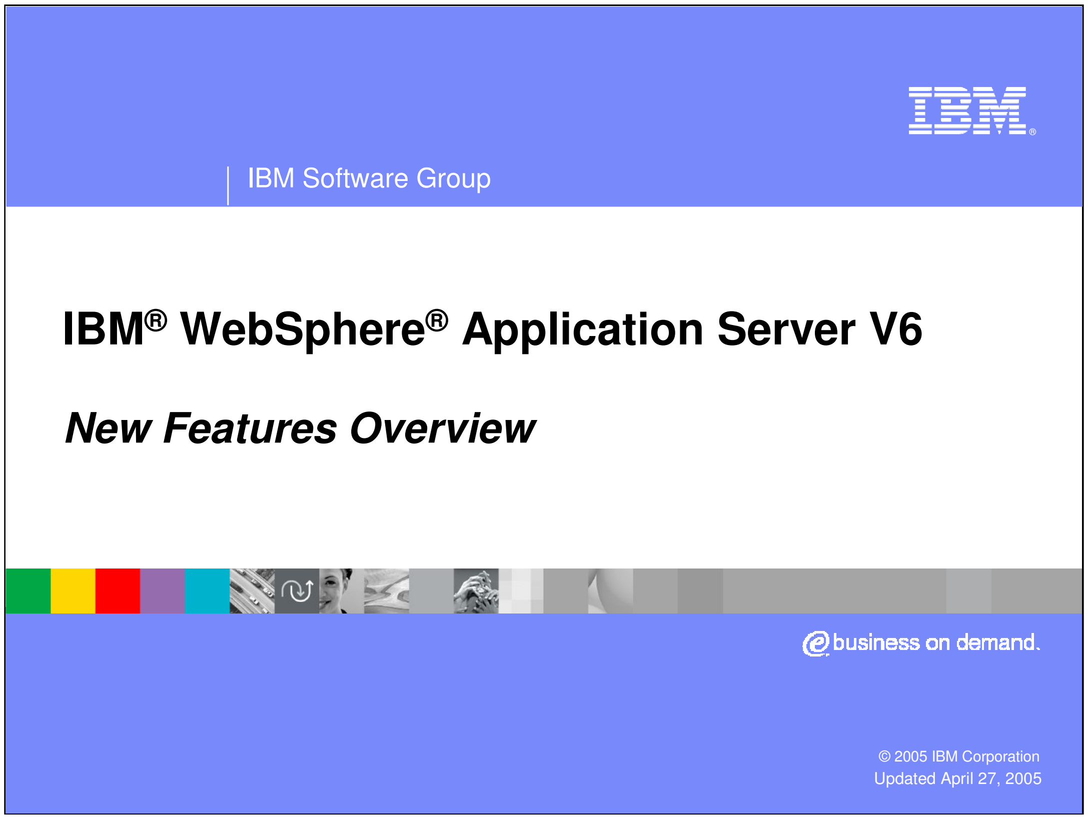


IBM Software Group

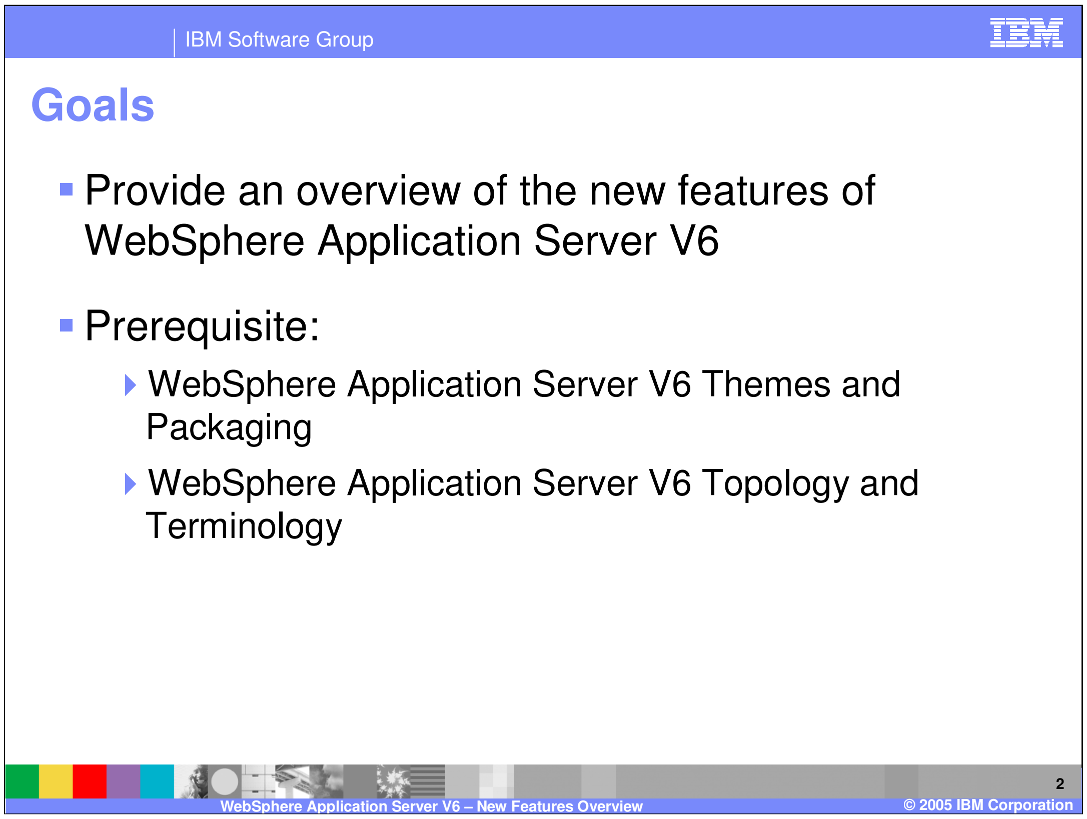
IBM® WebSphere® Application Server V6

New Features Overview

business on demand.

