Managing Transactions

At the end of this module you will be able to:

- ✓ Configure transactions using console
- ✓ Monitor transactions using console

Road Map

1. **Configuring and Monitoring Transactions**
   - Configuring Transactions
   - Monitoring Transactions
   - The Transaction Log

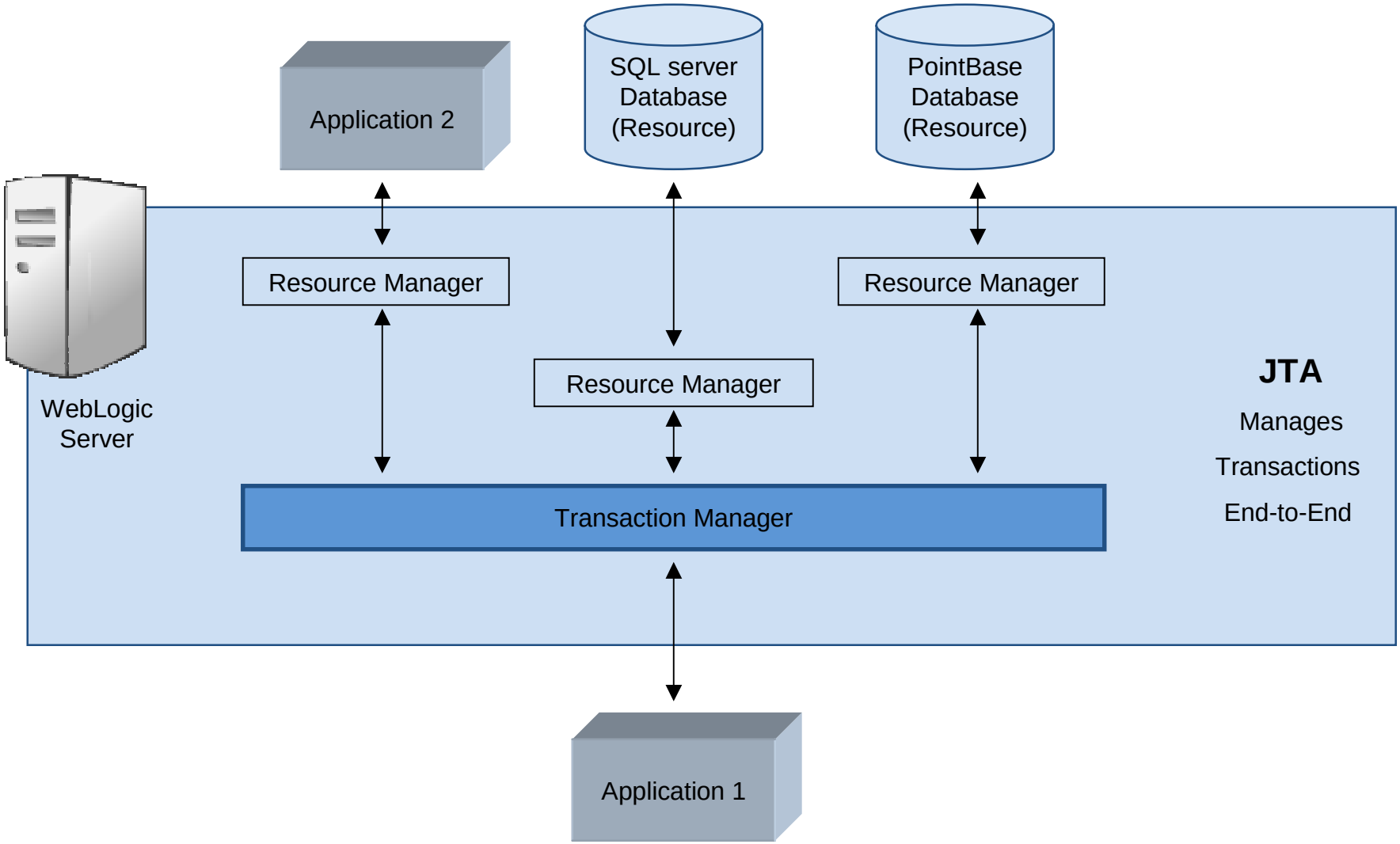
What Is a Transaction?

- A *transaction* is a mechanism to handle groups of operations as though they were one.
- Either all operations in a transaction occur or none at all.
- Operations involved in a transaction might rely on multiple servers and databases.

