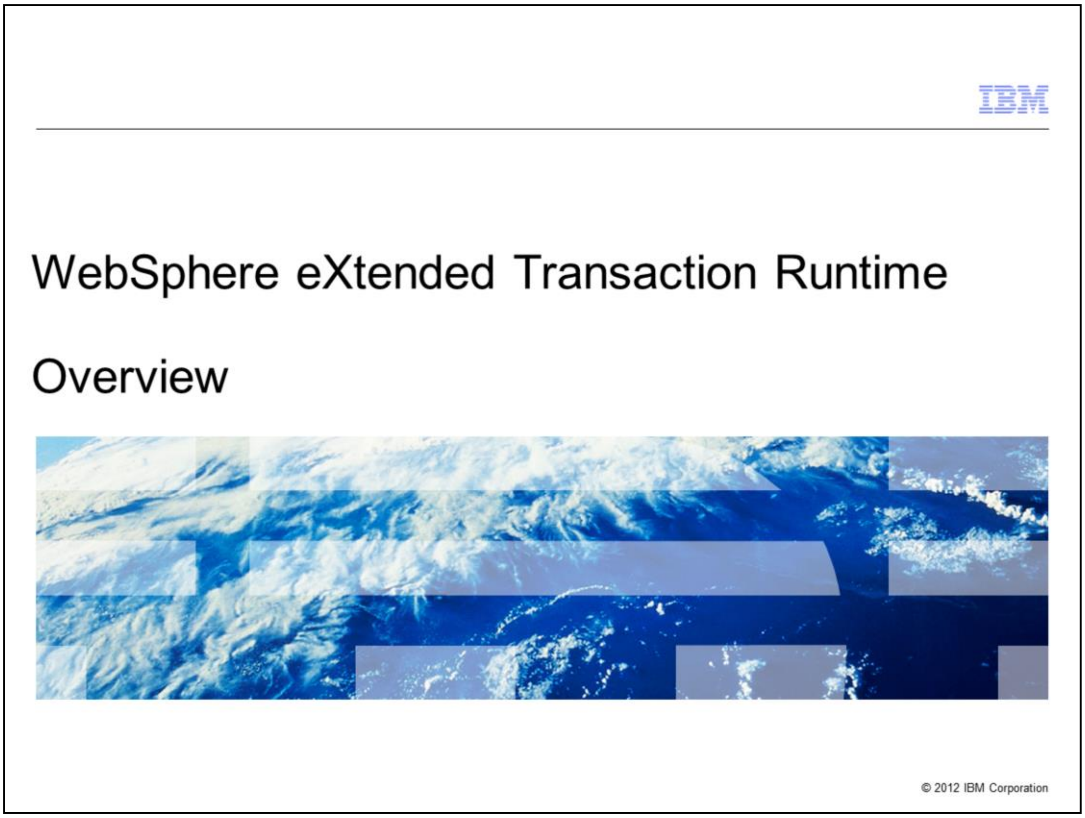
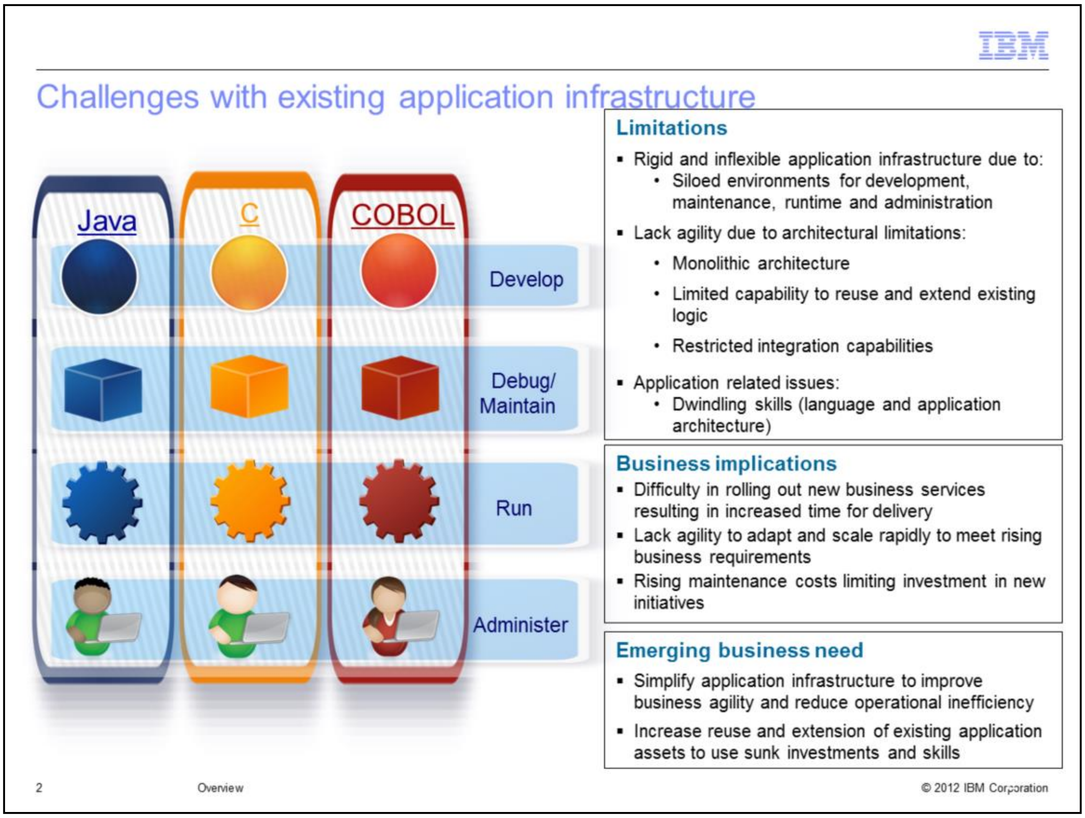


# WebSphere eXtended Transaction Runtime

# Overview

This presentation provides an overview of WebSphere® eXtended Transaction Runtime - or WXTR - and talks about the new features introduced in V2.1



## Challenges with existing application infrastructure

**Limitations**

- Rigid and inflexible application infrastructure due to:
  - Siloed environments for development, maintenance, runtime and administration
- Lack agility due to architectural limitations:
  - Monolithic architecture
  - Limited capability to reuse and extend existing logic
  - Restricted integration capabilities
- Application related issues:
  - Dwindling skills (language and application architecture)

**Business implications**

- Difficulty in rolling out new business services resulting in increased time for delivery
- Lack agility to adapt and scale rapidly to meet rising business requirements
- Rising maintenance costs limiting investment in new initiatives

**Emerging business need**

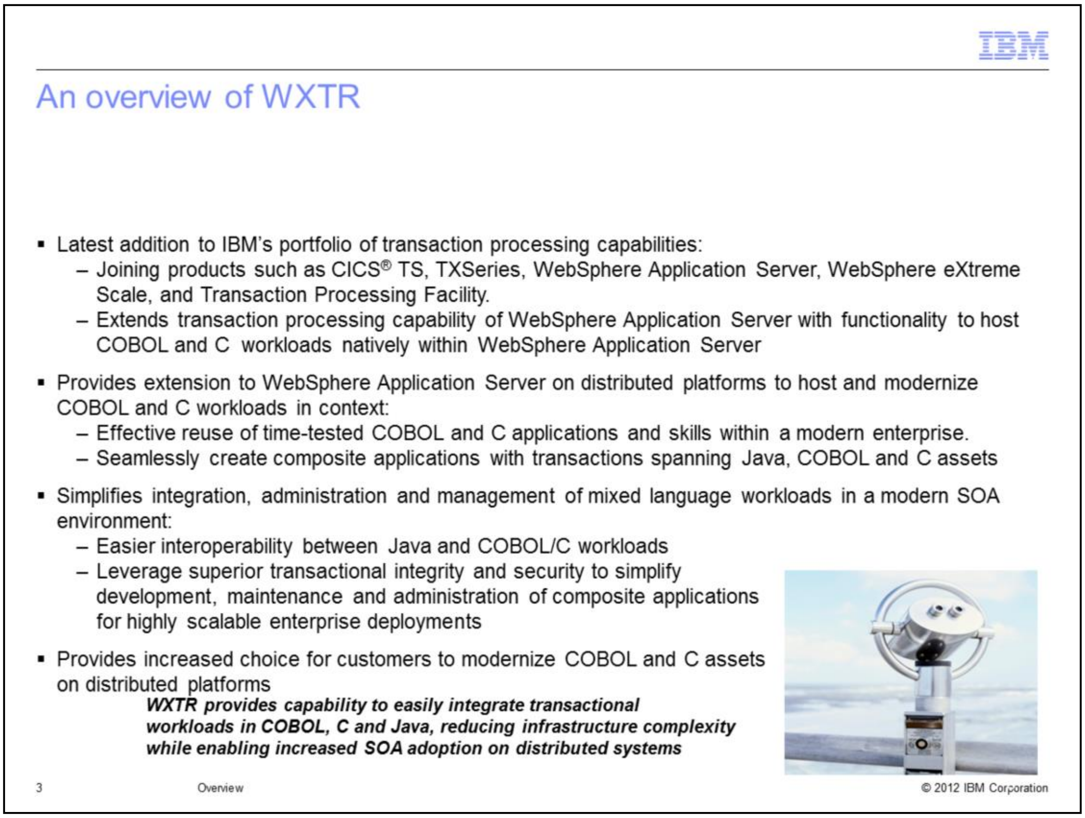
- Simplify application infrastructure to improve business agility and reduce operational inefficiency
- Increase reuse and extension of existing application assets to use sunk investments and skills

2 Overview © 2012 IBM Corporation

Aging applications have a lot of limitations, from architecture to performance to usability. If the application architecture is monolithic, reusing existing code and extending the logic is extremely difficult. What was once conceived as self-sufficient code has limited integration capabilities. The infrastructure has different components developed and maintained in a silo. And it is hard to find the skills required to maintain these applications.

Consider the business implications of these problems: The cost of maintaining these applications keeps increasing. Because of the rigid nature of these applications, the agility to respond to ever changing business needs is lacking. And it is difficult to extend your business to other channels.

The key requirement is to reduce cost by simplifying application infrastructure and improve business agility.