Updated April 27, 2005

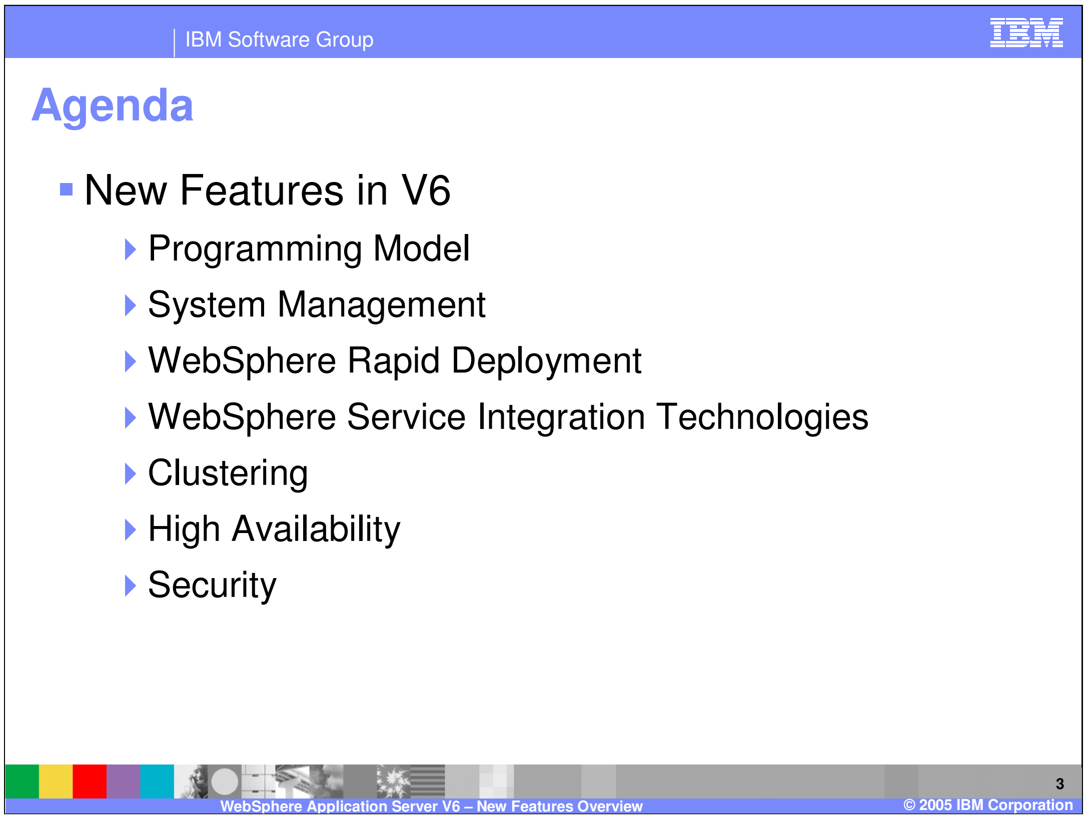
This presentation will provide an overview of the new features in WebSphere Application Server V6.



Goals

- Provide an overview of the new features of WebSphere Application Server V6
- Prerequisite:
  - WebSphere Application Server V6 Themes and Packaging
  - WebSphere Application Server V6 Topology and Terminology

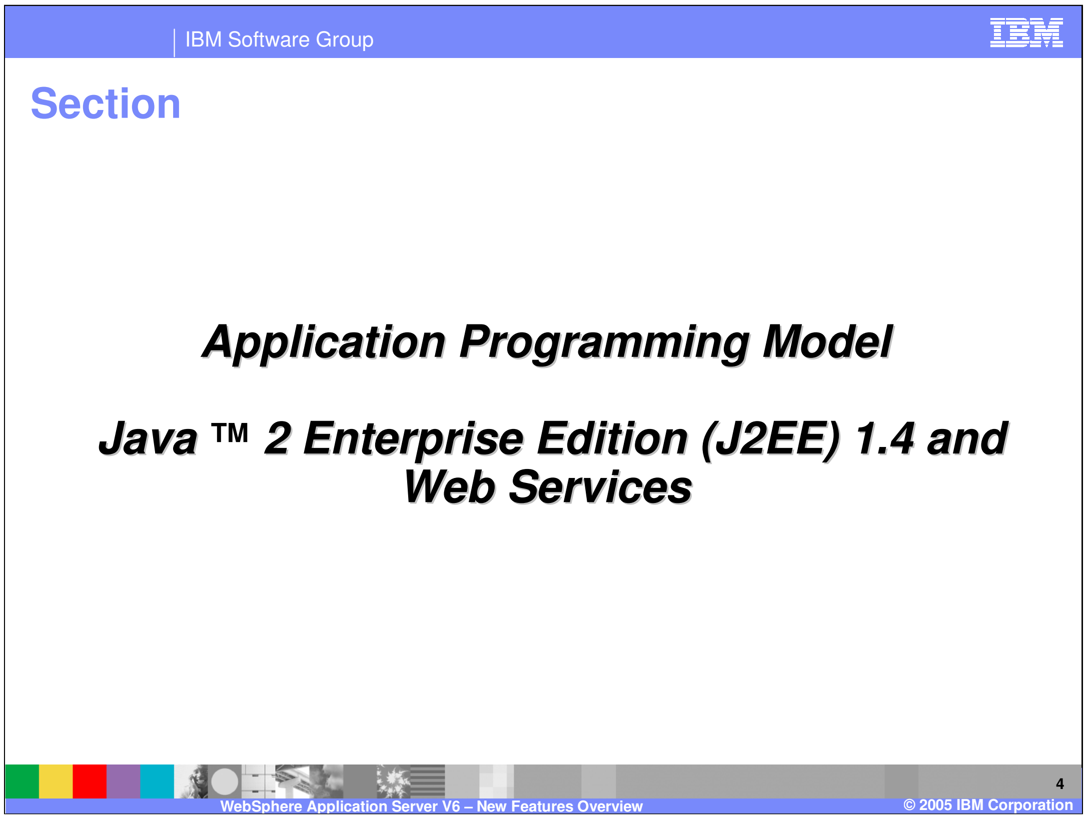
The goal of this presentation is to provide an overview of the new features available in WebSphere V6. To benefit the most from this presentation, first review the presentations “WebSphere Application Server V6 Themes and Packaging” and “WebSphere Application Server V6 Topology and Terminology”.