ACID Properties of a Transaction

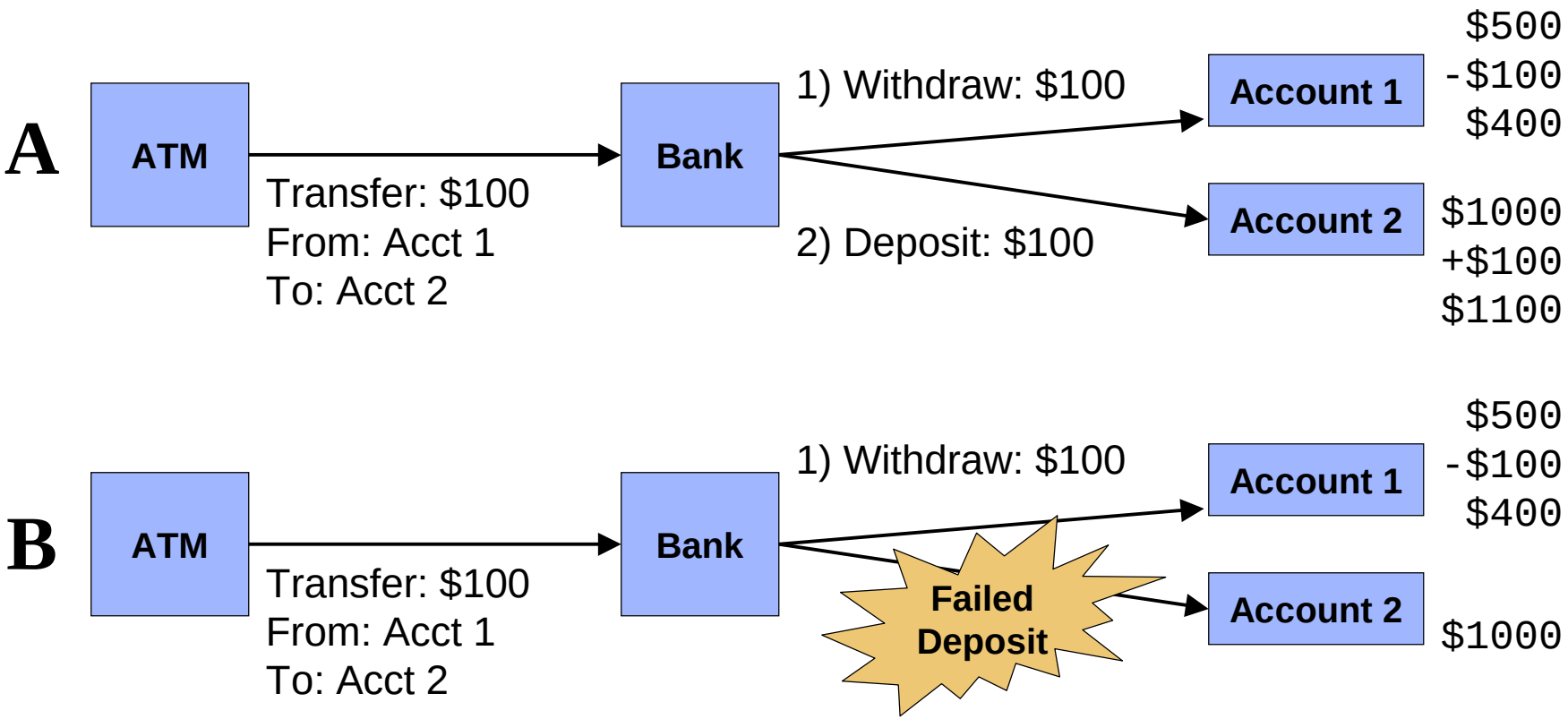
- A *transaction* is formally defined by the set of properties known by the acronym ACID.
- *Atomicity*: a transaction is done or undone completely. In the event of a failure, all operations and procedures are undone, and all data rolls back to its previous state.
- *Consistency*: a transaction transforms a system from one consistent state to another consistent state.
- *Isolation*: each transaction occurs independently of other transactions occurring at the same time.
- *Durability*: completed transactions remain permanent, even during system failure.