## An overview of WXTR

- Latest addition to IBM's portfolio of transaction processing capabilities:
  - Joining products such as CICS® TS, TXSeries, WebSphere Application Server, WebSphere eXtreme Scale, and Transaction Processing Facility.
  - Extends transaction processing capability of WebSphere Application Server with functionality to host COBOL and C workloads natively within WebSphere Application Server
- Provides extension to WebSphere Application Server on distributed platforms to host and modernize COBOL and C workloads in context:
  - Effective reuse of time-tested COBOL and C applications and skills within a modern enterprise.
  - Seamlessly create composite applications with transactions spanning Java, COBOL and C assets
- Simplifies integration, administration and management of mixed language workloads in a modern SOA environment:
  - Easier interoperability between Java and COBOL/C workloads
  - Leverage superior transactional integrity and security to simplify development, maintenance and administration of composite applications for highly scalable enterprise deployments
- Provides increased choice for customers to modernize COBOL and C assets on distributed platforms

***WXTR provides capability to easily integrate transactional workloads in COBOL, C and Java, reducing infrastructure complexity while enabling increased SOA adoption on distributed systems***

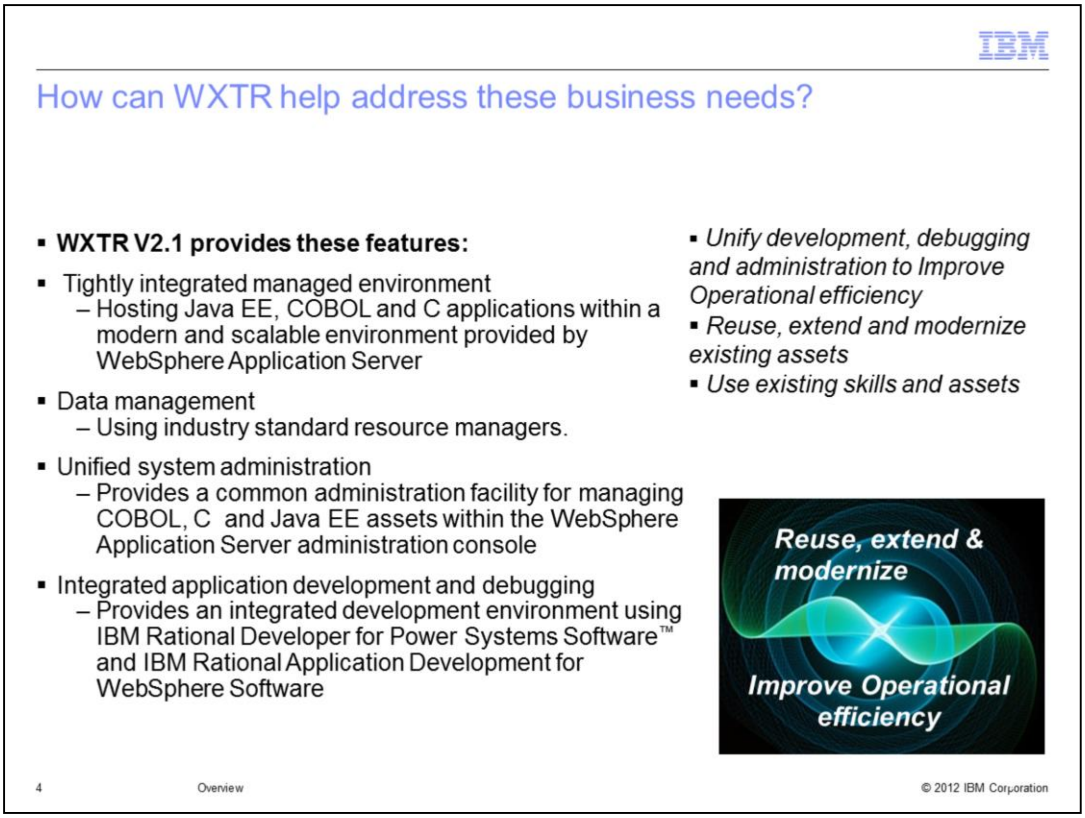
3 Overview © 2012 IBM Corporation

IBM WebSphere eXtended Transaction Runtime is the latest addition to the IBM Transaction Processing capabilities joining such superior solutions as CICS, Transaction Processing Facility on z/OS®, TXSeries, ALCS, WebSphere eXtreme Scale, and WebSphere Application Server to provide fast, scalable, and reliable transaction processing.

Many customers have invested time and effort in the development of business logic in CICS style COBOL applications and are looking to unlock the value of those applications and extend them using Java EE. By establishing WXTR as a core component of an enterprise IT architecture, you can have a highly responsive, tightly integrated composite application serving infrastructure for modern COBOL, C and Java workloads.

WXTR, along with IBM Rational® Application Development tools, delivers the capability to create and host COBOL and C business applications using CICS transactional services and extend them into a Java EE environment. It provides native runtime connectivity between Java EE and composite applications, significantly easing integration in an enterprise environment. Its tight integration with the system management capability of WebSphere Application Server enables easy deployment, administration, and optimization of services, significantly increasing administration efficiency unlike other competitor environments for managing mixed language transactional applications.

WXTR provides an increased choice in deployment options to progressively modernize your COBOL and C assets using CICS transactional services while standardizing IT infrastructure around WebSphere Application Server as a hosting platform on distributed systems, using extensive service-oriented architecture (SOA) enablement capability. This approach complements IBM's premier offering for modern application hosting on z/OS through products such as CICS Transaction Server V4 and WebSphere Application Server for z/OS, with WXTR providing a subset of capabilities to those found on System z®.



## How can WXTR help address these business needs?

- **WXTR V2.1 provides these features:**
- Tightly integrated managed environment
  - Hosting Java EE, COBOL and C applications within a modern and scalable environment provided by WebSphere Application Server
- Data management
  - Using industry standard resource managers.
- Unified system administration
  - Provides a common administration facility for managing COBOL, C and Java EE assets within the WebSphere Application Server administration console
- Integrated application development and debugging
  - Provides an integrated development environment using IBM Rational Developer for Power Systems Software™ and IBM Rational Application Development for WebSphere Software

- *Unify development, debugging and administration to Improve Operational efficiency*
- *Reuse, extend and modernize existing assets*
- *Use existing skills and assets*

4 Overview © 2012 IBM Corporation

WXTR provides a tightly integrated and managed environment to host composite applications in a modern platform provided by WebSphere Application Server. It allows you to manage and administer all your mixed language assets using a single WebSphere Application Server console. You can develop and deploy all your applications using one integrated application development environment provided by Rational. You can use Rational Developer for Power Systems Software to develop and debug applications in C and COBOL and Rational Application Developer for Java applications. This will help reduce infrastructure complexities and drive operational efficiency.

WXTR provides an execution environment to host modern COBOL and C business applications, enabling simplified interoperability between Java EE, COBOL and C workloads. This allows developers to focus on business logic rather than on how to inter-operate between composite assets, detect and recover from failures, and access data. It provides standard interfaces to invoke COBOL and C applications from Java. Java applications can connect through standard connection interfaces - Java Connector Architecture (JCA) or Service Component Architecture (SCA) - and invoke COBOL and C applications.

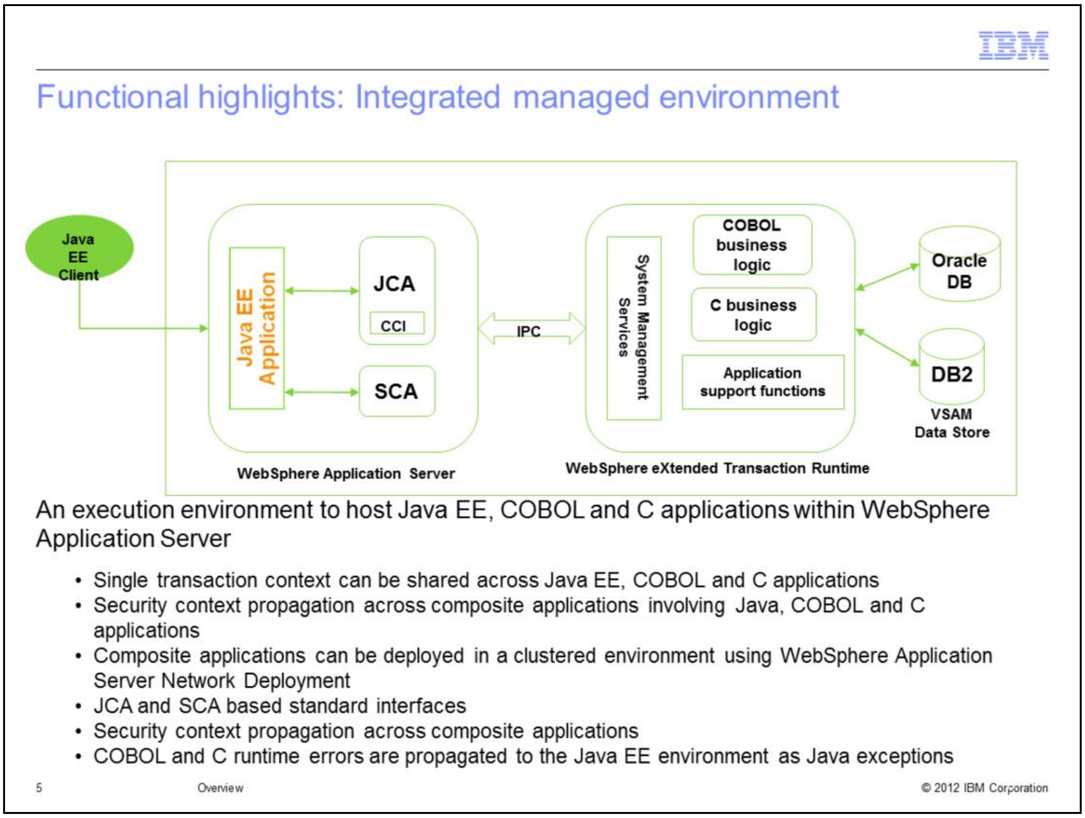
WXTR provides a common administration facility for managing both COBOL and Java EE assets from the WebSphere Application Server administrative console, reducing the learning curve for administration personnel and allowing them to deploy, administer, and optimize services. It also provides the capability for user groups with predefined privileges to run certain administrative tasks, thus reducing the dependence on administration personnel and improving their productivity.

WXTR enables COBOL and C Runtime library and application errors to be propagated as Java exceptions back to Java EE applications for further processing, simplifying development and integration effort and increasing developer productivity.

Using WXTR, mixed type data can be exchanged between C, COBOL and Java applications through the binding feature provided with IBM Rational Application Developer. Data can also be exchanged using COMMAREA in character format.

WXTR provides support for CICS data management facilities such as files and queues stored in a DB2® database. It also allows VSAM style access to data stored in DB2.

Rational development tools, along with WXTR, provide a modern application development experience across C, Java EE and COBOL applications. These tools let you develop, deploy, and debug applications within a single IDE. They also enable COBOL and C program inspection from within a Java EE application.