Agenda

- New Features in V6
  - Programming Model
  - System Management
  - WebSphere Rapid Deployment
  - WebSphere Service Integration Technologies
  - Clustering
  - High Availability
  - Security

The agenda for this presentation will cover new features in the Programming Model, System Management, WebSphere Rapid Deployment, WebSphere Service Integration Technologies, Clustering, High Availability and Security.


IBM Software Group

Section

Application Programming Model

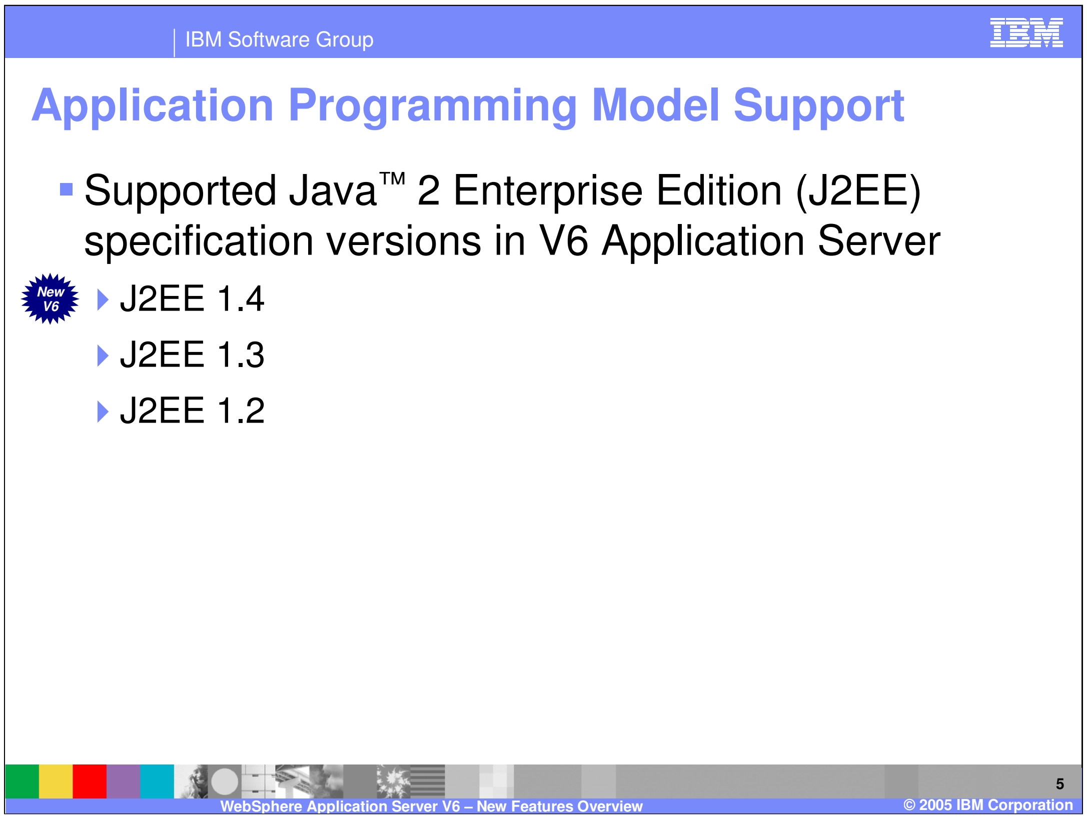
Java ™ 2 Enterprise Edition (J2EE) 1.4 and Web Services

4

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

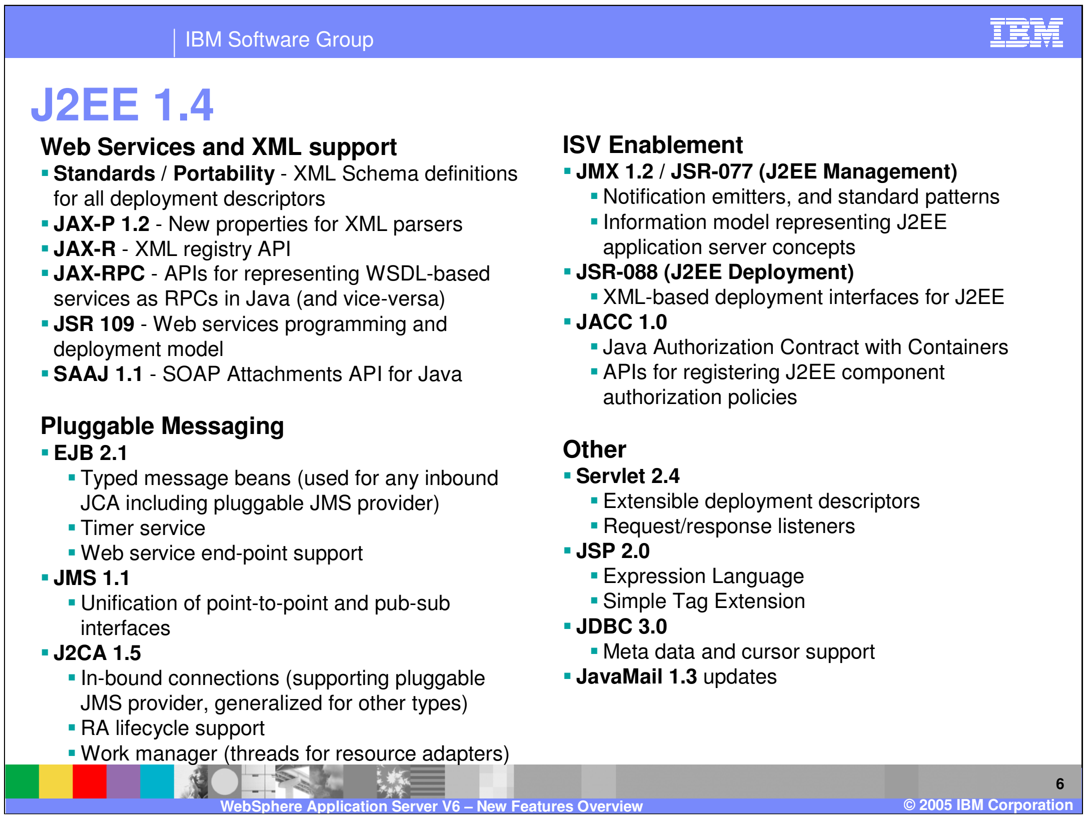
This section discusses the Application Programming Model which complies with the Java ™ 2 Enterprise Edition (J2EE) 1.4 specification.



