



IBM Cloud

Understanding Transaction Manager in WebSphere® Application Server

Kumaran Nathan (kumaran@us.ibm.com)

Krishna Jaladhi (krishnaj@us.ibm.com)

WebSphere Application Server Level 2 Support

May 11, 2016



Support Technical Exchange

Agenda

- Transaction service overview
- Transaction logs
 - ▶ Network Storage(NFS)/Database
 - ▶ Tranlog/Partner logs
- Transaction Timeout
- Transaction Manager high availability
- Common Issues and Troubleshooting
 - ▶ Transaction Recovery
 - ▶ Transaction Timeout
- Debug using Trace and Request trace analyzer (TRA)

Transaction Overview



Introduction to Transactions

- A transaction groups pieces of work that should be atomic (all or nothing). If any piece of the transactional work fails, the entire transaction will roll back.
- The application server uses a transaction manager that takes responsibility of managing transactions across multiple Resource Managers (RM).
- RM can support two types of transactions:
Resource managers include relational databases (JDBC™ data source), messaging systems (JMS™), Enterprise Information Systems (EIS)
- **Local Transaction**
 - ▶ A transaction that is managed internal to a resource manager
- **Global Transaction**
 - ▶ Multiple resources are involved, requiring an external transaction manager WebSphere Application Server

Global/2Phase/XA Transaction

A global transaction is also referred to as a distributed transaction, an XA transaction, a 2PC (2-phase commit) transaction, or just a transaction. This type of transaction is global, or distributed, in the sense that multiple resource managers may be accessed within the transaction and the transaction manager has the ability to coordinate among all the resource managers to ensure the atomicity of update

■ Phase 1—Prepare

- ▶ In this phase, each of the programs (resource managers) involved in the transaction sends a message to the transaction manager that it is ready to and capable of performing its part of the operation. When the Transaction Manager receives confirmation from each of the programs involved, it proceeds to phase 2.

■ Phase 2—Commit or Rollback

- ▶ In this phase, the Transaction manager instructs each of the programs to commit or rollback all of the changes that were requested as part of the transaction.



Transaction logs



Transaction logs

The WebSphere Application Server transaction service writes information to a transaction log for every global transaction that involves two or more resources, or that is distributed across multiple servers. The transaction service maintains transaction logs to ensure the integrity of transactions

- The **tranlog** subdirectory contains all of the files that hold record details of transactions that are managed by WebSphere Application Server, in particular, the current transition state
- The **partnerlog** subdirectory contains files that hold information on resources that are involved in a transaction. The partnerlog subdirectory is important in a recovery scenario to allow WebSphere Application Server to re-create a resource for recovery after the server is recycled
- **Caution** : Never delete these subdirectories in a production environment. If you delete the log files, the processes might not progress or might not complete an outstanding transaction. You might encounter critical data integrity issue, database corruption, pending transaction might never get completed, and so on.

Transaction Logs – Cont...

- Transaction logs can be stored in operating system files or in relational database(V8.5.5.7 or later release)
- High Available (HA) transaction support required the use of a shared file system to host the transaction logs, such as an NFSv4-mounted network attached storage (NAS) or a storage area network (SAN)

Storing transaction logs in a relational database for high availability

https://www.ibm.com/support/knowledgecenter/SSAW57_8.0.0/com.ibm.webSphere.nd.doc/info/ae/ae/tjta_store_logs_in_rdb.html

Transaction logs Cont...

- **NFSv4** releases locks held on behalf of a host in case that host fails. Peer recovery can occur automatically without restarting the failed hardware. Therefore, this version of NFS is **better suited for use with automated peer recovery**.
- NFSv3 holds file locks on behalf of a failed host until that host can restart. In this context, the host is the physical machine running the application server that requested the lock and it is the restart of the host, not the application server, that eventually triggers the locks to release.
- Its not recommended to store transaction file in a shared file system. i.e., Windows® mapped drive

Transaction Timeouts



Transaction Timeout

[Application servers](#) > [WSTEMember1](#) > [Transaction service](#)

Use this page to specify settings for the transaction service. The transaction service is a server runtime component that can coordinate updates to multiple resource managers to ensure atomic updates of data. Transactions are started and ended by applications or the container in which the applications are deployed.