## Functional highlights: Integrated managed environment

Java EE Client
Java EE Application
JCA
CCI
SCA
IPC
System Management Services
COBOL business logic
C business logic
Application support functions
Oracle DB
DB2
VSAM Data Store
WebSphere Application Server
WebSphere eXtended Transaction Runtime

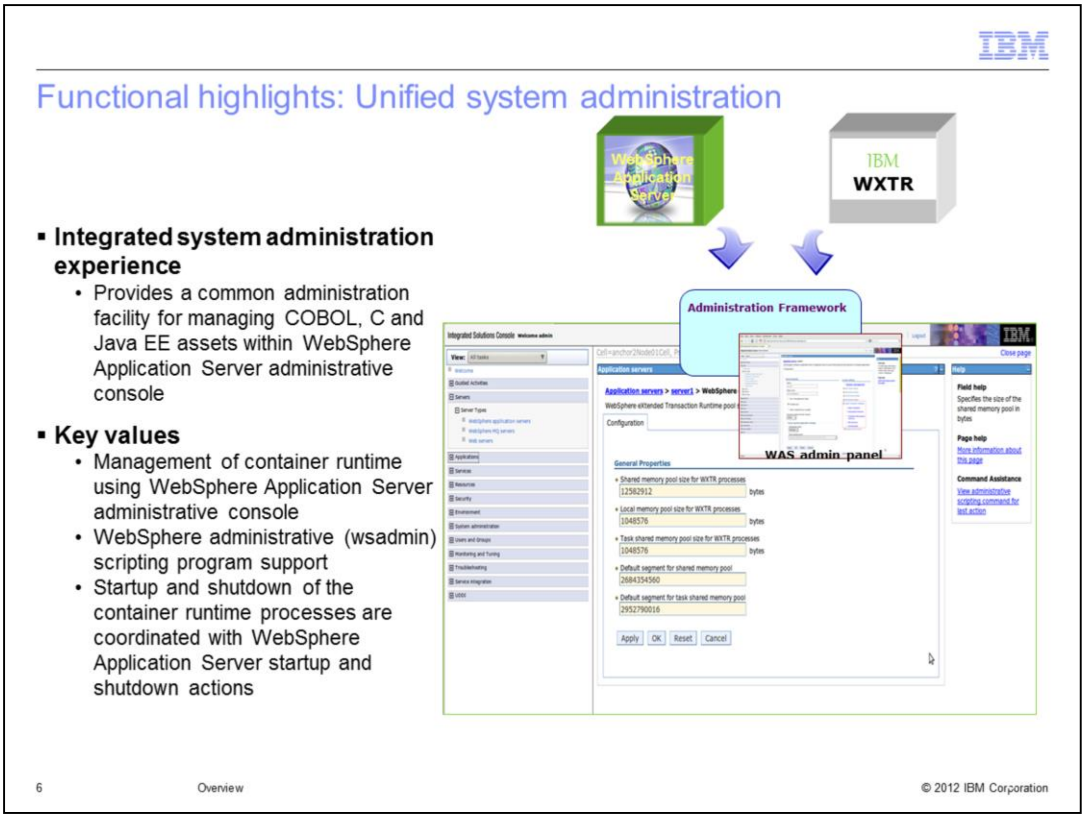
An execution environment to host Java EE, COBOL and C applications within WebSphere Application Server

- Single transaction context can be shared across Java EE, COBOL and C applications
- Security context propagation across composite applications involving Java, COBOL and C applications
- Composite applications can be deployed in a clustered environment using WebSphere Application Server Network Deployment
- JCA and SCA based standard interfaces
- Security context propagation across composite applications
- COBOL and C runtime errors are propagated to the Java EE environment as Java exceptions

5 Overview © 2012 IBM Corporation

This is the high level architecture of WXTR. The Java applications are hosted on WebSphere Application Server and they run in the JVM address space. COBOL and C applications are hosted on WXTR. The WXTR runtime itself is a set of UNIX® processes that run outside of the JVM process. Java application running in WebSphere Application Server can call a COBOL or C application using one of the two standard interfaces - JCA or SCA. The two runtimes (WXTR and the application server) are interconnected using an highly optimized proprietary intercommunication protocol.

The WXTR runtime supports DB2 and Oracle as resource managers. Along with other modules, WXTR has system administration and application support modules.



## Functional highlights: Unified system administration

- **Integrated system administration experience**
  - Provides a common administration facility for managing COBOL, C and Java EE assets within WebSphere Application Server administrative console
- **Key values**
  - Management of container runtime using WebSphere Application Server administrative console
  - WebSphere administrative (wsadmin) scripting program support
  - Startup and shutdown of the container runtime processes are coordinated with WebSphere Application Server startup and shutdown actions

Cell=anchor2Node01Cell, P

Application servers > server1 > WebSphere

WebSphere eXtended Transaction Runtime pool

General Properties

- Shared memory pool size for WXTR processes: 12582912 bytes
- Local memory pool size for WXTR processes: 1048576 bytes
- Task shared memory pool size for WXTR processes: 1048576 bytes
- Default segment for shared memory pool: 2684354560
- Default segment for task shared memory pool: 2952790016

Field help

Specifies the size of the shared memory pool in bytes

Page help

More information about this page

Command Assistance

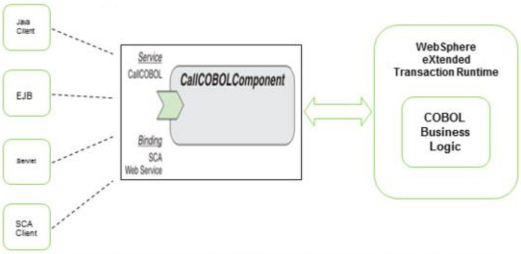
View administrative scripting command for last action

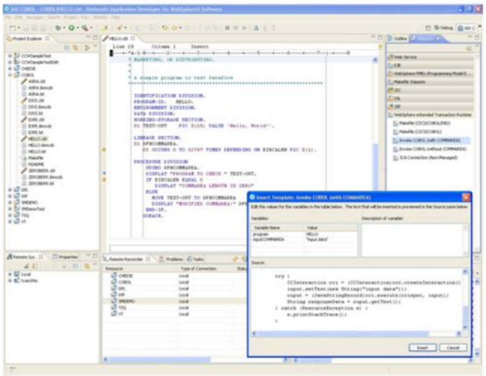
6 Overview © 2012 IBM Corporation

One of the pain points of managing composite applications in loosely coupled architecture is to manage different hosting environments. For example, Java assets can be hosted on a Java EE application server, COBOL and C assets can be hosted on TXSeries. The administration of these two environments are completely different and it is very inefficient to manage such an infrastructure. One of the key value propositions of WXTR is to simplify administration of composite assets using a single administration console. Most of the container runtime administration can be done from this console. There is also command line administration support using wsadmin commands. Startup and shutdown of WXTR processes are synchronized with WebSphere Application Server startup and shutdown. This screen capture shows the WXTR container administration using the WebSphere Application Server administration console.

## Functional highlights: Application development

- **Integrated application development and deployment experience using IBM Rational tools:**
  - IBM Rational Application Developer for WebSphere Software for developing Java EE applications
  - IBM Rational Developer for Power Systems Software for developing COBOL and C applications
- **Key values**
  - Build and run applications on the go
    - Automatic deployment of COBOL and C applications
  - Mixed language debugging support across Java EE, COBOL and C applications
  - For example, Invoke COBOL program from Java EE, inspect the COBOL program and back to Java EE using the same IDE
  - Snippets and code templates provided for ease in development experience



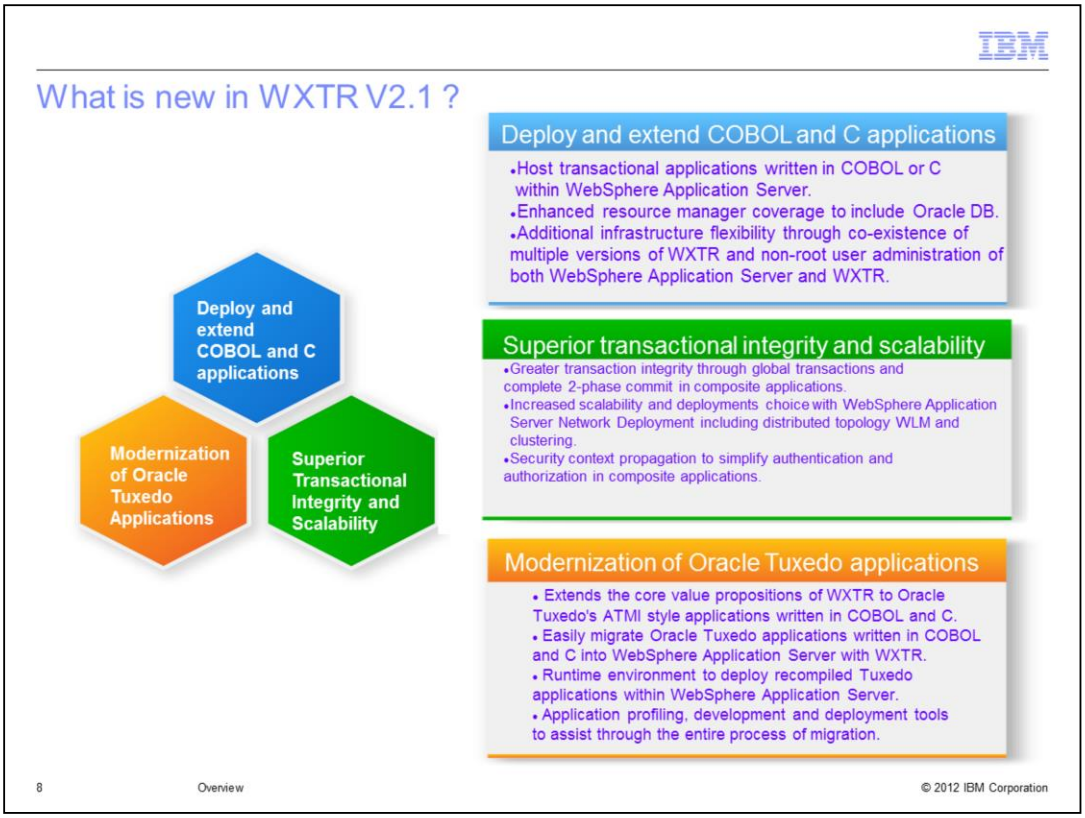


```
      * MARKETING, OR DISTRIBUTING.
      *
      *
      * A simple program to test Dataflow
      ****************************************************************

       IDENTIFICATION DIVISION.
       PROGRAM-ID.   HELLO.
       ENVIRONMENT DIVISION.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
       01 TEXT-OUT    PIC X(15) VALUE 'Hello, World'.

       LINKAGE SECTION.
       01 DFHCOMMAREA.
          03 OCCURS 0 TO 32767 TIMES DEPENDING ON EIBCALEN PIC X(1).

       PROCEDURE DIVISION
           USING DFHCOMMAREA.
           DISPLAY "PROGRAM TO CHECK " TEXT-OUT.
           IF EIBCALEN EQUAL 0
              DISPLAY "COMMAREA LENGTH IS ZERO"
           ELSE
             MOVE TEXT-OUT TO DFHCOMMAREA
             DISPLAY "MODIFIED COMMAREA:" DF
           END-IF.
          GOBACK.
```

```
try {
    CCInteraction cci = (CCInteraction)cxn.createInteraction();
    input.setText(new String("input data"));
    ouput = (JavaStringRecord)cci.execute(ccispec, input);
    String responseData = ouput.getText();
} catch (ResourceException e) {
    e.printStackTrace();
}
```

You can use a single integrated application development environment to develop and debug all composite applications. Rational Developer for Power Systems Software can be used to develop COBOL and C applications. Rational Application Developer can be used to develop Java EE applications. Debugging these application is almost seamless. If you have a Java application that calls COBOL, you start debugging Java, step into COBOL and come back to Java again all using the same IDE.