Application Programming Model Support

- Supported Java™ 2 Enterprise Edition (J2EE) specification versions in V6 Application Server
  - New V6 J2EE 1.4
  - J2EE 1.3
  - J2EE 1.2

WebSphere Application Server V6 supports three levels of the J2EE specification. J2EE 1.4 is the new level supported with V6. Existing J2EE 1.2 and J2EE 1.3 applications will continue to run on V6.


IBM Software Group

J2EE 1.4

Web Services and XML support
- **Standards** / **Portability** - XML Schema definitions for all deployment descriptors
- **JAX-P 1.2** - New properties for XML parsers
- **JAX-R** - XML registry API
- **JAX-RPC** - APIs for representing WSDL-based services as RPCs in Java (and vice-versa)
- **JSR 109** - Web services programming and deployment model
- **SAAJ 1.1** - SOAP Attachments API for Java

Pluggable Messaging
- **EJB 2.1**
  - Typed message beans (used for any inbound JCA including pluggable JMS provider)
  - Timer service
  - Web service end-point support
- **JMS 1.1**
  - Unification of point-to-point and pub-sub interfaces
- **J2CA 1.5**
  - In-bound connections (supporting pluggable JMS provider, generalized for other types)
  - RA lifecycle support
  - Work manager (threads for resource adapters)

ISV Enablement
- **JMX 1.2** / **JSR-077 (J2EE Management)**
  - Notification emitters, and standard patterns
  - Information model representing J2EE application server concepts
- **JSR-088 (J2EE Deployment)**
  - XML-based deployment interfaces for J2EE
- **JACC 1.0**
  - Java Authorization Contract with Containers
  - APIs for registering J2EE component authorization policies

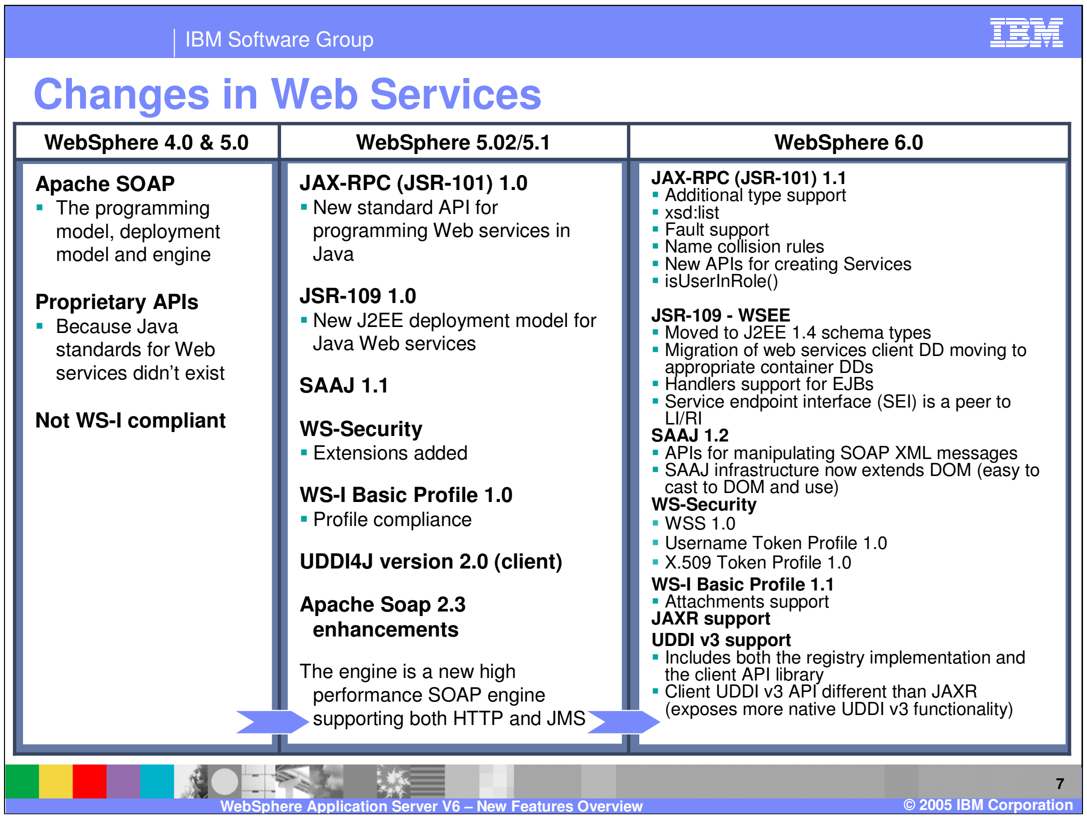
Other
- **Servlet 2.4**
  - Extensible deployment descriptors
  - Request/response listeners
- **JSP 2.0**
  - Expression Language
  - Simple Tag Extension
- **JDBC 3.0**
  - Meta data and cursor support
- **JavaMail 1.3** updates

6

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

The J2EE 1.4 specification encompasses several individual specifications. Many of the more notable ones are listed here.



