failed to find primary index to use for timestamp registry" indicate?

Answer: This means that the table you are trying to use, has no specified index. This error is generated on the client.

Check whether the Global Table Specs on your client machine are correct. Try deleting the glbltbl files and let them recreate when you sign onto OneWorld. You should also check whether the table exists in the specs at all. All the tables should be in the specs and should have at least a primary index defined for them. Only OneWorld specs need to be looked at. The index that is mentioned in this error is not the database index.

Gather the client and server (where TMS runs) jde.ini files, service files, and all logs, and contact JDE OneWorld Technical Support if necessary.

Question 11 - What does the error "JDB9909163 - Number of index fields exceed the maximum supported by timestamp registry" indicate?

Answer: The maximum number of fields in an index is 25. If you have more than 25 fields for a primary unique key, then the table needs to be examined and redesigned.
Contact

For additional information contact:

**Jon Steel**
Xe Upgrade Specialist
Chief Technology Officer
erpSOURCING LLC
1781 Bayside Boulevard
Jacksonville
Florida 32259

Work phone : (603) 487 5477
Cell phone : (904) 382 5701
Fax Phone : (904) 287 0720
Email (main) : cto@spla.sh
ICQ Number : 111747507