Transaction Management



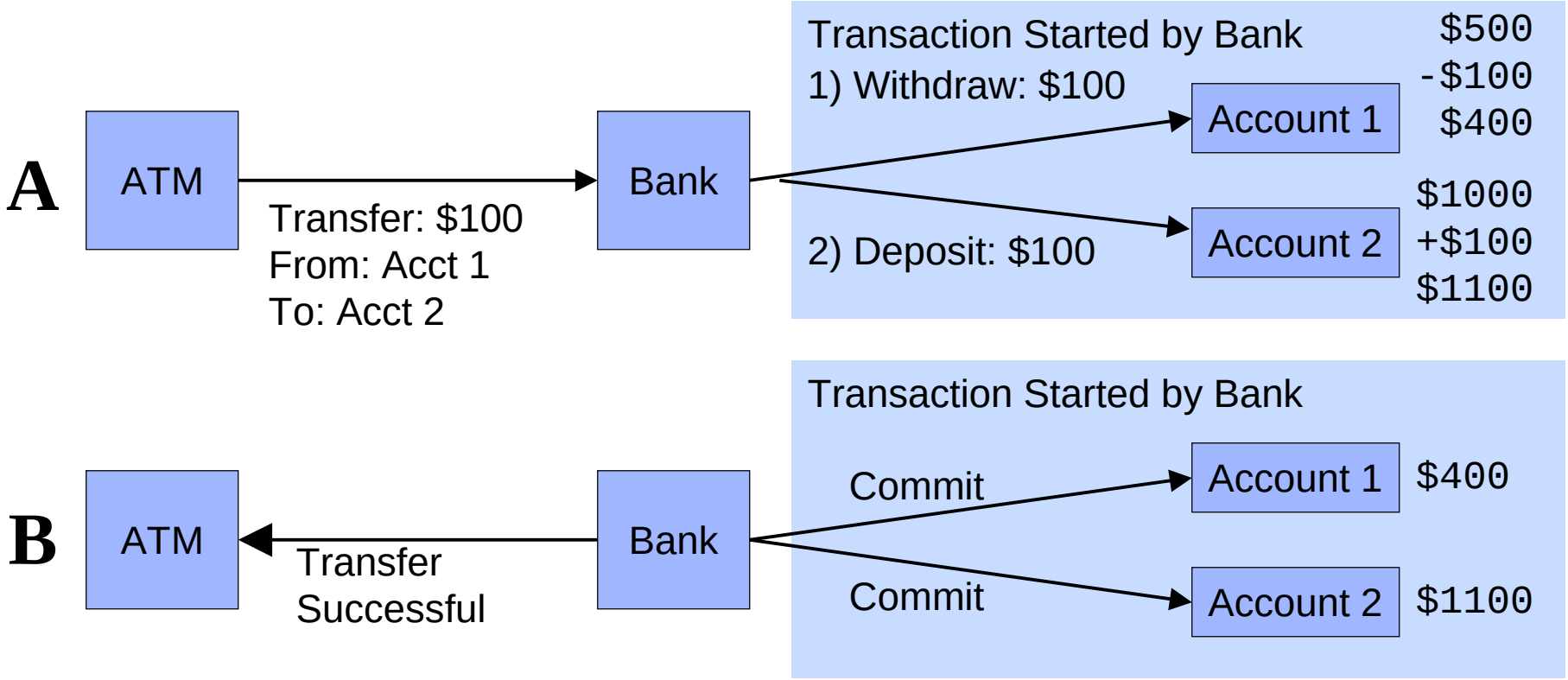
Transferring Without Transactions

- Successful transfer (A)
- Unsuccessful transfer (accounts are left in an inconsistent state) (B)



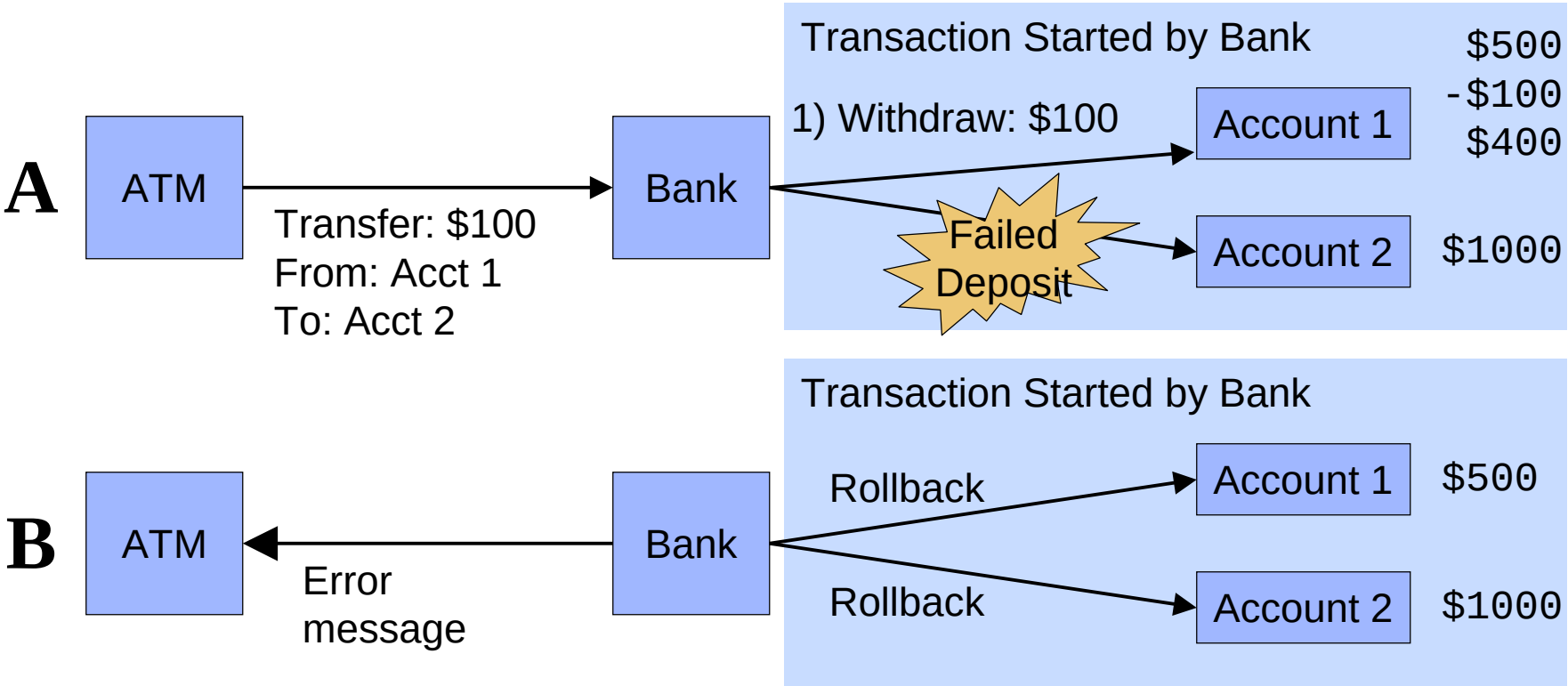
Successful Transfer with Transactions

- Changes within a transaction are buffered. (A)
- If a transfer is successful, changes are *committed* (made permanent). (B)