IBM Software Group

Changes in Web Services

WebSphere 4.0 & 5.0

Apache SOAP
- The programming model, deployment model and engine

Proprietary APIs
- Because Java standards for Web services didn't exist

Not WS-I compliant

WebSphere 5.02/5.1

JAX-RPC (JSR-101) 1.0
- New standard API for programming Web services in Java

JSR-109 1.0
- New J2EE deployment model for Java Web services

SAAJ 1.1

WS-Security
- Extensions added

WS-I Basic Profile 1.0
- Profile compliance

UDDI4J version 2.0 (client)

Apache Soap 2.3 enhancements

The engine is a new high performance SOAP engine supporting both HTTP and JMS

WebSphere 6.0

JAX-RPC (JSR-101) 1.1
- Additional type support
- xsd:list
- Fault support
- Name collision rules
- New APIs for creating Services
- isUserInRole()

JSR-109 - WSEE
- Moved to J2EE 1.4 schema types
- Migration of web services client DD moving to appropriate container DDs
- Handlers support for EJBs
- Service endpoint interface (SEI) is a peer to LI/RI

SAAJ 1.2
- APIs for manipulating SOAP XML messages
- SAAJ infrastructure now extends DOM (easy to cast to DOM and use)

WS-Security
- WSS 1.0
- Username Token Profile 1.0
- X.509 Token Profile 1.0

WS-I Basic Profile 1.1
- Attachments support

JAXR support

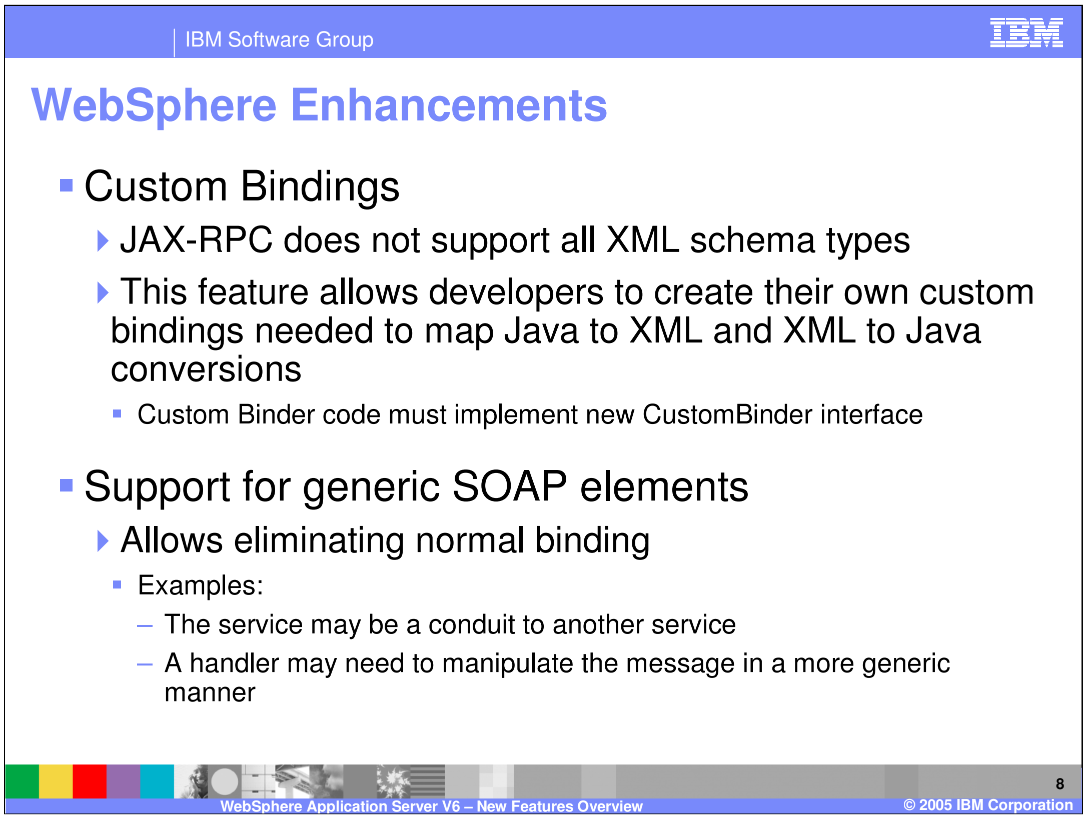
UDDI v3 support
- Includes both the registry implementation and the client API library
- Client UDDI v3 API different than JAXR (exposes more native UDDI v3 functionality)

7

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

Here you see the evolution of Web Services support across the WebSphere Application Server Versions 4.0 through 6.0. In Application Server V4, Apache SOAP was the Web Services programming model, deployment model, and engine. Additional specifications were supported in Application Server V5. Today, in Application Server V6, the strategic Web Services support complies with the specifications within J2EE 1.4 as well as other standards supporting interoperability and additional functionality.


IBM Software Group

WebSphere Enhancements

- Custom Bindings
  - JAX-RPC does not support all XML schema types
  - This feature allows developers to create their own custom bindings needed to map Java to XML and XML to Java conversions
    - Custom Binder code must implement new CustomBinder interface
- Support for generic SOAP elements
  - Allows eliminating normal binding
    - Examples:
      - The service may be a conduit to another service
      - A handler may need to manipulate the message in a more generic manner