To help make writing applications easier, WXTR provides a lot of scaffolding code in the form of code snippets and templates.



## What is new in WXTR V2.1 ?

Deploy and extend COBOL and C applications

Modernization of Oracle Tuxedo Applications

Superior Transactional Integrity and Scalability

### Deploy and extend COBOL and C applications

- Host transactional applications written in COBOL or C within WebSphere Application Server.
- Enhanced resource manager coverage to include Oracle DB.
- Additional infrastructure flexibility through co-existence of multiple versions of WXTR and non-root user administration of both WebSphere Application Server and WXTR.

### Superior transactional integrity and scalability

- Greater transaction integrity through global transactions and complete 2-phase commit in composite applications.
- Increased scalability and deployments choice with WebSphere Application Server Network Deployment including distributed topology WLM and clustering.
- Security context propagation to simplify authentication and authorization in composite applications.

### Modernization of Oracle Tuxedo applications

- Extends the core value propositions of WXTR to Oracle Tuxedo's ATMI style applications written in COBOL and C.
- Easily migrate Oracle Tuxedo applications written in COBOL and C into WebSphere Application Server with WXTR.
- Runtime environment to deploy recompiled Tuxedo applications within WebSphere Application Server.
- Application profiling, development and deployment tools to assist through the entire process of migration.

8 Overview

This slide lists the new capabilities and features introduced in WXTR version 2.1. The features are broadly classified into three categories:

First, deploy and extend COBOL and C applications.

The language support has been extended to support C applications in WXTR. You can now write applications in both C and COBOL.

Resource manager support has been extended to Oracle along with DB2.

Multiple versions and instances of WXTR can coexist on the same server, providing scalability and version to version migration. (This was a restriction on V1.0.)

Second, superior transactional integrity and scalability.

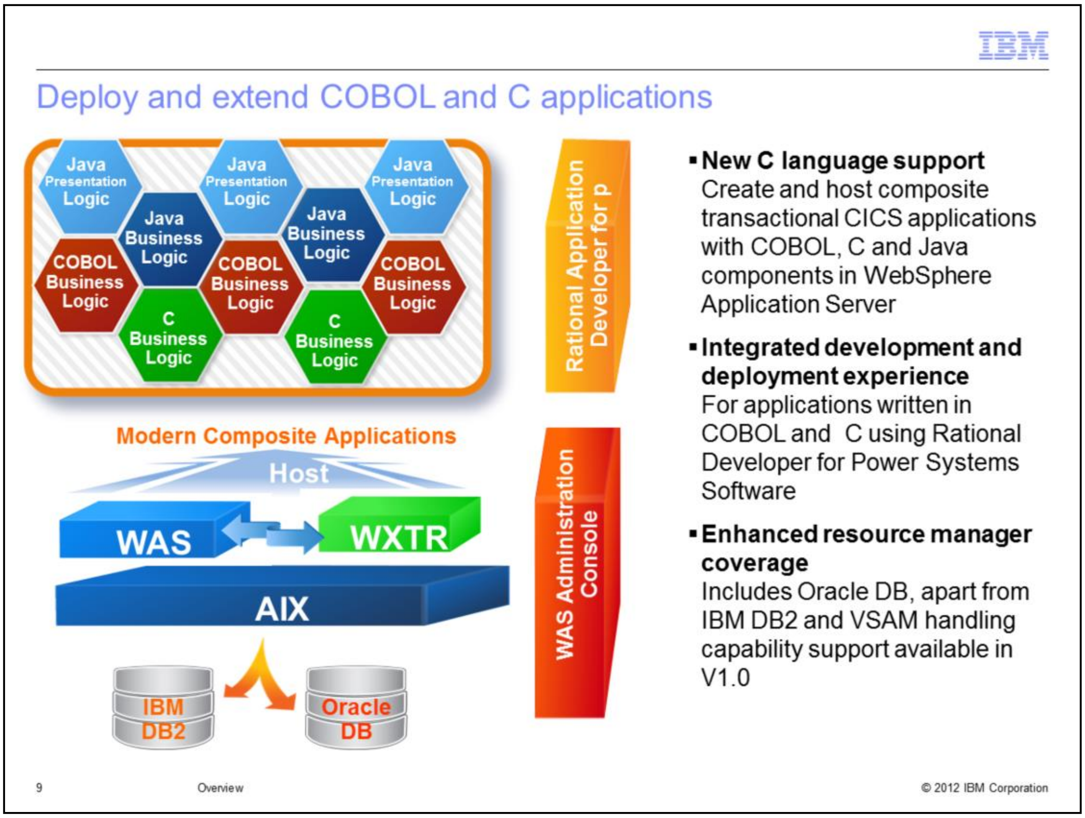
Composite applications can now share the same global transaction context with full two-phase commit protocol support.

Now with WebSphere Application Server Network Deployment support, applications can scale both horizontally and vertically.

A new feature has been added to propagate security context to WXTR. Authentication happens in WebSphere Application Server and the ID is propagated to WXTR. WXTR does the authorization for the resources that are defined with WXTR.

Third, modernization of Oracle Tuxedo applications.

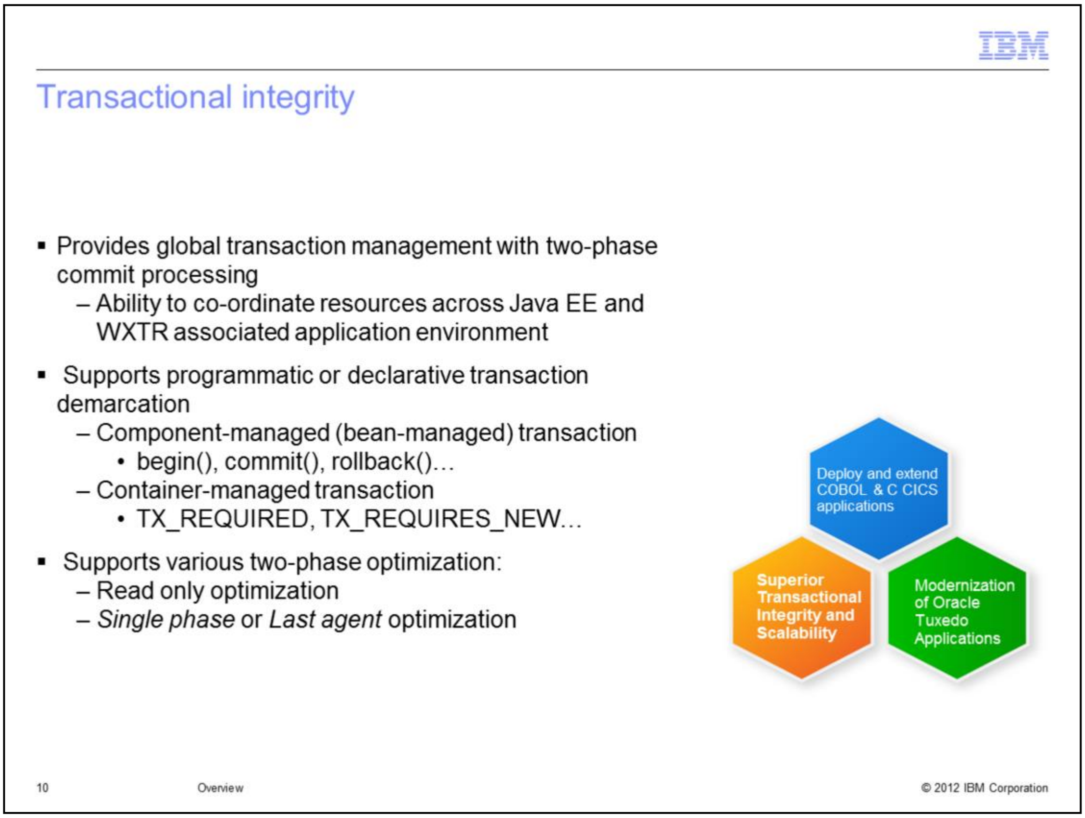
This is a feature pack that is released with WXTR V2.1 that enables almost seamless migration of Oracle Tuxedo C and COBOL ATMI applications.



## Deploy and extend COBOL and C applications

- **New C language support**
  Create and host composite transactional CICS applications with COBOL, C and Java components in WebSphere Application Server
- **Integrated development and deployment experience**
  For applications written in COBOL and C using Rational Developer for Power Systems Software
- **Enhanced resource manager coverage**
  Includes Oracle DB, apart from IBM DB2 and VSAM handling capability support available in V1.0

One of the key capabilities introduced in WXTR V2.1 is the C language support. This lets you host business applications, both CICS and non-CICS style, that are written in C. The same theme of simplified infrastructure holds good for developing and deploying C applications. You can use Rational Developer for Power Systems Software to develop and deploy both COBOL and C applications. Resource manager coverage has been extended to include Oracle database, along with BD2. You can configure Oracle DB as a resource manager, but the VSAM capabilities are restricted to DB2 only.



## Transactional integrity

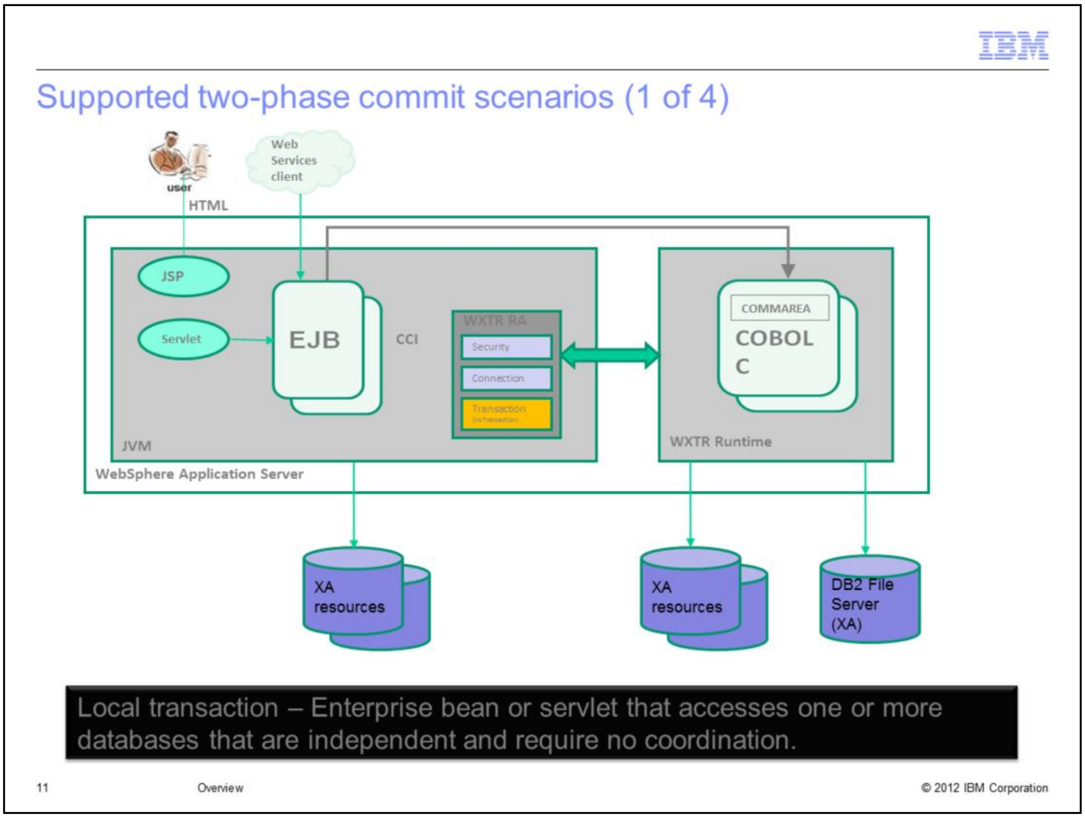
- Provides global transaction management with two-phase commit processing
  - Ability to co-ordinate resources across Java EE and WXTR associated application environment
- Supports programmatic or declarative transaction demarcation
  - Component-managed (bean-managed) transaction
    - begin(), commit(), rollback()…
  - Container-managed transaction
    - TX_REQUIRED, TX_REQUIRES_NEW…
- Supports various two-phase optimization:
  - Read only optimization
  - *Single phase* or *Last agent* optimization