Unsuccessful Transfer with Transactions

- Changes within a transaction are buffered. (A)
- If a problem occurs, the transaction is *rolled back* to the previous consistent state. (B)



Types of Transactions

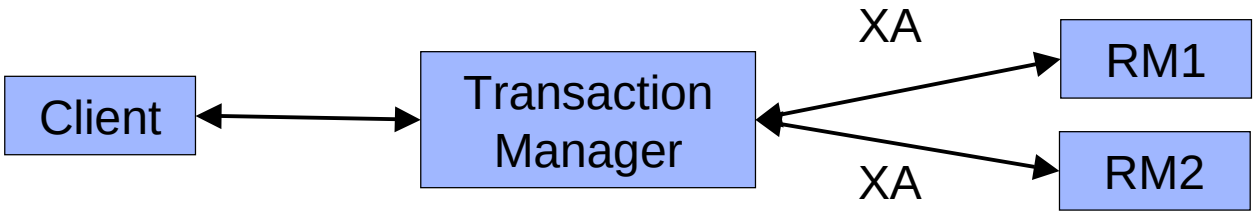
- A *local transaction* deals with a single resource manager. They use the non-Extended Architecture (non-XA) interface between WebLogic Server and resource managers.
- A *distributed transaction* coordinates or *spans* multiple resource managers.
- Global transactions can deal with multiple resource managers. They use the Extended Architecture (XA) interface between WebLogic Server and resource managers.
  - You need to create non-XA or XA resources for local transactions. However, for global transactions, you only need to create XA resources.

The Two-Phase Commit Protocol

- The *Two-Phase Commit (2PC)* protocol uses two steps to commit changes within a distributed transaction.
  - Phase 1 asks RMs to *prepare* to make the changes
  - Phase 2 asks RMs to commit and make the changes permanent, or to roll back the entire transaction
- A *global transaction ID* (`XID`) is used to track all changes associated with a distributed transaction.

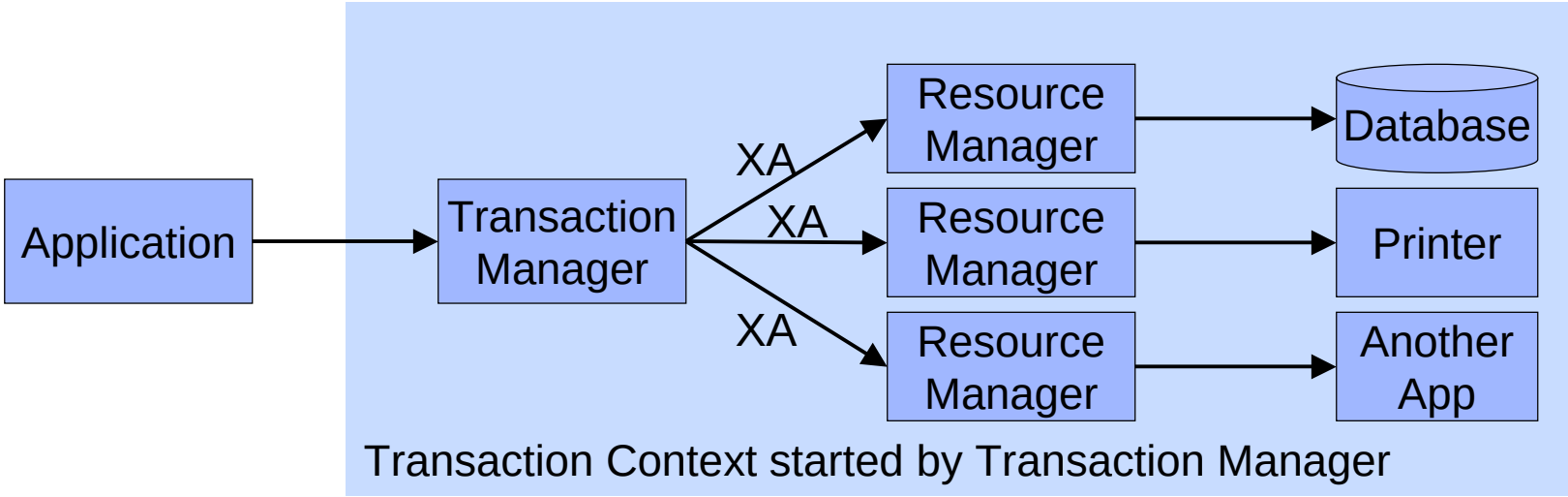
Extended Architecture Protocol

- The *Extended Architecture (XA)* protocol:
  - Is the interface used between WLS and RMs
  - Implements the 2PC protocol
  - Allows programs to control RMs that are involved in distributed transactions