8

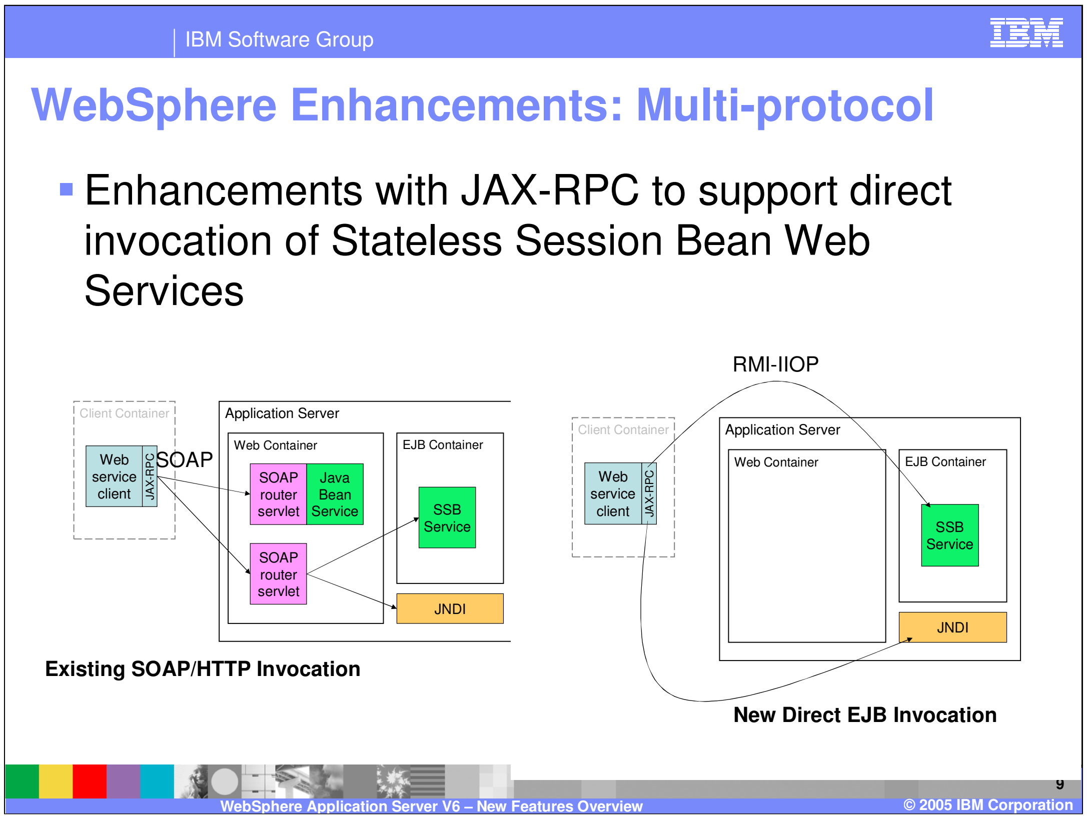
WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

Two enhancements available in WebSphere Application Server V6 are support for custom bindings and support for generic SOAP elements.

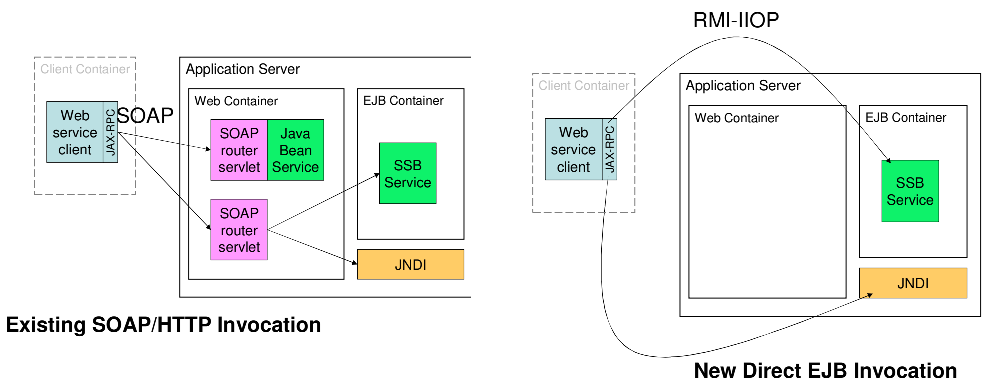
With support for custom bindings, you can provide your own code to perform the necessary binding from Java to XML and from XML to Java in those cases where the XML schema type is not supported by the Java API for XML-based Remote Procedure Call (JAX-RPC). To use this support, your code must implement the CustomBinder interface.

There are scenarios where you might have an intermediary service or gateway between the client and the end provider. In these cases, you may want to avoid any binding conversion in the intermediary service and let the end provider do the binding. With support for generic SOAP elements, you can do this. Avoiding an unnecessary conversion in the middle of the flow can help with performance.


IBM Software Group

WebSphere Enhancements: Multi-protocol

- Enhancements with JAX-RPC to support direct invocation of Stateless Session Bean Web Services



9

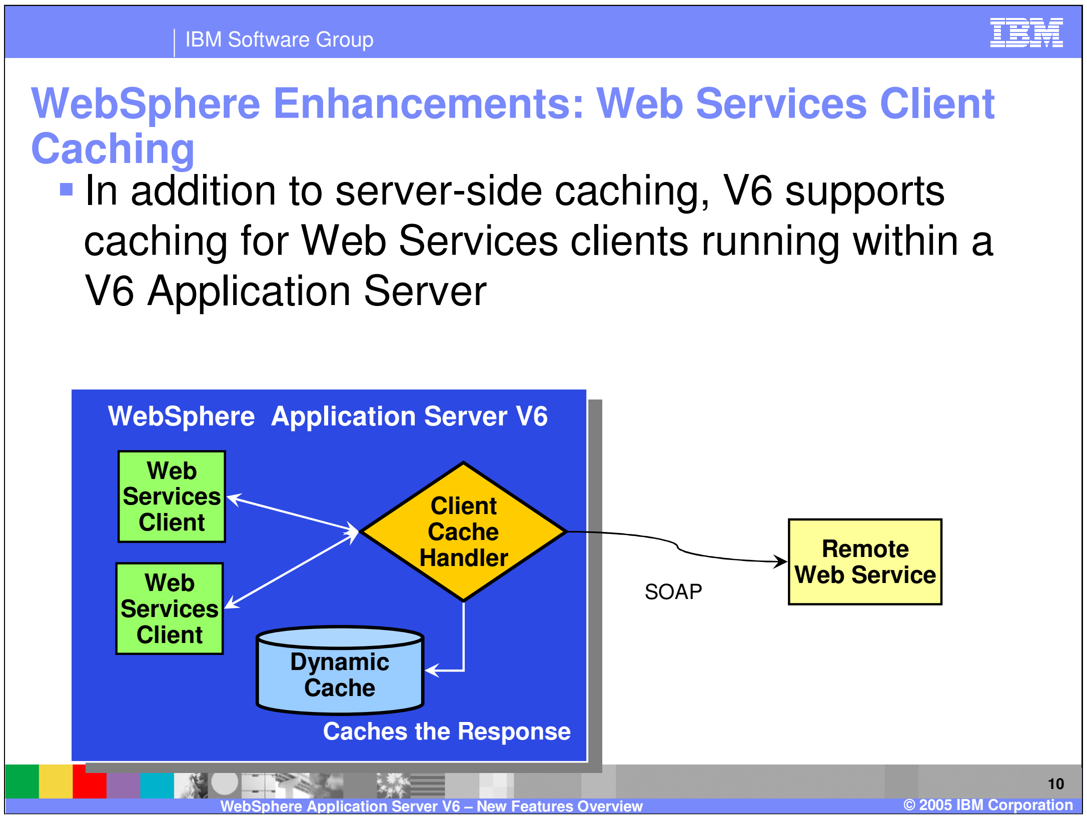
WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