Runtime
Configuration

General Properties

Transaction log directory

* Total transaction lifetime timeout
 seconds **←** The number of seconds to allow for a transaction that is started on this server, before the transaction service initiates timeout completion

* Async response timeout
 seconds

* Client inactivity timeout
 seconds

* Maximum transaction timeout
 seconds **←** The number of seconds a transaction that is propagated into this application server can remain inactive before it is ended by the transaction service
This value must be equal to, or greater than, the total transaction lifetime timeout

Heuristic retry limit
 retries

Heuristic retry wait
 seconds

Enable logging for heuristic reporting

Heuristic completion direction
 ▼

Accept heuristic hazard

Enable file locking **←** Must be unchecked for NFS3

Enable transaction coordination authorization

Additional Properties

- [Custom properties](#)



QueryTimeouts

■ **webSphereDefaultQueryTimeout**

- ▶ Available from WebSphere Application server 7.0.0.21.
- ▶ Establishes a default querytimeout, which is the number of seconds (0 mean infinite, or no timeout) that an SQL statement may run before timing out. This default value is overridden during a JTA™ Transaction if custom property **syncQueryTimeoutWithTransactionTimeout** is enabled.
- ▶ Application can override the query timeout for a statement at any time by invoking the **java.sql.Statement.setQueryTimeout**.

■ **syncQueryTimeoutWithTransactionTimeout**

- ▶ Uses the time remaining (if any) in a JTA transaction as the default querytimeout for SQL statements.
- ▶ The default query timeout value is computed immediately before executing each statement.

SyncQueryTimeoutWithTransactionTimeout - Cont..

- Below is a code snippet of an application demonstrating a scenario using `webSphereDefaultQueryTimeout = 20` and `syncQueryTimeoutWithTransactionTimeout = true`

```
statement = connection.createStatement(); // query timeout of 20 seconds is used
statement.executeUpdate(sqlcommand1); // query timeout of 20 seconds is used
statement.executeUpdate(sqlcommand2);
transaction.setTransactionTimeout(30);
transaction.begin();
try{
// query timeout of 30 seconds is used
statement.executeUpdate(sqlcommand3);
// assume the above operation took 5 seconds, remaining time = 30 - 5 seconds
// query timeout of 25 seconds is used
statement.executeUpdate(sqlcommand4);
}
finally
{
transaction.commit();
}
// query timeout of 20 seconds is used
statement.executeUpdate(sqlcommand6);
```

Transaction Manager high availability



Transaction Manager high availability

- The high availability of the transaction service enables any server in a cluster to recover the transactional work for any other server in the same cluster. This facility forms part of the overall WebSphere Application Server high availability (HA) strategy.
- As a vital part of providing recovery for transactions, the transaction service logs information about **active transactional work in the *transaction recovery log***. The transaction recovery log stores the information in a persistent form, which means that any transactional work in progress at the time of a server failure can be resolved when the server is restarted. This activity is known as *transaction recovery processing*. In addition to completing outstanding transactions, this processing also ensures that any locks held in the associated resource managers are released.

Peer Recovery

- The standard recovery process requires server restart(to complete pending transaction). WebSphere Application Server provides a high availability strategy known as *transaction peer recovery* to avoid server restart in the event of server failure.
- Peer recovery is provided within a server cluster. A peer server (another cluster member) can process the recovery logs of a failed server while the peer continues to manage its own transactional workload.
- **Automated peer recovery**
 - ▶ This style is the default for peer recovery initiation. If an application server fails, WebSphere Application Server automatically selects a server to undertake peer recovery processing on its behalf, and passes recovery back to the failed server when it restarts. To use this model, enable transaction log recovery and configure the recovery log location for each cluster member.
- **Manual peer recovery**
 - ▶ You must explicitly configure this style of peer recovery. If an application server fails, you use the administrative console to select a server to perform recovery processing on its behalf.

Configuring transaction peer recovery

Step 1 : Enable transaction log recover

[WebSphere application server clusters](#) > **HAWLMCluster**

Use this page to change the configuration settings for a cluster. If the application servers that is a member of the cluster fails,

Runtime Configuration Local Topology Reports

General Properties

* Cluster name

HAWLMCluster

Bounding node group name

DefaultNodeGroup

Prefer local

Enable failover of transaction log recovery

Step 2: Provide Log directory. Network file system or Relational Database

Application servers

[Application servers](#) > **WSTEMember1** > Transaction service

Use this page to specify settings for the transaction service. The updates to multiple resource managers to ensure atomic updates in a container in which the applications are deployed.