Deploy and extend COBOL & C CICS applications
Superior Transactional Integrity and Scalability
Modernization of Oracle Tuxedo Applications

10 Overview © 2012 IBM Corporation

The transactional integrity feature enables sharing of transaction context across composite applications. For example, if an application has some part of business logic written in Java and some in COBOL with a requirement to participate in a single transaction, it is now possible with this feature. WebSphere Application Server will co-ordinate the entire transaction across multiple languages and resources.

Support is provided for both programmatic and declarative transaction demarcation. It also has support for various two-phase commit protocol optimizations. Certain scenarios do not require complete two-phase support; for example, if there are two resources participating in a transaction and one of them is a read only resource, there is no need to flow a “prepare” command to that resource. Similarly, if a resource is the last participant in a transaction there is no need to flow a “prepare” command.

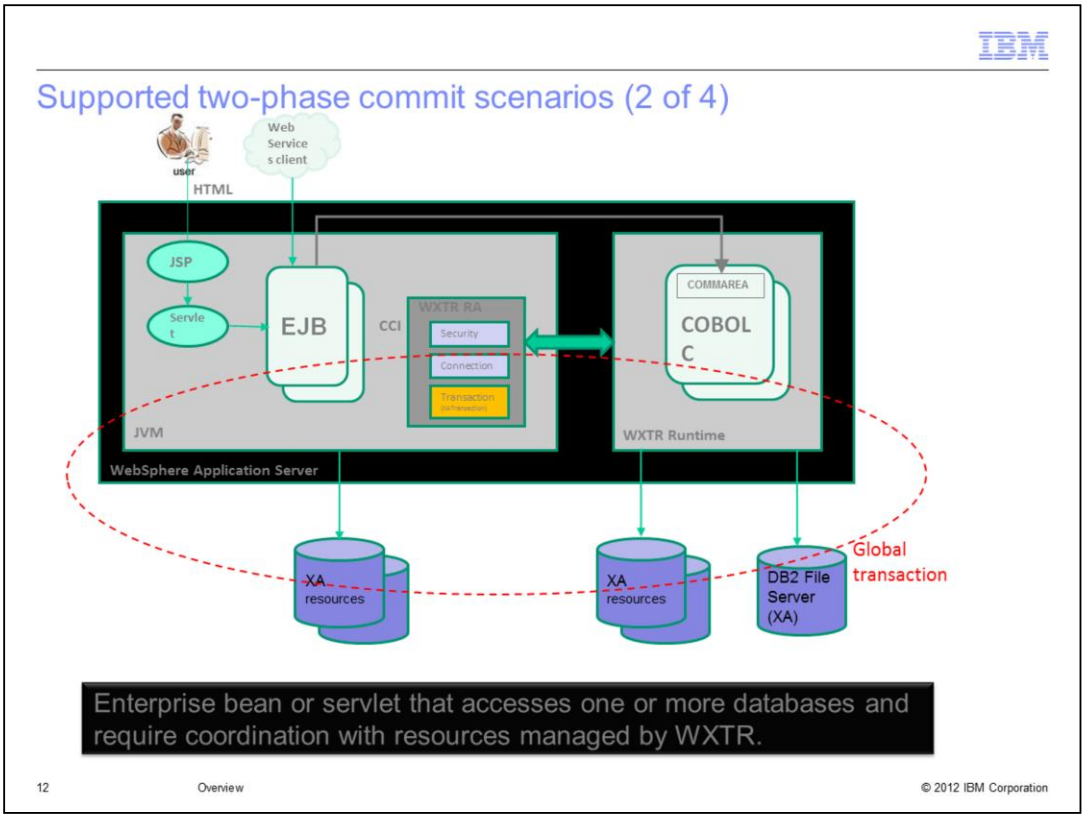


## Supported two-phase commit scenarios (1 of 4)

Local transaction – Enterprise bean or servlet that accesses one or more databases that are independent and require no coordination.

11 Overview © 2012 IBM Corporation

This slide depicts a scenario for a local transaction. The EJB part of the transaction is local to WebSphere Application Server and COBOL part of the transaction is local to WXTR. The two business logic does not participate in a global transaction. Both EJB and COBOL access different databases and do not need any coordination.

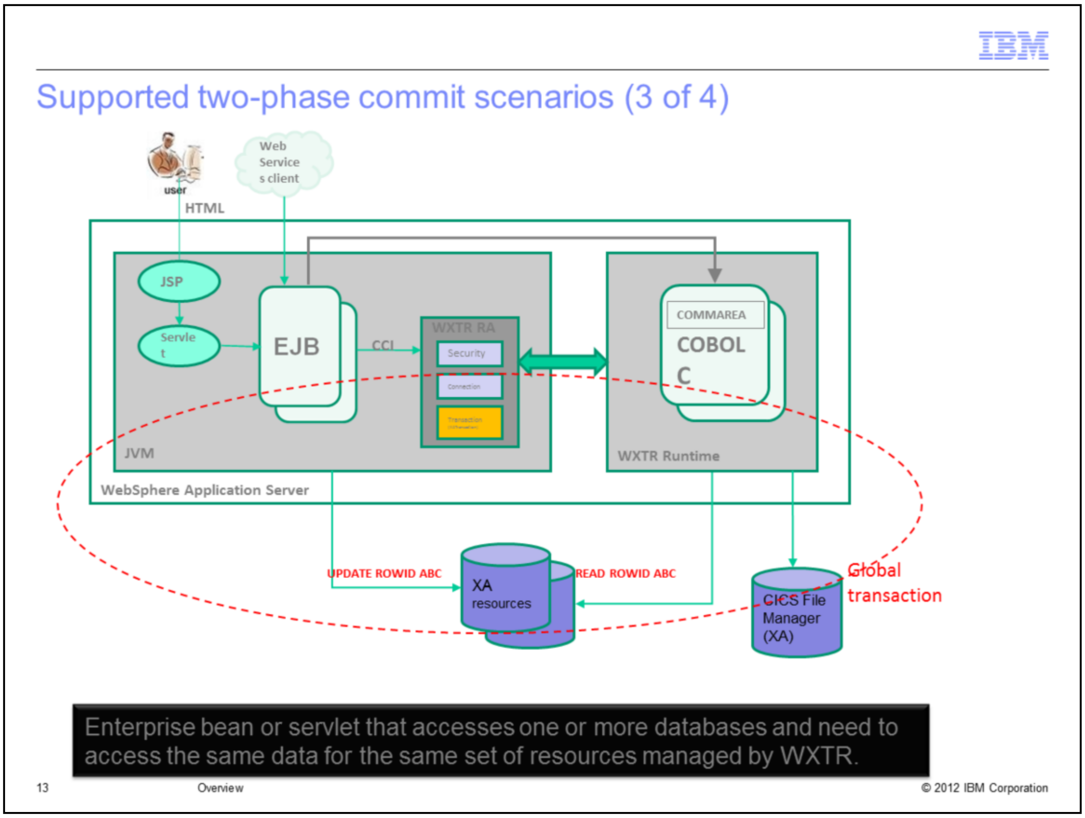


## Supported two-phase commit scenarios (2 of 4)

Enterprise bean or servlet that accesses one or more databases and require coordination with resources managed by WXTR.

12 Overview © 2012 IBM Corporation

This slide depicts a scenario that requires global transaction support. The business logic is spread across an EJB running in WebSphere Application Server and a COBOL application running in WXTR. COBOL applications access one or more databases managed by WXTR. The COBOL application is called from EJB business logic, which is transactional. This requires two phase co-ordination with resources managed by WXTR. Both COBOL and EJB business logic become a single logical unit of work, a global transaction.

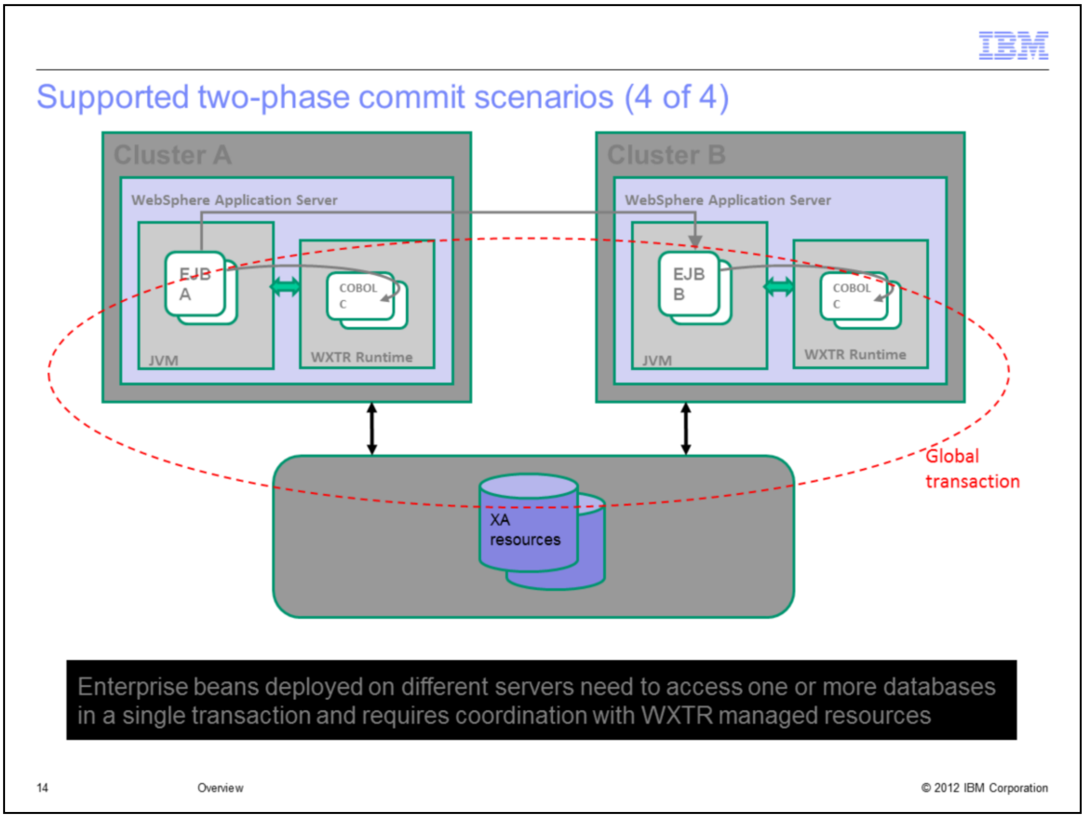


## Supported two-phase commit scenarios (3 of 4)

Enterprise bean or servlet that accesses one or more databases and need to access the same data for the same set of resources managed by WXTR.

