



IBM Software Group

# Transactions in WebSphere® Application Server: Overview and Problem Determination

David Tiler

WebSphere Application Server L2 Support Analyst



WebSphere® Support Technical Exchange



# Agenda

- Overview of Transactions and concepts
- Transaction Support in WebSphere Application Server
- Common Problems and Troubleshooting Steps

# Part One

## Overview of Transactions and concepts



# 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.
- Resource Manager Local Transaction
  - ▶ Transaction is contained within one resource manager
  - ▶ Resource managers include relational databases (JDBC™ data source), messaging systems (JMS), Enterprise Information Systems (EIS)
- Global Transaction
  - ▶ Multiple resources are involved, requiring an external transaction manager → WebSphere Application Server

# WebSphere Application Server Transaction Service

- Transaction Service provides:
  - ▶ A callable interface to manage transactional work
  - ▶ Coordination of global transactions
  - ▶ Automatic propagation of transaction context with remote requests to other servers
  - ▶ Coordination of transaction completion with other servers
  - ▶ Recovery after server failure

## JTA and JTS

- JTS – Java™ Transaction Service
  - ▶ Java implementation of CORBA Object Transaction Service
  - ▶ Responsible for the coordination and propagation of transactions between J2EE™ servers
- JTA – Java Transaction API
  - ▶ J2EE interface to the JTS
  - ▶ Responsible for transaction boundary demarcation, enlistment of resources through XA interface
- WebSphere Application Server implements JTA and JTS

# UserTransaction Interface

- JTA comprises three interfaces
  - ▶ TransactionManager and Transaction interfaces are used internally by the application server itself
  - ▶ UserTransaction is used by application code to manage transactions
- Application does a JNDI™ lookup of UserTransaction interface: “java:comp/UserTransaction”
- Methods available to manage transactions:
  - ▶ begin(); commit(); rollback(); getStatus(); setRollbackOnly(); setTransactionTimeout(int seconds);
  - ▶ Javadoc™ available at:  
<http://java.sun.com/j2ee/1.4/docs/api/javax/transaction/UserTransaction.html>

## XA and Two-Phase Commit

- Global transactions involve multiple resources
  - ▶ Examples: two databases, database and messaging system, etc.
  - ▶ Resource manager must support two-phase commit through XAResource interface
  - ▶ Data sources and connection factories can be enabled for XA in the Administrative Console
- Cannot enlist one-phase capable resource in transaction with other resources (without Last Participant Support)
- Result -> WTRN0062E: An illegal attempt to use multiple resources that have only one-phase capability has occurred within a global transaction.



## Part Two

# Transaction Support in WebSphere Application Server



# Bean and Container Managed Transactions

- Bean Managed Transactions
  - ▶ Can only be used with session beans
  - ▶ Application code manages transactions using UserTransaction API
- Container Managed Transactions
  - ▶ Can be used with session beans and must be used with entity beans
  - ▶ Transactions managed by EJB™ Container
  - ▶ Configured by specifying transaction attribute in EJB deployment descriptor (ejb-jar.xml)
  - ▶ Transparent to the application
  - ▶ Usually easier to implement and better performance

# Container Managed Transactions

- Transaction attribute is set per bean or per bean method
- Set using tooling (WSAD, RAD, AST)
- Six different transaction attributes define how the EJB container manages the transaction when EJB method is invoked
  - ▶ Descriptions available at:  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tejb\\_addcontainertran.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tejb_addcontainertran.html)
- UserTransaction cannot be used in CMT

## Transactions in Servlets

- Servlets can start a global transaction with the UserTransaction interface
- Transaction context can be propagated from a servlet
- Transaction must be completed before exiting the servlet
- Without a global transaction, a Local Transaction Containment (LTC) is started

# Local Transaction Containment

- LTC is started in any servlet or EJB without a global transaction
- No programming interface
- Managed by Web and EJB Containers
- Involves only a single resource manager

## LTCs and Connection Pooling

- Shareable connections are used by default
- Shareable connections do not return to the free pool when closed by the application in an LTC
- Connections are automatically released at end of LTC
- If LTC is long-running, the maximum connection pool size can be reached and `ConnectionWaitTimeoutExceptions` may occur
- Solution is to use `UserTransaction` or unshareable connections
- developerWorks® article:  
[http://www.ibm.com/developerworks/websphere/library/techarticles/0506\\_johnsen/0506\\_johnsen.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0506_johnsen/0506_johnsen.html)