Configuration

General Properties

Transaction log directory

Manual Peer recovery :

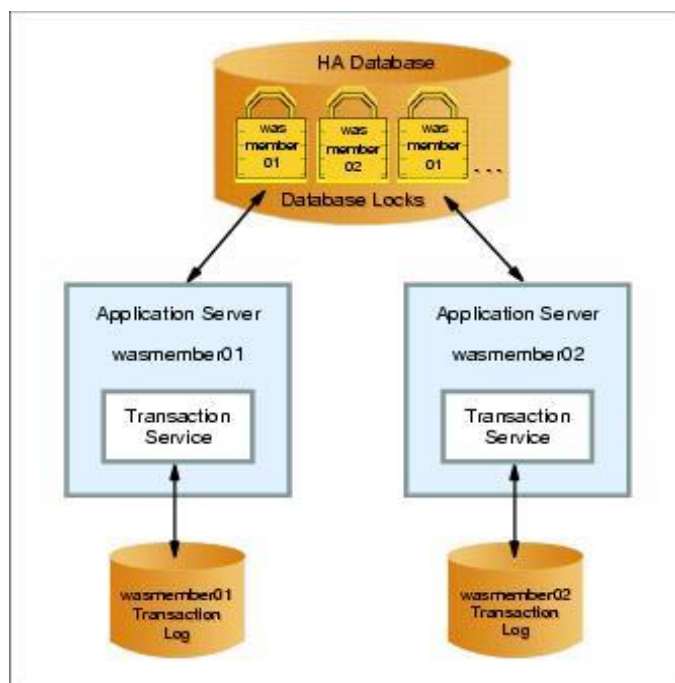
http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=tjta_cfgpeer_man

Automated peer recovery

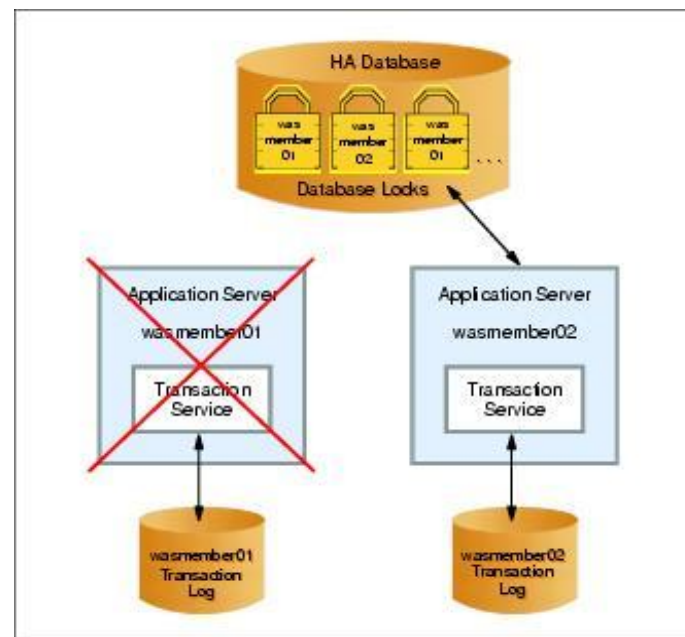
https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/tjta_cfgpeer_auto.html?cp=SSAW57_8.5.5

Transaction Manager high availability – Sample Scenario

Two application servers perform two-phase commit (2PC) transactions, they place table or row locks in the database