13 Overview © 2012 IBM Corporation

This is a global transaction scenario where both EJB and COBOL applications access the same database and have a requirement to update or access the same record of the same table in the database. This is achieved by using the lock sharing feature provided by DB2, where the DB2 allows two transaction branches of the same global transaction to share locks on the same data, thus allowing updates to the same record in the database.

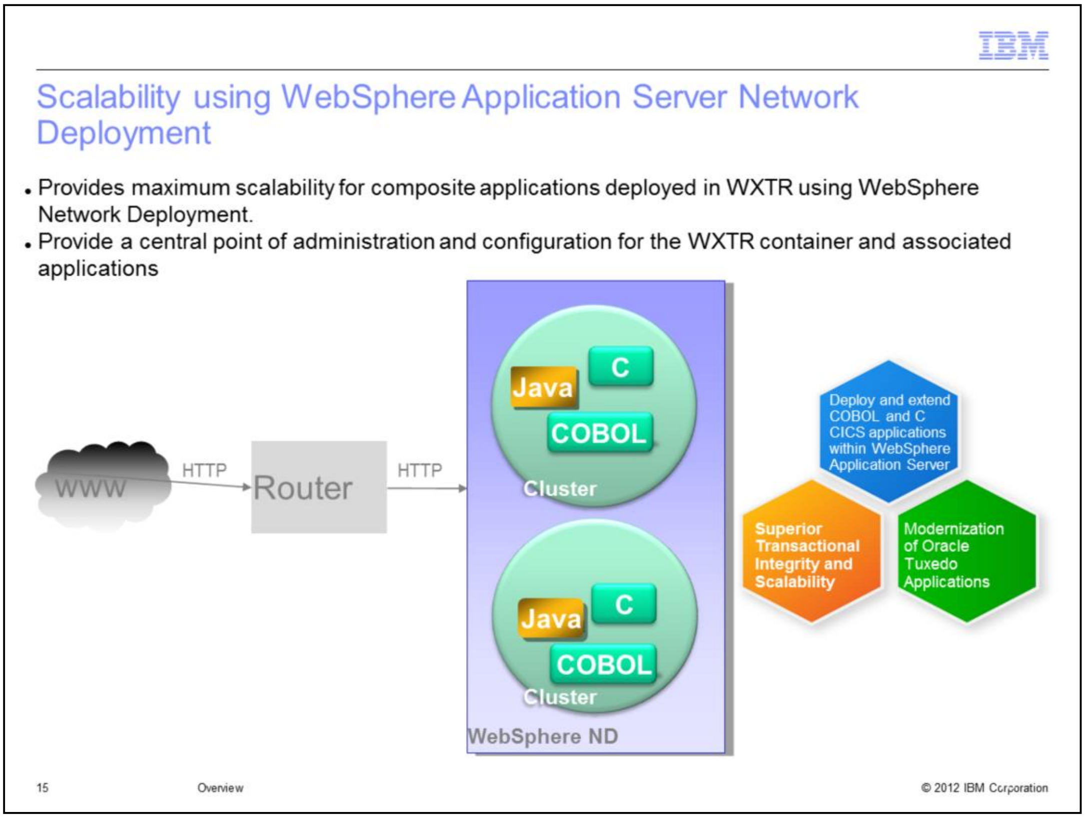


## Supported two-phase commit scenarios (4 of 4)

Enterprise beans deployed on different servers need to access one or more databases in a single transaction and requires coordination with WXTR managed resources

14 Overview © 2012 IBM Corporation

This is a global transaction scenario in clustered environment. In this, EJB A deployed on WebSphere Application Server in cluster A invokes the same business logic of EJB B deployed on WebSphere Application Server in cluster B. Along with this, both EJB A and EJB B invoke a COBOL application deployed in the WXTR container of their respective clusters. This results in a coordination requirement in both WXTR containers in both clusters for the resources managed by WXTR.



## Scalability using WebSphere Application Server Network Deployment

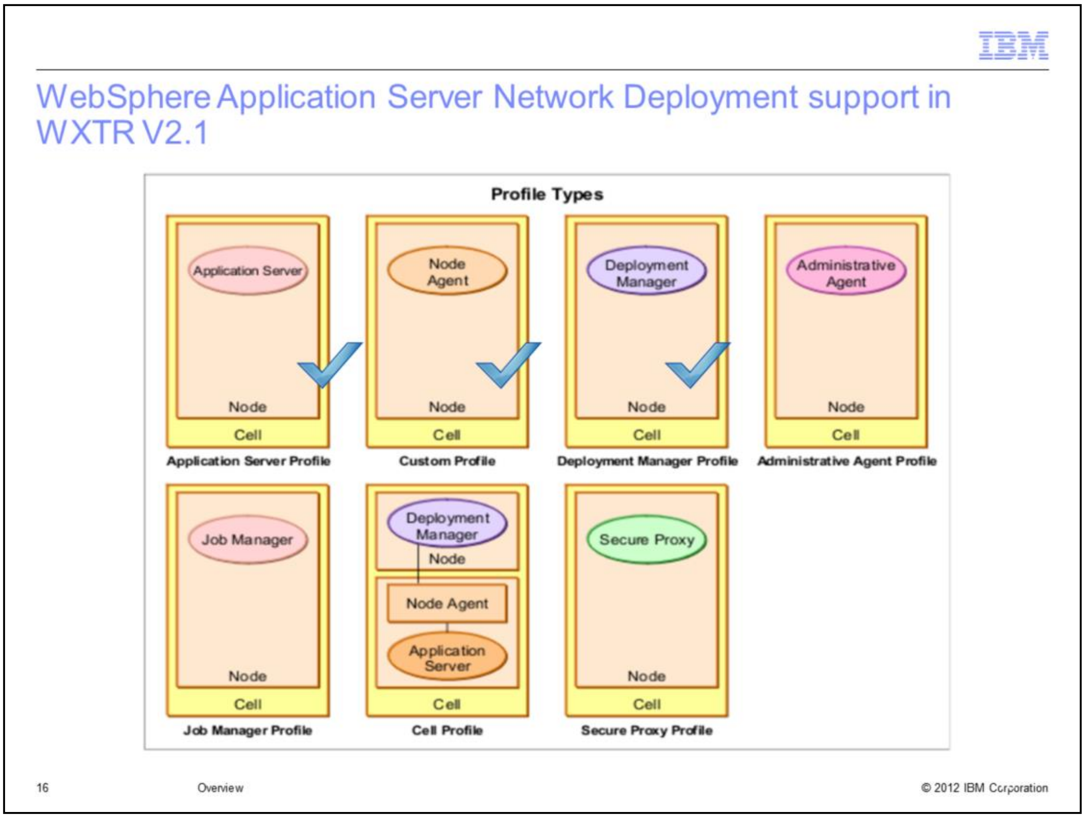
- Provides maximum scalability for composite applications deployed in WXTR using WebSphere Network Deployment.
- Provide a central point of administration and configuration for the WXTR container and associated applications

WWW
HTTP
Router
HTTP
Java
C
COBOL
Cluster
Java
C
COBOL
Cluster
WebSphere ND
Deploy and extend COBOL and C CICS applications within WebSphere Application Server
Superior Transactional Integrity and Scalability
Modernization of Oracle Tuxedo Applications

15 Overview © 2012 IBM Corporation

WXTR V1.0 supported stand-alone profiles in both WebSphere Application Server base and Network Deployment. WXTR V2.1 supports other profiles in WebSphere Application Server Network Deployment. For example, deployment manager profile, custom profile and so on. This not only lets you to create clustered topologies, but you can also manage them from a single location using deployment manager.

Composite applications containing Java EE, COBOL, and C business logic can be deployed in a horizontal or vertical cluster of application servers configured with WXTR, and requests can be routed to them through any of the supported routing agents like IBM proxy server.



IBM

## WebSphere Application Server Network Deployment support in WXTR V2.1

Profile Types

Application Server
Node
Cell
Application Server Profile ✓

Node Agent
Node
Cell
Custom Profile ✓

Deployment Manager
Node
Cell
Deployment Manager Profile ✓

Administrative Agent
Node
Cell
Administrative Agent Profile

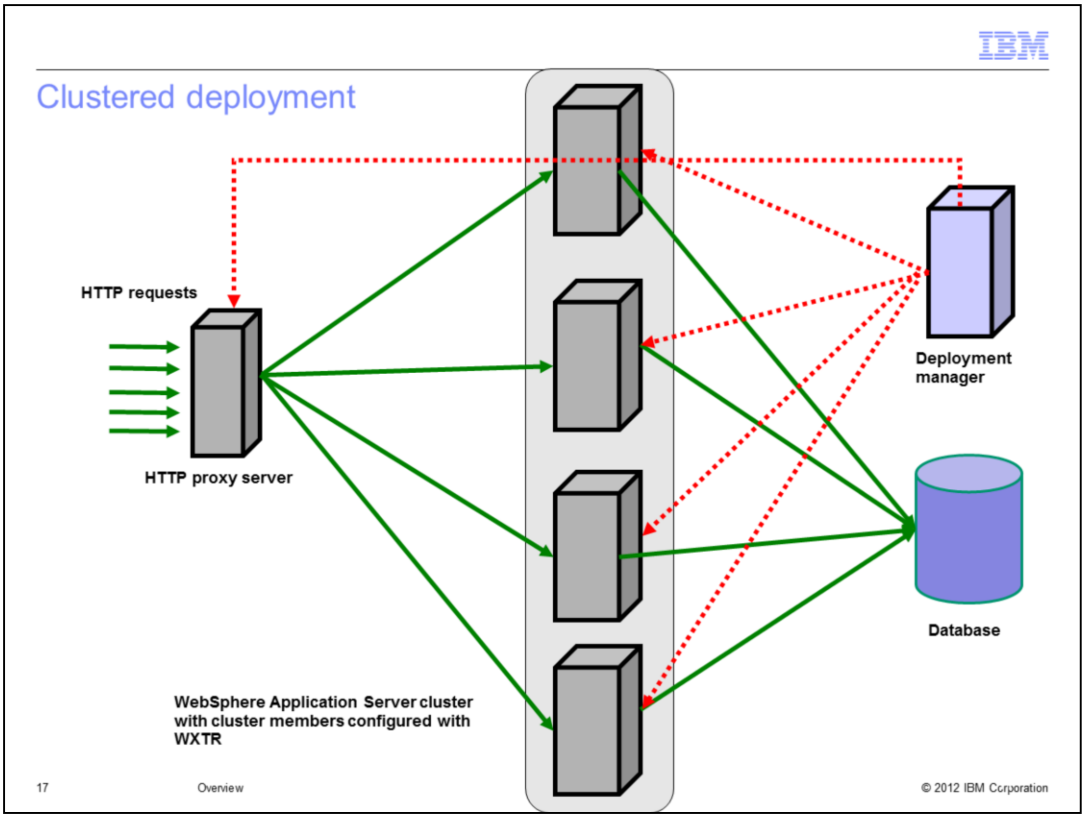
Job Manager
Node
Cell
Job Manager Profile

Deployment Manager
Node
Node Agent
Application Server
Cell
Cell Profile

Secure Proxy
Node
Cell
Secure Proxy Profile

16 Overview © 2012 IBM Corporation

A WebSphere Application Server profile defines the runtime environment. A profile includes all files that the server processes in the runtime environment and that you can change. WebSphere Application Server supports different types of profiles to cater to different requirements. However, WXTR does not support all profile types. The application server, custom, and deployment manager profiles are supported with WXTR.