Java API for XML-based Remote Procedure Call (JAX-RPC) is the Java standard API for invoking Web services through remote procedure calls. A transport is used by a programming language to communicate over the Internet. You can invoke Web services using protocols with the transport such as SOAP and Remote Method Invocation (RMI).

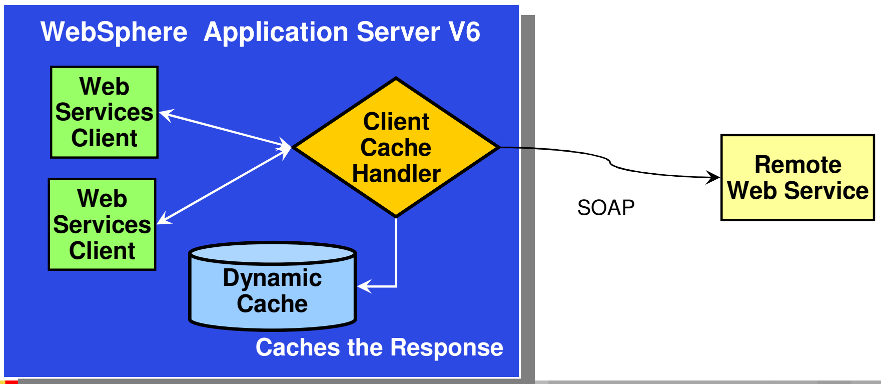
With WebSphere Application Server V6, you can use Remote Method Invocation over Internet Inter-ORB Protocol (RMI-IIOP) with JAX-RPC to support non-SOAP bindings. Using RMI-IIOP with JAX-RPC enables WebSphere Java clients to invoke enterprise beans using a WSDL file and the JAX-RPC programming model instead of using the standard J2EE programming model. When a Web service is implemented by an EJB, multi-protocol JAX-RPC permits the Web service invocation path to be optimized for WebSphere Java clients.

Using the RMI/IIOP protocol instead of a SOAP- based protocol yields better performance and enables you to get support for client transactions, which are not standard for Web services. Benefits include: XML processing is not required to send and receive messages; Java serialization is used instead. The client JAX-RPC call can participate in a user transaction, which is not the case when SOAP is used.

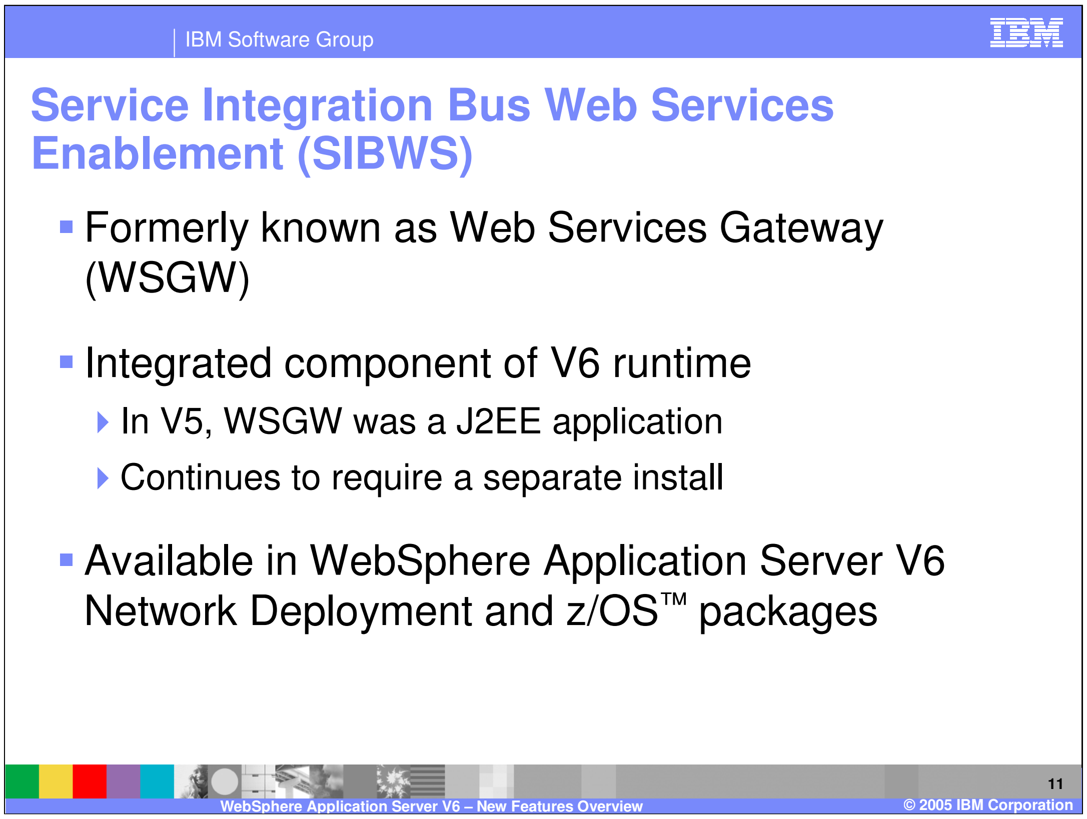


WebSphere Enhancements: Web Services Client Caching

- In addition to server-side caching, V6 supports caching for Web Services clients running within a V6 Application Server



Another enhancement available in Application Server V6 is the web services client cache. This functionality is part of the dynamic cache service that is used to increase the performance of Web services clients by caching responses from remote Web services. After a response is returned from a remote Web service, the response is saved in the client cache on the Application Server. Any identical requests that are made to the same remote Web service are then responded to from the cache for a specified period of time.