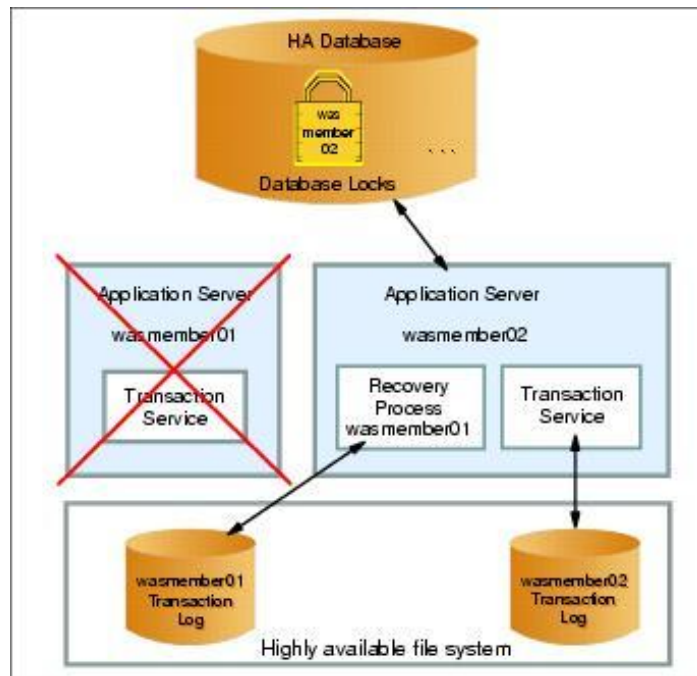
Server Failure : Transaction service on failed server is out of service. The in-flight transactions might leave locks in the database, which blocks the peer server from gaining access to the locked records.



There are only two ways to complete the transactions and release the locks. One is to restart the failed server and the other is to start an application server process on another box that has access to the transaction logs.

Sample Scenario – Cont ..

Using the HAManager support, a highly available file system and a lease-based locking protocol, a **peer recovery process** will be started in a peer member of the cluster. The recovery locks are released and in-flight transactions are committed.



Common Issues and Trouble Shooting



WTRN0062E and WTRN0063E Exceptions

- **WTRN0062E:** An illegal attempt to use multiple resources that have only one-phase capability has occurred within a global transaction.

There can be only a single resource that is limited to one-phase capability can be enlisted in a global transaction.

This error occurs if multiple resource managers that are only one-phase capable are used in a global transaction, or if multiple unshared connections from a single such resource manager are used in a global transaction.

Solution is to modify the application either to use a single resource that is limited to one-phase capability, or to use two-phase capable XAResources

WTRN0062E and J2CA0030E Errors While Trying To Do Last Participant Support Extensions

<http://www.ibm.com/support/docview.wss?uid=swg21244805>

- **WTRN0063E:** An illegal attempt to commit a one phase capable resource with existing two phase capable resources has occurred.

The transaction service has refused an attempt to commit a one-phase capable resource with a transaction that already involves other two-phase capable resources. Either Last Participant Support (LPS) is not available, or if LPS is available, the application does not accept the heuristic risk that this would involve.

Ensure that one- and two-phase capable resources are not involved in the same transaction, or if LPS is available, reconfigure the application to accept the heuristic risk that this would involve.

Transaction Recovery Issue



Transaction Recovery Issue

- tranlogs are used to record details of transactions, specifically the transition of transaction states.
- partnerlogs are used to record information about the resources involved in transactions, allowing the resources to be recreated when recovery takes place during a server restart.
- Messages WTRN0132I to WTRN0151I are associated with XA recovery messages which appear during the server startup depending upon the scenario, for more information about these messages refer knowledge center
http://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.messages.doc/com.ibm.ws.Transaction.resources.TransactionMsgs.html?lang=en

Transaction Recovery Messages

WTRN0028I: Transaction service recovering {0} transactions
WTRN0132I: Transaction recovery for {0} initiated with server uuid {1} and restart epoch {2}
WTRN0133I: Transaction recovery processing for this server is complete
WTRN0134I: Recovering {0} XA resource manager(s) from the transaction partner logs
WTRN0135I: Transaction service recovering no transactions.
WTRN0136I: Processing recovered transaction {0} (tid={1}) with {2}
WTRN0137I: Recovered transaction (tid={0}) committing xid {1} with {2}
WTRN0138I: Recovered transaction (tid={0}) rolling back xid {1} with {2}
WTRN0139I: Recovered transaction (tid={0}) forgetting xid {1} with {2}
WTRN0140I: Recovered transaction (tid={0}) committed xid {1} successfully with {2}
WTRN0141I: Recovered transaction (tid={0}) commit of xid {1} with {2} resulted in {3}
WTRN0142I: Recovered transaction (tid={0}) rolled back xid {1} successfully with {2}
WTRN0143I: Recovered transaction (tid={0}) rollback of xid {1} with {2} resulted in {3}
WTRN0144I: Recovered transaction (tid={0}) forgot xid {1} successfully with {2}
WTRN0145I: Recovered transaction (tid={0}) forget of xid {1} with {2} resulted in {3}
WTRN0146I: Obtained {0} xid(s) from xa recover on {1} of which {2} will be processed by this server
WTRN0147W: Recovered xid {0} from {1} is from a earlier restarted server instance with epoch {2}
WTRN0148I: Recovered xid {0} from {1} - xid has no associated transaction and will be rolled back
WTRN0149I: Recovered xid {0} from {1} - xid has associated transaction (tid={2}) with logged state {3}
WTRN0150I: Response from rolling back recovered xid {0} from {1} - {2}
WTRN0151I: Preparing to call xa recover on {0}