## Clustered deployment

HTTP requests
HTTP proxy server
Deployment manager
Database
WebSphere Application Server cluster with cluster members configured with WXTR

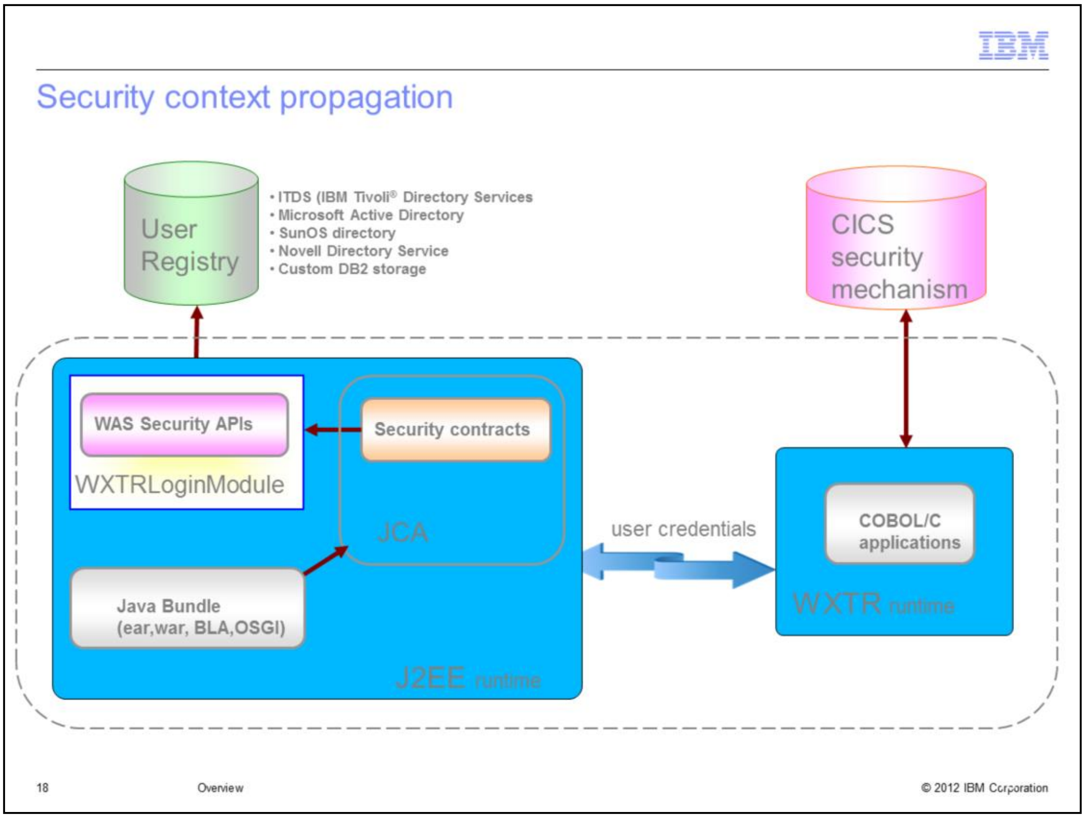
17 Overview © 2012 IBM Corporation

Production environments need clustering. This image shows a clustered deployment with four cluster members. HTTP requests are getting routed through the HTTP proxy server to these cluster members and load balancing happens in the proxy server.

The whole environment, including the proxy server, is being managed through deployment manager.

It can get more complex than this with multiple clusters.

There is a heartbeat mechanism in place for WXTR in WebSphere Application Server. Whenever a WXTR instance goes down abruptly, it will attempt to restart after recovery. Manual intervention might not be required if the instance comes up after recovery.



## Security context propagation

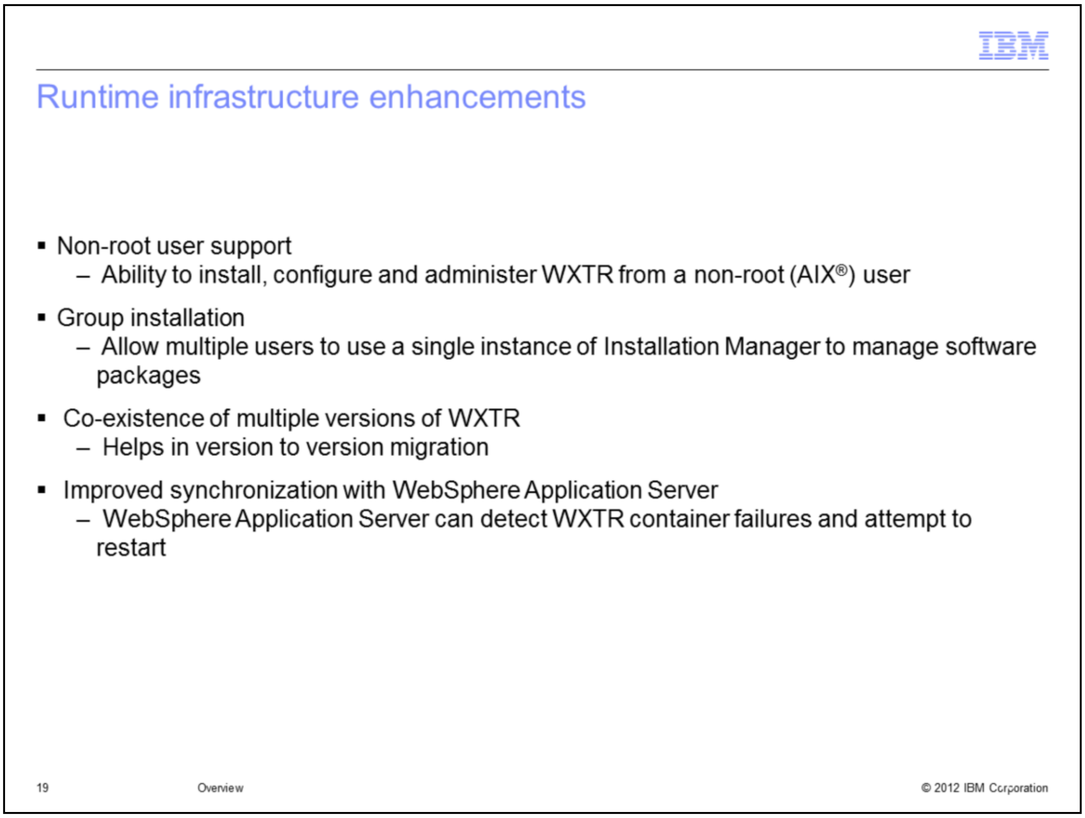
18 Overview © 2012 IBM Corporation

WebSphere Application Server provides facilities that allow you to secure administrative applications and services that are used to manage and configure a WebSphere environment. It allows you to secure applications running in the environment. These configuration activities are done separately, although they can share common settings.

In WXTR, focus is on application security, and not on administrative security. Application security infrastructure provides application isolation and requirements for authenticating users of the applications. Applications can be secured in a declarative manner or programmatically.

WebSphere Application Server will make use of the user registry for authenticating a user. The JCA adapter connecting the application server and WXTR runtime has implemented security contracts allowing propagation of credentials, which are attached with the authenticated user. WXTR provides a JAAS (Java Authentication and Authorization Services) module, which allows the propagation of user identity from WebSphere Application Server to WXTR.

The authentication of a user is carried out by WebSphere Application Server. The authorization of WXTR resources is done by WXTR. This type of user identity propagation provides end-to-end security and consistent accountability when the applications are composite with Java EE and COBOL applications.



## Runtime infrastructure enhancements

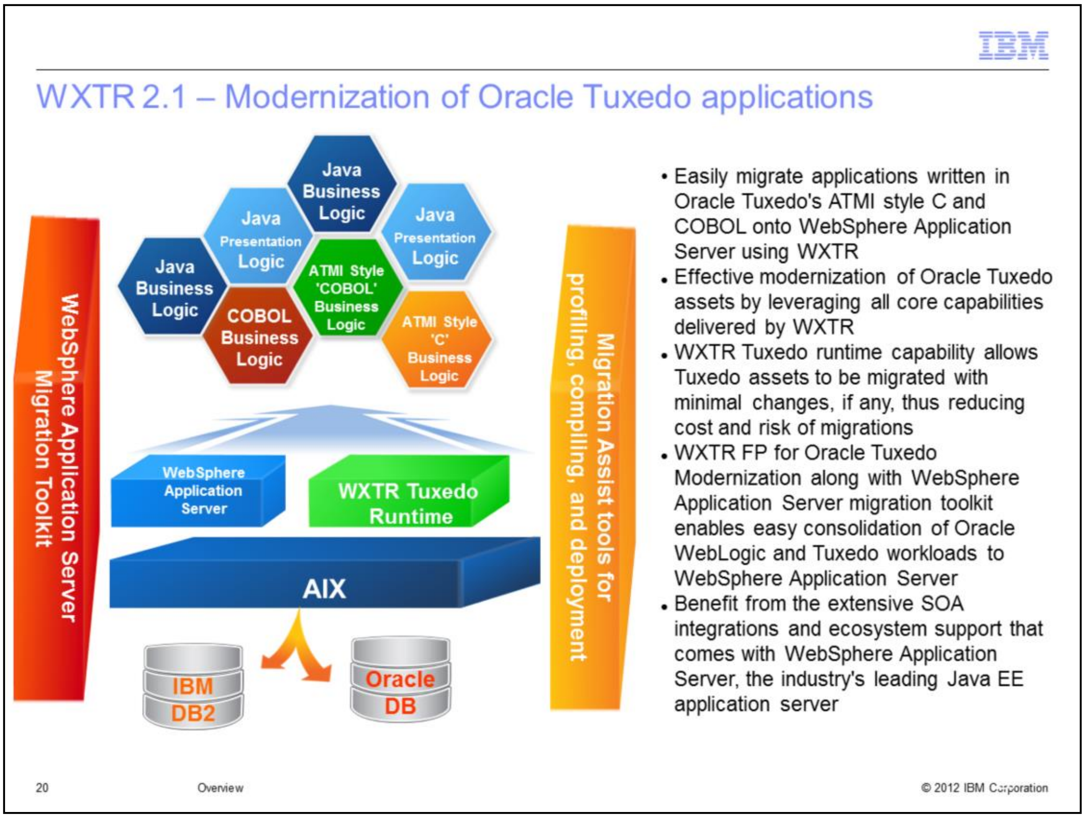
- Non-root user support
  - Ability to install, configure and administer WXTR from a non-root (AIX®) user
- Group installation
  - Allow multiple users to use a single instance of Installation Manager to manage software packages
- Co-existence of multiple versions of WXTR
  - Helps in version to version migration
- Improved synchronization with WebSphere Application Server
  - WebSphere Application Server can detect WXTR container failures and attempt to restart

WXTR V1.0 had a restriction that all administrative tasks had to be done by the root user. This restriction has been removed in the new version. With this, any WebSphere Application Server administrator can administer WXTR resources.