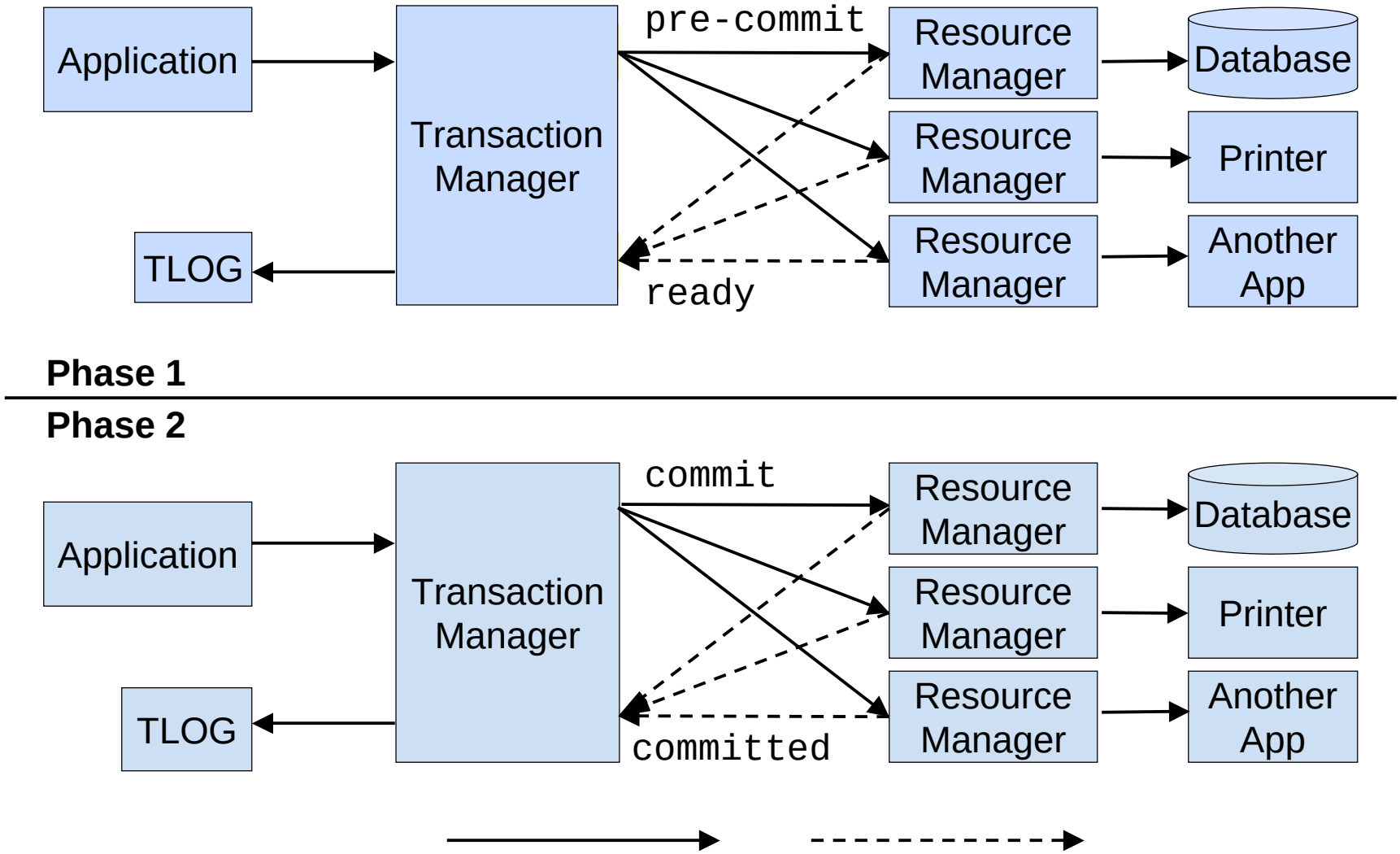


Transaction and Resource Managers

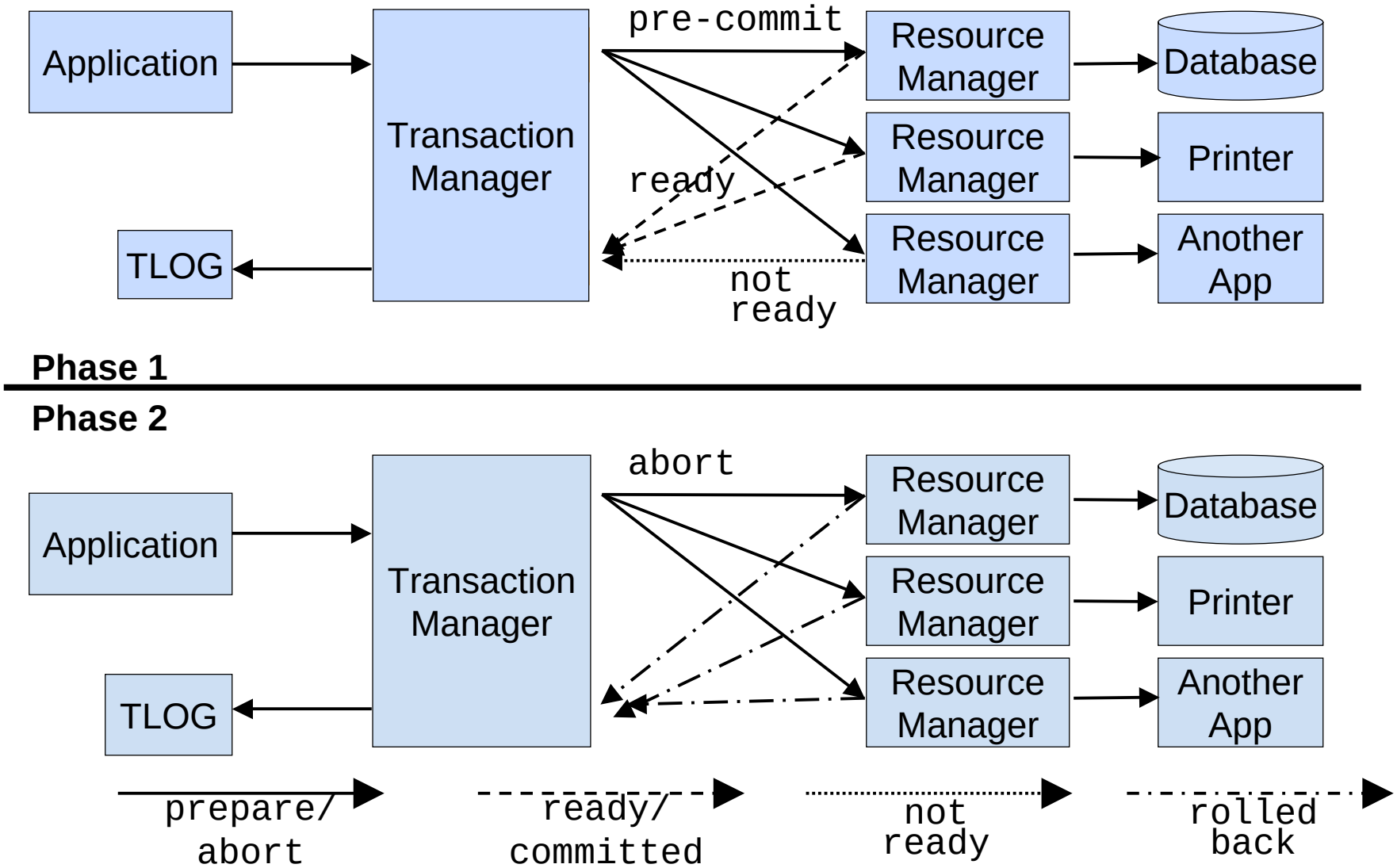
- A *transaction manager* coordinates multiple resource managers.
- The 2PC protocol is used to coordinate the transaction.
- The XA protocol implements 2PC.