Transaction Recovery Resources

https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/recovering_from_failed_transaction_recovery_websphere_application_server?lang=en

https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/recovering_failed_sibus_transactions_in_websphere_application_server?lang=en

<https://developer.ibm.com/answers/questions/168157/how-to-resolve-transaction-and-partnerlog-recovery.html>

Exception occurs during recovery of Oracle database transactions

<https://www.ibm.com/support/docview.wss?uid=swg21196663>



Resolving WTRN0005W messages

- **WTRN0005W:** The XAResource for a transaction participant could not be recreated and transaction recovery may not be able to complete properly. The resource was **<XXXXXXXXXXXX>**
- **Enable trace on Transaction=all**
- From trace search for **“recovered=false”**
- **WTRN0151I:** Preparing to call xa recover on XAResource:
SIBus:TestBus:XXXhostNode01.server1-TestBus XARecoveryDat < auditXaRecover
Exit XARecoveryDat > getXARminst Entry <null> index=1, **recoveryID=2,**
recovered=false xxxxxxxx>

Resolving WTRN0005W messages - Cont..

- Configure following custom property and enter values discovered from trace (**recoveryID=xx** **recovered=false**)

Name: **REMOVE_PARTNER_LOG_ENTRY**

Value: 2,4,6(comma separated values, Default is null)

APAR info

<http://www.ibm.com/support/docview.wss?uid=swg1PM62977>

Cell=ADMINIB-NDLUO2TNode07 Cell, Profile=AppSrv06

Application servers

Application servers > server1 > Transaction service > Custom properties > REMOVE_PARTNER_LOG_ENTRY

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.

Configuration

General Properties

* Name
REMOVE_PARTNER_LOG_ENTRY

* Value
2

Description

Apply OK Reset Cancel

Configuring Transaction service custom properties refer knowledge center URL

https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/rjta_transerv_custproperties.html

Resolving WTRN0005W messages - Cont..

- Start Application Server in recovery mode.

For example, <WebSphere_home>/<profile_name>/bin>**startServer.bat/sh server1 -recovery**

- After successful completion, start server in normal mode and make sure to delete REMOVE_PARTNER_LOG_ENTRY, save Configuration changes

Note: REMOVE_PARTNER_LOG_ENTRY custom property is effective only when both of the following situations apply.

- The application server has no transactions that currently require recovery. You can establish this by checking the SystemOut.log file.

WTRN0135I: Transaction service recovering no transactions.

WTRN0134I: Recovering 0 XA resource manager(s) from the transaction partner logs

WTRN0133I: Transaction recovery processing for this server is complete

- When you start application server in -recovery mode

Transaction Timeout Issue



Transaction Timeout (WTRN0006W)

- **WTRN0006W** occurs when a application can't complete transactional work within the timeout specified in the transaction service(**Total transaction lifetime timeout**).
- Message **WTRN0006W: Transaction "xxxxxxxx...(XID)" has timed out after xxx seconds.** Is logged into SystemOut.log. Default value is 120secs.
- Message "**WTRN0124I: When the timeout occurred the thread with which the transaction is, or was most recently, associated was Thread[WebContainer : 50,5,main]. The stack trace of this thread when the timeout occurred was:....**" , logged after **WTRN0006W**.