## Programming Model Extensions

- In V6.0 and V6.1 all PME's are included in the Base product
- Previously available in V5.0 Enterprise Edition and WBI Server Foundation V5.1
- PME's related to Transactions:
  - ▶ Last Participant Support
  - ▶ Extended JTA Support
  - ▶ ActivitySessions

## Last Participant Support

- Enables a single one-phase (non-XA) capable resource to be enlisted with one or more two-phase (XA) capable resources in a global transaction
- Can be enabled per application through tooling (WSAD, RAD, AST)
- LPS coordinates the commit or rollback of the two-phase and one-phase resources



## Extended JTA Support

- Provides additional APIs beyond the UserTransaction interface
- Application does a JNDI lookup of ExtendedJTATransaction: "java:comp/websphere/ExtendedJTATransaction"
- Access global and local transaction IDs associated with the thread:  
getLocalID(); getGlobalID();
- Register a transaction synchronization callback to be notified of transaction completion:  
registerSynchronizationCallback(SynchronizationCallback);  
unregisterSynchronizationCallback(SynchronizationCallback);
- Can be used within all J2EE application modules

# ActivitySessions

- An alternative unit-of-work scope
- Used to coordinate multiple local transactions whose outcomes are independent
- Can be used in place of or in addition to global transactions
- Can encapsulate global transactions, but global transaction cannot encapsulate ActivitySession
- Used as a “session” context without coordinating two-phase commit among resources
- Available to all J2EE modules
- Work is either checkpointed or reset (to last point of consistency)
- EJBs can use container-managed or bean-managed ActivitySessions
- For bean-managed ActivitySessions, application does a JNDI lookup of “java:comp/websphere/UserActivitySession”

## Web Services Atomic Transaction Support

- Based on WS-Atomic Transaction specification
- Configured in deployment descriptors, not programmatically
- Must be JSR-109 client
- If web service is requested within a global transaction, then transaction is propagated
- Intended for transactions of short duration
- Cannot be used with firewall
- Not recommended for use across enterprises

# Transaction timeout configuration

- Timeouts for transactions are configurable in the Administrative Console under Transaction Service
- Total transaction lifetime timeout (default 120 seconds)
  - ▶ Maximum lifetime of a transaction started on the server
  - ▶ Overridden programmatically or in deployment descriptor
- Client inactivity timeout (default 60 seconds)
  - ▶ Maximum time between transactional requests from remote client (when transaction is imported from a client)
- Maximum transaction timeout
  - ▶ Maximum lifetime of transactions that are imported from remote clients
- Transaction rolls back when timeout is reached
- Rollback will occur when control returns to the container

# Tracing

- Almost any Transaction problem will need a Transaction trace:
  - ▶ Transaction=all (V6.1 and V6.0)
  - ▶ Transaction=all=enabled (V5.1)
  - ▶ com.ibm.ejs.jts.\*=all=enabled (V5.0)
- Additional traces:
  - ▶ WAS.j2c almost always needed
  - ▶ RRA if database is involved
  - ▶ Messaging and JMSApi if messaging is involved
  - ▶ EJBContainer if EJBs are involved
- Ensure that enough historical trace data is collected
  - ▶ Default transaction timeout is 120 seconds
  - ▶ Set Maximum File Size to 50 MB and Maximum Number of Historical Files to 5 to ensure that entire transaction life cycle is traced

## Part Three

# Common Problems and Troubleshooting Steps



## Common Problem: Transaction Rollback

- TransactionRolledbackException or CSITransactionRolledbackException is seen in logs
- This simply means that a transaction has rolled back
- Common causes
  - ▶ An exception occurred – could be an application exception, exception from resource manager (SQLException) or exception from WebSphere Application Server
  - ▶ Transaction timed out