The group installation option is an installation enhancement that allows multiple users to use a single instance of Installation Manager to manage software packages.

Multiple versions and instances of WXTR can coexist on the same machine. With this feature, if you have multiple application server instances on a machine, you can install same number of WXTR instances. This feature also helps in achieving vertical scalability and version to version migration.

The synchronization of the WXTR container and the application server has been improved. WebSphere Application Server can now detect WXTR container failures and attempt restart. This ensures that WXTR is always available.



## WXTR 2.1 – Modernization of Oracle Tuxedo applications

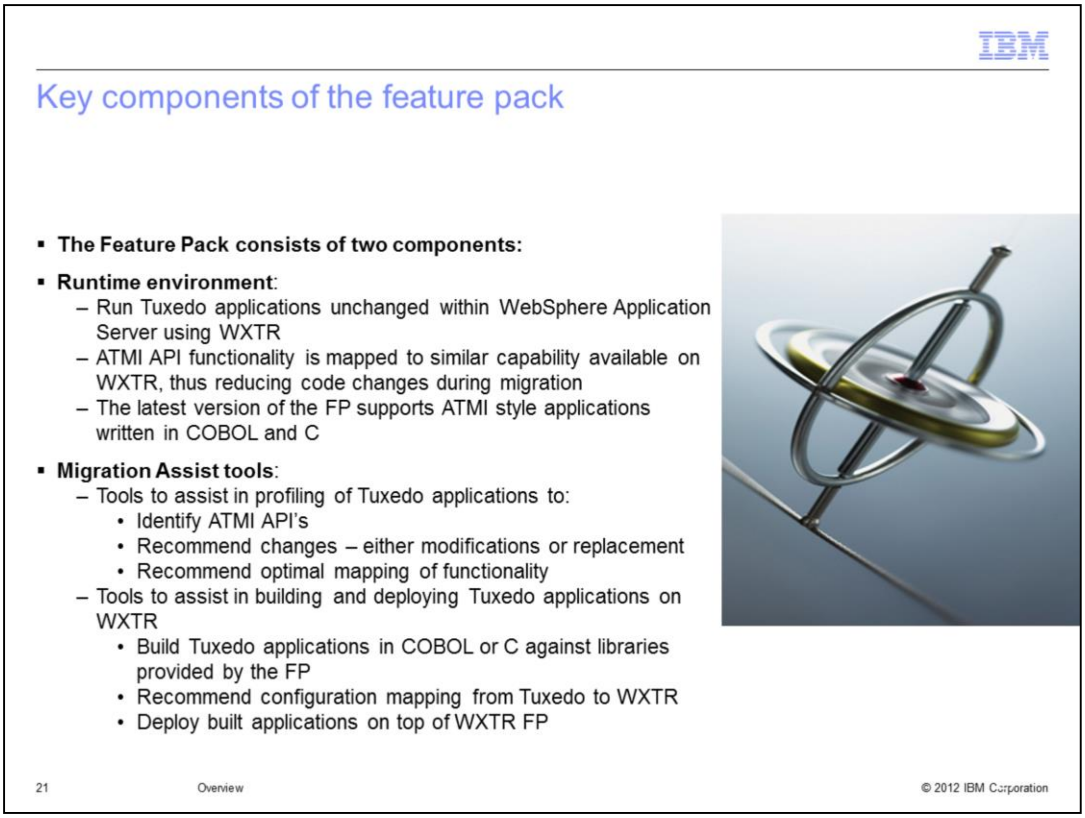
Java Business Logic
Java Presentation Logic
Java Presentation Logic
Java Business Logic
ATMI Style 'COBOL' Business Logic
COBOL Business Logic
ATMI Style 'C' Business Logic
WebSphere Application Server Migration Toolkit
Migration Assist tools for profiling, compiling, and deployment
WebSphere Application Server
WXTR Tuxedo Runtime
AIX
IBM DB2
Oracle DB

- Easily migrate applications written in Oracle Tuxedo's ATMI style C and COBOL onto WebSphere Application Server using WXTR
- Effective modernization of Oracle Tuxedo assets by leveraging all core capabilities delivered by WXTR
- WXTR Tuxedo runtime capability allows Tuxedo assets to be migrated with minimal changes, if any, thus reducing cost and risk of migrations
- WXTR FP for Oracle Tuxedo Modernization along with WebSphere Application Server migration toolkit enables easy consolidation of Oracle WebLogic and Tuxedo workloads to WebSphere Application Server
- Benefit from the extensive SOA integrations and ecosystem support that comes with WebSphere Application Server, the industry's leading Java EE application server

20 Overview © 2012 IBM Corporation

You can now migrate Oracle Tuxedo applications to WebSphere Application Server and WXTR using the Modernizing Oracle Tuxedo Applications Feature Pack. You can install this Feature Pack on WXTR V2.1. The Feature Pack allows easy migration of ATMI style C and COBOL Tuxedo applications. WXTR Tuxedo runtime capability allows Tuxedo assets to be migrated with minimal changes, if any, thus reducing cost and risk of migrations. This feature pack, along with WebSphere Application Server, enables easy consolidation of Oracle WebLogic and Tuxedo workloads to WebSphere Application Server.

The migrated applications benefit from the extensive SOA integrations and ecosystem support that comes with WebSphere Application Server.

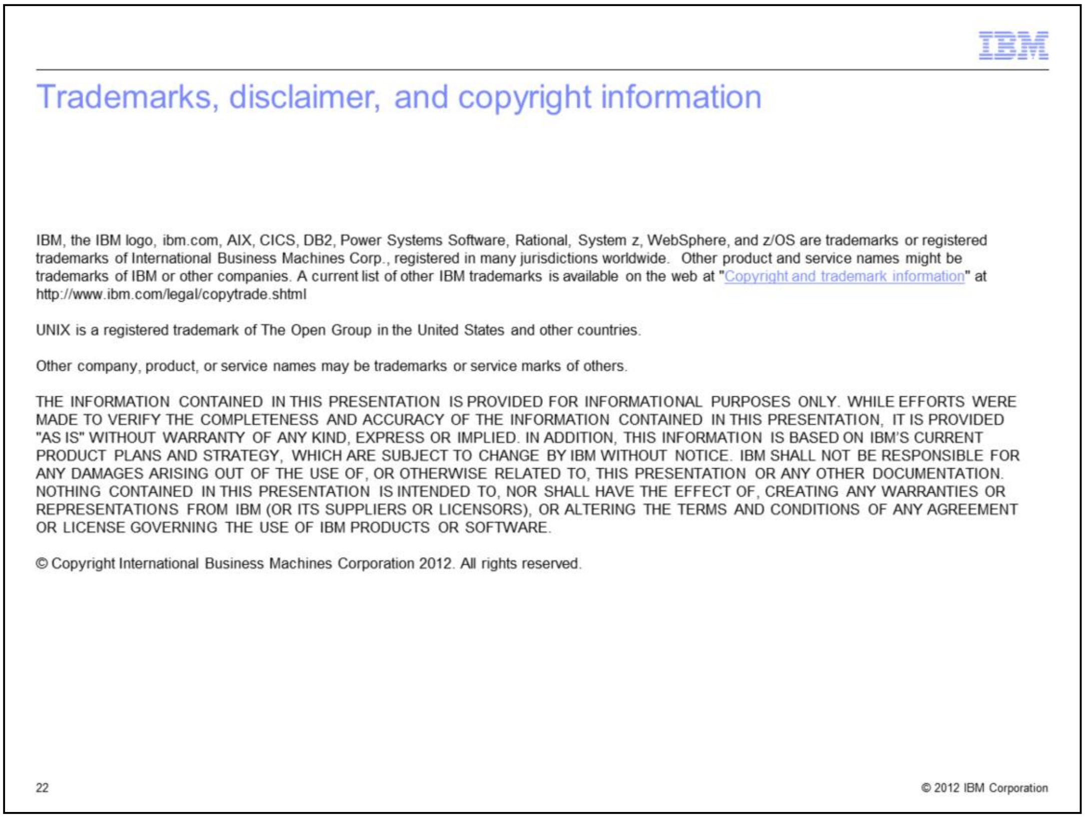


## Key components of the feature pack

- **The Feature Pack consists of two components:**
- **Runtime environment**:
  - Run Tuxedo applications unchanged within WebSphere Application Server using WXTR
  - ATMI API functionality is mapped to similar capability available on WXTR, thus reducing code changes during migration
  - The latest version of the FP supports ATMI style applications written in COBOL and C
- **Migration Assist tools**:
  - Tools to assist in profiling of Tuxedo applications to:
    - Identify ATMI API’s
    - Recommend changes – either modifications or replacement
    - Recommend optimal mapping of functionality
  - Tools to assist in building and deploying Tuxedo applications on WXTR
    - Build Tuxedo applications in COBOL or C against libraries provided by the FP
    - Recommend configuration mapping from Tuxedo to WXTR
    - Deploy built applications on top of WXTR FP

The feature pack consists of two components. The first component is the runtime environment, which provides API functionality and error handling. The API functionality is achieved by mapping tuxedo ATMI calls to one or more appropriate WXTR functions. The latest version of the runtime supports ATMI applications written in C and COBOL.

The second component is the set of tools. There are tools to profile an application to identify unsupported API's if there are any. This will assist the services team to suggest modifications to the applications. There are tools to build and deploy applications to WXTR. These set of tools are designed in such a way that you do not need to make changes to your make file.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, AIX, CICS, DB2, Power Systems Software, Rational, System z, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.