Symptoms that can cause WTRN0006W

- Hung thread messages in SystemOut logs, logged with message [WSVR0605W](#) / [WSVR0606W](#)
- Long running database queries
- Long running garbage collection activity
- Resource contention on application server / back end resources

Debugging WTRN006W from trace

- Common trace string used to gather trace, variations are available depending upon components involved.
*=info:Transaction=all:EJBContainer=all:WAS.j2c=all:RRA=all

- Get **XID** from WTRN006W message

```
[1/7/16 16:38:48:831 EDT] 00000011 TimeoutManage I WTRN006W: Transaction
00000129A979A63200000001000000A502880914D5789071AC9584125FE1BC807189AE3900000129A979A63200000001000000A502880914D5789071A
C9584125FE1BC807189AE3900000001 has timed out after 120 seconds.
```

- Search XID in the trace and find out the thread id on which the transaction has started

```
[1/7/16 16:36:48:818 EDT] 0000006b XidImpl < XidImpl Exit
{XidImpl: formatId(57415344), gtrid_length(36), bqual_length(40),
data(00000129A979A63200000001000000A502880914D5789071AC9584125FE1BC807189AE3900000129A979A63200000001000000A502880914D57890
71AC9584125FE1BC807189AE3900000001)}
```

```
[1/7/16 16:36:48:818 EDT] 0000006b TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 165
```

- Use text editor or any other tools to pull the thread and investigate the activity on the thread and for possible delays.
- Useful trace points

```
[1/7/16 16:36:48:833 EDT] 0000006b RegisteredRes 1 (SPI) RESOURCE registered with Transaction. TX: 165, Resource:
```

```
[7/6/10 16:37:12:776 EDT] 0000006b TransactionCo 1 Suspending TX cntxt:
```

```
com.ibm.ws.Transaction.JTA.TransactionImpl@28d828d8#tid=165
```

```
[7/6/10 16:38:53:689 EDT] 0000006b TransactionCo 1 Resumed TX cntxt:
```

```
com.ibm.ws.Transaction.JTA.TransactionImpl@28d828d8#tid=165
```

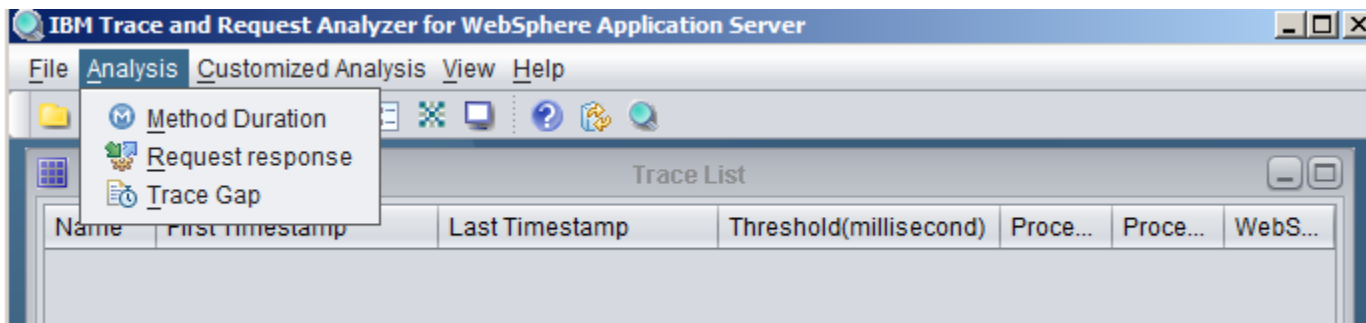
```
[7/6/10 16:38:53:708 EDT] 0000006b TranManagerIm 1 Transaction has timed out
```

```
[7/6/10 16:38:55:762 EDT] 0000006b TranStrategy 1 Rolling back TX cntxt due to bean exception:
```

```
com.ibm.ws.Transaction.JTA.TransactionImpl@28d828d8#tid=165
```


Debug long running transaction using TRA

- Trace and Request trace analyzer(TRA) provides Analysis of Method duration, Request/Response duration and Trace gap
- Provides various options, has built-in analysis pattern and customizable trace pattern for almost all components for customized analysis.
- Provides ability to perform analysis on pulled thread from multiple trace files.
- Example showing customized analysis for Transaction timeouts



Using Trace and Request trace analyzer (TRA) Contd..