Successful Two-Phase Commit



Unsuccessful Two-Phase Commit



Java Transaction API (JTA)…

- WLS uses JTA to implement and manage transactions.
- WLS JTA provides the following support:
  - It creates unique transaction identifier (`XID`)
  - It supports an optional transaction name
  - It tracks objects involved in transactions
  - It notifies databases of transactions
  - It orchestrates 2PC using XA
  - It executes rollbacks
  - It executes automatic recovery procedures when failure
  - It manages time-outs

Configuring Transactions

View changes and restarts

Pending changes exist. They must be activated to take effect.

Activate Changes

Undo All Changes

Domain Structure

- wl_server
  - Environment
    - Servers
    - Clusters
    - Virtual Hosts
    - Migratable Targets
    - Machines
    - Work Managers
    - Startup & Shutdown Classes
  - Deployments
  - Services
    - Messaging
    - JDBC
    - Persistent Stores
    - Path Services
    - Foreign JNDI Providers
    - Work Contexts
    - XML Registries
    - XML Entity Caches
    - jCOM
    - Mail Sessions
    - File T3
    - JTA
  - Security Realm
  - Interoperability
  - Diagnostics

How do I...

- Configure domain JTA options
- Configure JTA

System Status

Health of Running Servers

Settings for wl_server

Configuration | Monitoring | Control | Security | WebService Security | Notes

General | JTA | EJBs | Web Applications | SNMP | Logging | Log Filters

Save

Use this page to define the Java Transaction API (JTA) configuration of this WebLogic Server domain.

Timeout Seconds:	500	The transaction timeout seconds for active transactions, before the prepared state. More Info...
Abandon Timeout Seconds:	86400	The transaction abandon timeout seconds for transactions in the second phase of the two-phase commit (prepared and later). More Info...
Before Completion Iteration Limit:	10	The maximum number of cycles that the transaction manager will perform the beforeCompletion synchronization callback for this WebLogic Server domain. More Info...
Max Transactions:	10000	The maximum number of simultaneous in-progress transactions allowed on a server in this WebLogic Server domain. More Info...
Max Unique Name Statistics:	1000	The maximum number of unique transaction names for which statistics will be maintained. More Info...
Checkpoint Interval Seconds:	300	The interval at which the transaction manager creates a new transaction log file and checks all old transaction log files to see if they are ready to be deleted. More Info...
☑ **Forget Heuristics**		Specifies whether the transaction manager will automatically perform an XAResource forget operation for heuristic transaction completions. More Info...
Unregister Resource Grace Period:	30	The grace period (number of seconds) that the transaction manager waits for transactions involving the resource to complete before unregistering a resource. The grace period can help minimize the risk of abandoned transactions because of an unregistered resource, such as a JDBC data source module packaged with an application. More Info...

1

2

Configuring the Transaction Log

- Each server has a transaction log which stores information about committed transactions coordinated by the server that may not have been completed.
  - WebLogic Server uses the transaction log when recovering from system crashes or network failures.
- You cannot directly view the transaction log as the records are in a binary format
  - Stored in the default persistent store for the server
- T-log files must be migrated if migrating to new machine.