## Common Problem: Transaction Rollback

- Analyze the Transaction trace to determine the cause
- Find the TransactionRolledbackException in the trace and note the Thread ID
- Follow the Thread ID up (review previous trace entries on the Thread ID)
- Look for exceptions on the thread
- Alternate approach: look for beginning of the transaction:  
[10/24/07 14:57:52:659 EST] 0000015d  
TransactionIm 1 (SPI) Transaction BEGIN  
occurred for TX: 116708  
and follow the thread down until you find the cause
- Also look for transaction timeouts (WTRN0006W)



## Common Problem: Transaction Timeout

- **WTRN0006W** messages are seen in the logs:  
[10/24/07 14:59:52:662 EST] 0000000f  
TimeoutManage I WTRN0006W: Transaction  
0000011056531F8D0000000050001C7E40B304AAB611AB  
4FC574CE136F63A9E07BE78A01B0000011056531F8D00  
0000050001C7E40B304AAB611AB4FC574CE136F63A9E0  
7BE78A01B00000001 has timed out after 120  
seconds.
- This means that a transaction did not commit or rollback within the “Total transaction lifetime timeout” (120 seconds by default)

# Common Problem: Transaction Timeout

- Common causes
  - ▶ Slow performance
  - ▶ Thread is stuck waiting for something or hung
  - ▶ Simply a lot of work happening; needs more than the “Total transaction lifetime timeout” to complete

## Common Problem: Transaction Timeout

- Analyze the Transaction trace to determine the cause
- It is necessary to follow multiple threads because a separate thread is used to timeout the transaction
- Find the Transaction ID for the transaction that timed out and locate the thread that the Transaction ID was started on, then follow that thread to find the cause of the problem
- Detailed steps to analyze the trace to resolve a transaction timeout can be found in Trace Example #2 in this document:  
[http://www.ibm.com/developerworks/websphere/techjournal/0704\\_supauth/0704\\_supauth.html](http://www.ibm.com/developerworks/websphere/techjournal/0704_supauth/0704_supauth.html)

# Common Problem: Transaction Recovery

- Background information
  - ▶ Data is written to transaction logs any time a transaction is started  
*install\_root/profiles/app\_server\_name/tranlog*
  - ▶ During normal shutdown, WebSphere Application Server resolves all in-flight (incomplete) transactions
  - ▶ We cannot resolve in-flight transactions when an application server is stopped ungracefully (due to a crash, a “kill -9”, power outage, etc.)
  - ▶ When an application server starts, the transaction logs are checked for unresolved transactions
  - ▶ If found, transaction recovery is attempted in order to commit or rollback the transactions

## Common Problem: Transaction Recovery

- Transaction recovery often completes with no problems
- Transaction recovery errors could occur when an application server is started
- You'll know the problem is related to transaction recovery if you see `com.ibm.ws.Transaction.JTA.RecoveryManager` in the stack trace
- Common causes of problems
  - ▶ Backend resource not available
  - ▶ XA errors (such as `XAER_RMERR`) coming from JDBC driver or JMS code (not a WebSphere Application Server problem)
  - ▶ Possible defect in WebSphere Application Server transaction recovery

# Common Problem: Transaction Recovery

- Solution
  - ▶ If the transactions are being recovered on a test system:
    - Enable Transaction tracing In server.xml
    - Restart application server to produce a trace of the problem and send the trace to support
    - Delete all files under *install\_root/profiles/app\_server\_name/tranlog*
    - Restart the application server
    - With no data in the transaction logs, there will be no transactions to recover when the application server restarts
  - ▶ If the transactions are being recovered on a production system:
    - Enable Transaction tracing in server.xml
    - Restart application server to produce a trace of the problem and send the trace to support
    - Route traffic to other servers until the problem is resolved to ensure that no transaction data is lost

# Summary

- Overview of Transactions and concepts
- Transaction Support in WebSphere Application Server
- Common Problems and Troubleshooting Steps

# Additional WebSphere Product Resources

- 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/>
- Learn about other upcoming webcasts, conferences and events:  
[http://www.ibm.com/software/websphere/events\\_1.html](http://www.ibm.com/software/websphere/events_1.html)
- Join the Global WebSphere User Group Community: <http://www.websphere.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a Flash replay with step-by-step instructions for using the Electronic Service Request (ESR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My support emails:  
<http://www.ibm.com/software/support/einfo.html>



# Questions and Answers