The screenshot shows the 'Analysis Pattern' dialog box with the 'Trace Pattern (version 1.1) from repository server' list. The list contains the following entries:

Active	Name	Entry Expression	Exit Expression	Description
<input type="checkbox"/>	SessionProc	SessionCo...	SessionConte... < Ses...	Session processing time Entry to Exit (includes inter processing time and ServletSp pro...
<input type="checkbox"/>	getConnection	Connection...	ConnectionMan < ina...	Specific filter for WAS 51 when tackling Connection Pool exhausted or performance iss...
<input type="checkbox"/>	executeJdbc	> execute	< execute	Statement execution response time to database useful when tackling J2CA0045E issue...
<input type="checkbox"/>	SessionFreq	SessionCo...	SessionConte... creat...	Time between new sessions Useful if investigating session locking. Trace option com.ib...
<input type="checkbox"/>	EJBMethod	EJSContain...	EJSContainer > EJBp...	Time for EJB Method to complete Trace option EJBContainer=all
<input type="checkbox"/>	EJBpost	EJSContain...	EJSContainer < EJB...	Time for EJBpost Method to complete Note Commit occurs during Post invoke. Trace opti...
<input type="checkbox"/>	QueryExecut	WSJdbcCo...	WSJdbcPrepare < ex...	Time it takes to prepare the statment and execute the Query sent to and return from the...
<input type="checkbox"/>	Connectionr	Connection...	ConnectionEve > con...	Time taken to create a connection - useful for Connection Pool exhausted and performa...
<input type="checkbox"/>	executequery	WSJdbcPre...	WSJdbcPrepare < ex...	Trace option RRA=all
<input type="checkbox"/>	executeJdbc	WSJdbcPre...	WSJdbcPrepare < ex...	Trace option RRA=all (adapter)
<input checked="" type="checkbox"/>	EJBpre	EJSContain...	EJSContainer < EJB...	Use for transaction timeout issues WTRN0006W Trace option Transaction=all
<input checked="" type="checkbox"/>	TransactionC	Transaction...	TranManagerSe < co...	Use for transaction timeout issues WTRN0006W Trace option Transaction=all
<input checked="" type="checkbox"/>	TransactionC	Transaction...	TransactionIm < notif...	Use for transaction timeout issues WTRN0006W Trace option Transaction=all
<input type="checkbox"/>	Webcontaine	WebContai...	SRTServletRes < fini...	Webcontainer start and end point

Using Trace and Request trace analyzer (TRA) Contd..

Time(ms)	Trace	Line N...	File
236,470	[7/15/08 8:44:44:143 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurre...	120725	trace_...
	[7/15/08 8:48:40:613 EDT] 0000004f TransactionIm < notifyCompletion Exit	210089	trace_...
231,855	[7/15/08 8:52:29:794 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurre...	288740	trace_...
	[7/15/08 8:56:21:649 EDT] 0000004f TransactionIm < notifyCompletion Exit	66713	trace.I...
229,909	[7/15/08 8:56:21:654 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurre...	66844	trace.I...
	[7/15/08 9:00:11:563 EDT] 0000004f TransactionIm < notifyCompletion Exit	191269	trace.I...
229,166	[7/15/08 8:48:40:620 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurre...	210220	trace_...
	[7/15/08 8:52:29:786 EDT] 0000004f TransactionIm < notifyCompletion Exit	288609	trace_...
7,452	[7/15/08 8:44:36:707 EDT] 00000062 TransactionIm 1 (SPI) Transaction BEGIN occurre...	114221	trace_...
	[7/15/08 8:44:44:159 EDT] 00000062 TransactionIm < notifyCompletion Exit	121278	trace_...
868	[7/15/08 8:47:04:663 EDT] 0000003d TransactionIm 1 (SPI) Transaction BEGIN occurre...	168415	trace_...
	[7/15/08 8:47:05:531 EDT] 0000003d TransactionIm < notifyCompletion Exit	169574	trace_...


```

[7/15/08 8:44:44:142 EDT] 0000004f TxPrimaryKey < <init> Exit
    com.ibm.ws.Transaction.JTA.TxPrimaryKey@73;11B26C056EE:1:73
[7/15/08 8:44:44:142 EDT] 00000062 EJSContainer > EJBpreInvoke(6:remove) Entry
[7/15/08 8:44:44:142 EDT] 0000004f FailureScopeC > registerTransaction Entry
    com.ibm.ws.Transaction.JTA.FailureScopeController@34803480
    com.ibm.ws.Transaction.JTA.TransactionImpl@535c535c#tid=115
    false
[7/15/08 8:44:44:142 EDT] 00000062 TransactionCo > preInvoke Entry
[7/15/08 8:44:44:142 EDT] 0000004f FailureScopeC < registerTransaction Exit
[7/15/08 8:44:44:143 EDT] 00000062 TransactionCo > completeTxTimeout Entry
[7/15/08 8:44:44:143 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 115
[7/15/08 8:44:44:143 EDT] 00000062 TranManagerIm < completeTxTimeout Exit
  
```

Using Trace and Request trace analyzer (TRA) Contd..