JTA Configuration Options…

Field	Description
Timeout Seconds	Specifies the time in which a transaction will timeout, if uncommitted.
Abandon Timeout Seconds	Specifies the maximum time that a transaction manager will persist in attempting to complete a transaction during the second phase of the transaction.
Before Completion Iteration Limit	Specifies the maximum number of cycles that the transaction manager will perform the beforeCompletion() synchronization callback for this WebLogic Server domain.
Max Transactions	Specifies the maximum number of simultaneous in-progress transactions allowed on a server in the domain.
Max Unique Name Statistics	Specifies the maximum number of unique transaction names for which statistics will be maintained.

…JTA Configuration Options

Field	Description
Checkpoint Interval Seconds	Specifies the interval at which the transaction manager creates a new transaction log file and checks all old transaction log files to see if they are ready to be deleted.
Forget Heuristics	Specifies whether the transaction manager will automatically perform an XAResource forget() operation for heuristic transaction completions.
Unregister Resource Grace Period	Specifies the grace period, in seconds, that the transaction manager waits for transactions involving the resource to complete before unregistering a resource.
Security Interoperability Mode	Specifies the security mode to use for XA calls in inter-domain transactions.

Creating XA Resources

Create a New JDBC Data Source

Back | Next | Finish | Cancel

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

What would you like to name your new JDBC data source?

Name: dwJDBCDataSource

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name: dwJDBCDataSource

What database type would you like to select?

Database Type: PointBase

What database driver would you like to use to create database connections?

Database Driver: *PointBase's Driver (Type 4 XA) Versions:4.X,5.X

Back | Next | Finish | Cancel

Creating Non-XA Resources…

Create a New JDBC Data Source

Back | Next | Finish | Cancel

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

What would you like to name your new JDBC data source?

Name: dwJDBCDataSource

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name: dwJDBCDataSource

What database type would you like to select?

Database Type: PointBase

What database driver would you like to use to create database connections?

Database Driver: *PointBase's Driver (Type 4) Versions:4.X,5.X

Back | Next | Finish | Cancel

…Creating Non-XA Resources

Transaction Options

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

☑ **Supports Global Transactions**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

○ **Logging Last Resource**

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.

○ **Emulate Two-Phase Commit**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.

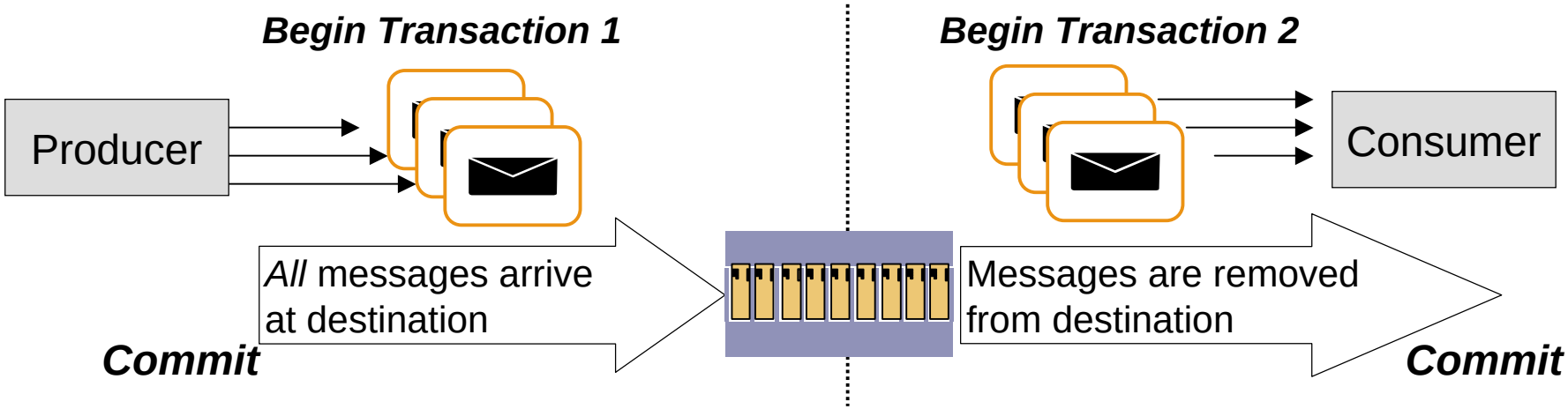
◉ **One-Phase Commit**

Logging Last Resource

- You can configure a JDBC data source to enable the Logging Last Resource (LLR) transaction optimization, which:
  - Enables one non-XA resource to participate in a global transaction
  - Has improved performance and the same ACID guarantee as XA
- The LLR optimization improves performance by:
  - Removing the need for an XA JDBC driver to connect to the database. XA JDBC drivers are typically inefficient compared to non-XA JDBC drivers.
  - Reducing the number of processing steps to complete the transaction, which also reduces network traffic and I/O.
  - Removing the need for XA processing at the database level (if the database is the one non-XA resource).

Transacted Messaging

- A JMS client can use JTA to participate in a distributed transaction.
- Alternatively, a JMS client can demarcate transactions local to the JMS Session, through a transacted session.
- Participation in a transaction is optional.