Time(ms)	Trace	Line N...	File
236,470	[7/15/08 8:44:44:143 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 115	120725	trace_...
	[7/15/08 8:48:40:613 EDT] 0000004f TransactionIm < notifyCompletion Exit	210089	trace_...
231,855	[7/15/08 8:52:29:794 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 141	288740	trace_...
	[7/15/08 8:56:21:649 EDT] 0000004f TransactionIm < notifyCompletion Exit	66713	trace.l...
229,909	[7/15/08 8:56:21:654 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 152	66844	trace.l...
	[7/15/08 9:00:11:563 EDT] 0000004f TransactionIm < notifyCompletion Exit	191269	trace.l...
229,166	[7/15/08 8:48:40:620 EDT] 0000004f TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 128	210220	trace_...
	[7/15/08 8:52:29:786 EDT] 0000004f TransactionIm < notifyCompletion Exit	288609	trace_...
7,452	[7/15/08 8:44:36:707 EDT] 00000062 TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 1...	114221	trace_...
	[7/15/08 8:44:44:159 EDT] 00000062 TransactionIm < notifyCompletion Exit	121278	trace_...
868	[7/15/08 8:47:04:663 EDT] 0000003d TransactionIm 1 (SPI) Transaction BEGIN occurred for TX: 1...	168415	trace_...
	[7/15/08 8:47:05:531 EDT] 0000003d TransactionIm < notifyCompletion Exit	169574	trace_...


```

[7/15/08 8:48:40:613 EDT] 0000004f LocalTIDTable > removeLocalTID Entry
    115
[7/15/08 8:48:40:613 EDT] 0000004f LocalTIDTable < removeLocalTID Exit
[7/15/08 8:48:40:613 EDT] 0000004f FailureScopeC > deregisterTransaction Entry
    com.ibm.ws.Transaction.JTA.FailureScopeController@34803480
    com.ibm.ws.Transaction.JTA.TransactionImpl@535c535c#tid=115
    false
[7/15/08 8:48:40:613 EDT] 0000004f FailureScopeC < deregisterTransaction Exit
[7/15/08 8:48:40:613 EDT] 0000004f TransactionSt 3 reset
[7/15/08 8:48:40:613 EDT] 0000004f TransactionIm < forgetTransaction Exit
[7/15/08 8:48:40:613 EDT] 0000004f TransactionIm < notifyCompletion Exit
[7/15/08 8:48:40:613 EDT] 0000004f TransactionIm < rollback (SPI) Exit
[7/15/08 8:48:40:613 EDT] 0000004f TranManagerIm < rollback (SPI) Exit
[7/15/08 8:48:40:613 EDT] 0000004f TransactionCo < postInvoke Exit
[7/15/08 8:48:40:613 EDT] 0000004f ContainerTx 3 releaseResources : State = Rolledback
[7/15/08 8:48:40:613 EDT] 0000004f EJSContainer < EJBpostInvoke(0:onMessage)**** throws
    com.ibm.ejs.container.UnknownLocalException: Exit
[7/15/08 8:48:40:613 EDT] 0000004f MessageEndpoi < MessageEndpoint.afterDelivery Exit
  
```

Resources for learning and debugging

- **Learning more about Java Transaction Service (JTS)**
<http://www.ibm.com/support/docview.wss?uid=swg21247193>
- **Troubleshooting: Java Transaction Service (JTS) problems**
<http://www.ibm.com/support/docview.wss?uid=swg21247192>
- **The Support Authority: Interpreting a WebSphere Application Server trace file**
http://www.ibm.com/developerworks/websphere/techjournal/0704_supauth/0704_supauth.html
- **IBM Trace and Request Analyzer for WebSphere Application Server**
https://www.ibm.com/support/knowledgecenter/S5LLVC_5.0.0/com.ibm.esupport.tool.tra.doc/helpdoc/readme.html

Summary

- Transaction in WebSphere Application Server
- Transaction Timeouts
- Transaction recovery process
- Highly available Transaction Manager
- Most common issues

Connect with us!

1. Get notified on upcoming webcasts

Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

Questions and Answers



Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:
<http://www.ibm.com/software/support/einfo.html>