Inter-Domain Transactions

- WebLogic Server supports global transactions across domains on different versions of WebLogic Server.
- To enable WebLogic Server domains for inter-domain transactions:
  - Enable trust between the different domains
  - Ensure that each contributing XA resource has a unique name
  - Ensure that only one of the participating resources in the distributed transaction can emulate the two-phase commit or XA protocol

Enable Trust Among Different Domains

Settings for dizzyworld

Configuration | Monitoring | Control | Security | Web Service Security | Notes

General | Filter | Unlock User | Embedded LDAP | Roles | Policies

Save

This page allows you to define the general security settings for this WebLogic Server domain. Use this page to change the default security realm for the WebLogic domain.

Default Realm: myrealm — Select the security realm that should be used as the default (active) realm for this WebLogic Server domain. More Info...

☐ **Anonymous Admin Lookup Enabled** — Specifies whether anonymous, read-only access to WebLogic Server MBeans should be allowed from the MBeanHome API. More Info...

Advanced

Security Interoperability Mode: default — Specifies the security mode to use for XA calls in cross-domain transactions. Only applies to transactions in which some participating resources are running on older versions of WebLogic Server. More Info...

☐ **Enable Generated Credential** — Specifies whether a credential (usually a password) should be generated randomly for this WebLogic Server domain. This credential is used to enable a trust relationship between two domains. If you want to establish trust between two domains, you must ensure that they have the same credential by unchecking Enable Generated Credential and specifying the same value as the credential for both domains. More Info...

Credential: ********** — The credential for this WebLogic Server domain. If Enable Generated Credential is unchecked because you want to establish trust between two domains, specify a credential here and in the other domain. More Info...

Confirm Credential: ********** — Re-enter the credential. More Info...

Monitoring Transactions

Settings for JTARuntime

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security | Notes

General | Health | Channels | Performance | Threads | Timers | Workload | Security | Default Store | JMS | JTA

Summary | Transactions By Name | XA Resources | Non-XA Resources | Transactions | Recovery Services

This page shows the summary of all transaction information for all resource types on the server.

Transactions Total Count:	56	The total number of transactions processed. This total includes all committed, rolled back, and heuristic transaction completions
Transactions Committed Total Count:	56	The total number of transactions committed since the server was started. More Info...
Transactions Rolled Back for Timeout Total Count:	0	The number of transactions that were rolled back due to a timeout expiration. More Info...
Transactions Rolled Back for Resource Errors Total Count:	0	The number of transactions that were rolled back due to a resource error. More Info...
Transactions Rolled Back for Application Errors Total Count:	0	The number of transactions that were rolled back due to an application error. More Info...
Transactions Rolled Back for System Errors Total Count:	0	The number of transactions that were rolled back due to an internal system error. More Info...
Heuristic Completions Total Count:	0	The number of transactions that completed with a heuristic status since the server was started. More Info...
Abandoned Transactions Total Count:	0	The total number of transactions that were abandoned since the server was started. More Info...
Active Transactions Total Count:	0	The number of active transactions on the server. More Info...
Average Commit Time:	0	The average amount of time (in milliseconds) that transactions coordinated by this server have taken to commit. More Info...

Monitoring Transactions by Resource

- For a particular resource, the console provides monitoring of transactional outcomes:
  - Number of transactions attempted
  - Number of commits/rollbacks
  - Number of heuristic outcomes

Monitoring Transactions

Settings for mainserver

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security | Notes

General | Health | Channels | Performance | Threads | Timers | Workload | Security | Default Store | JMS | **JTA**

Summary | Transactions By Name | **XA Resources** | Non-XA Resources | Transactions | Recovery Services

Use this page to monitor transactions coordinated by this server.

Customize this table

XA Resources

Showing 1 - 1 of 1 Previous | Next

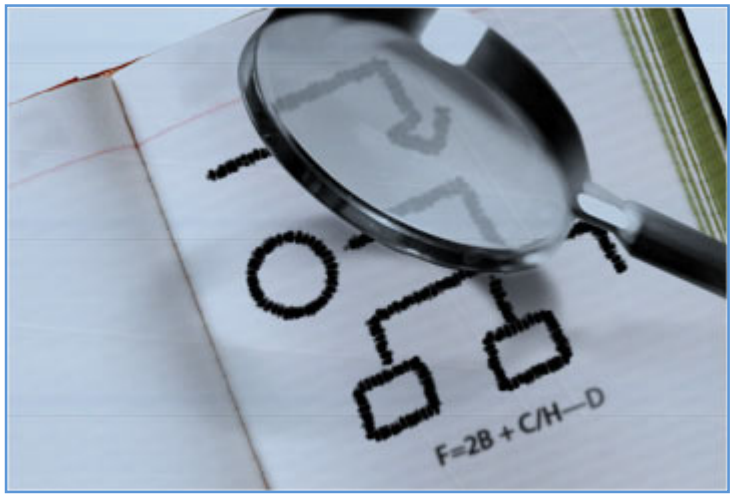
Name	Transactions	Commits	Rollbacks	Timeout Rollbacks	Resource Rollbacks	Application Rollbacks	Transaction Abandoned Total Count
MyJDBC Data Source	0	0	0	0	0	0	0

Showing 1 - 1 of 1 Previous | Next

Section Review

In this section we discussed:

- ✓ Configuring transactions
- ✓ Monitoring transactions
- ✓ Transaction logging



Exercise

Data Sources and Transactions

- In these labs you will work with monitoring Data Sources.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Module Review

In this module we discussed:

- ✓ Configure transactions using console
- ✓ Monitoring transactions in WebLogic Server