Service Integration Bus Web Services Enablement (SIBWS)

- Formerly known as Web Services Gateway (WSGW)
- Integrated component of V6 runtime
  - In V5, WSGW was a J2EE application
  - Continues to require a separate install
- Available in WebSphere Application Server V6 Network Deployment and z/OS™ packages

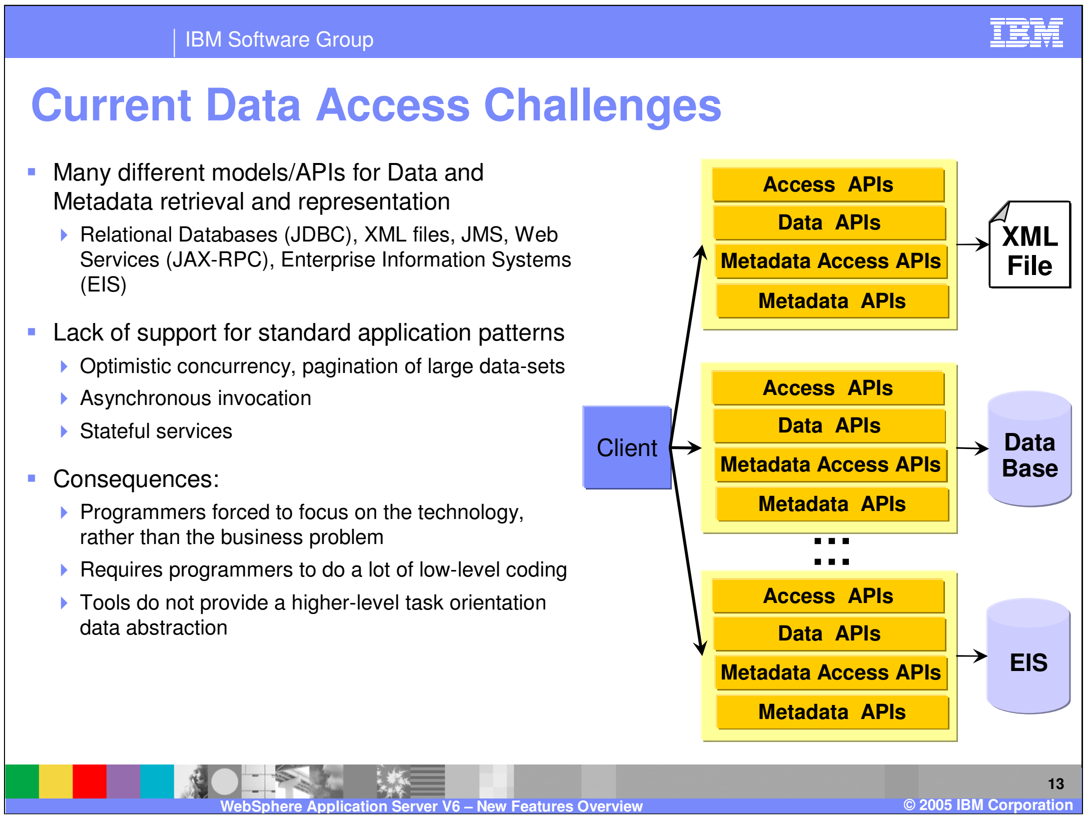
The function formerly known as Web Services Gateway is now an integrated component of the V6 runtime and is part of the Services Integration Technology support. This is available in the WebSphere Application Server V6 Network Deployment and z/OS packages.

Section

Application Programming Model

Service Data Object (SDO)

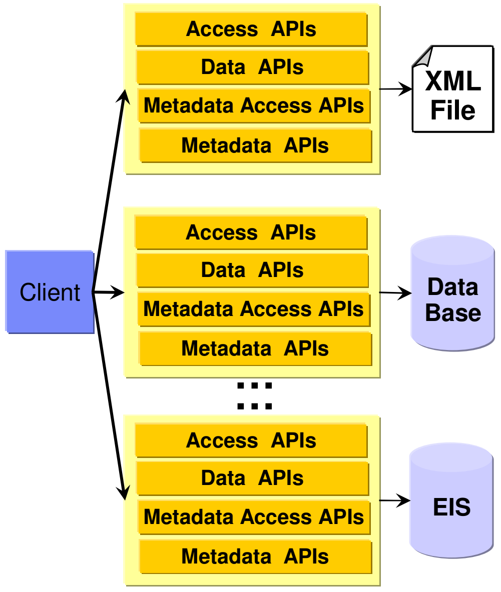
The next section will discuss the Service Data Object (SDO) API.



IBM Software Group

Current Data Access Challenges

- Many different models/APIs for Data and Metadata retrieval and representation
  - Relational Databases (JDBC), XML files, JMS, Web Services (JAX-RPC), Enterprise Information Systems (EIS)
- Lack of support for standard application patterns
  - Optimistic concurrency, pagination of large data-sets
  - Asynchronous invocation
  - Stateful services
- Consequences:
  - Programmers forced to focus on the technology, rather than the business problem
  - Requires programmers to do a lot of low-level coding
  - Tools do not provide a higher-level task orientation data abstraction

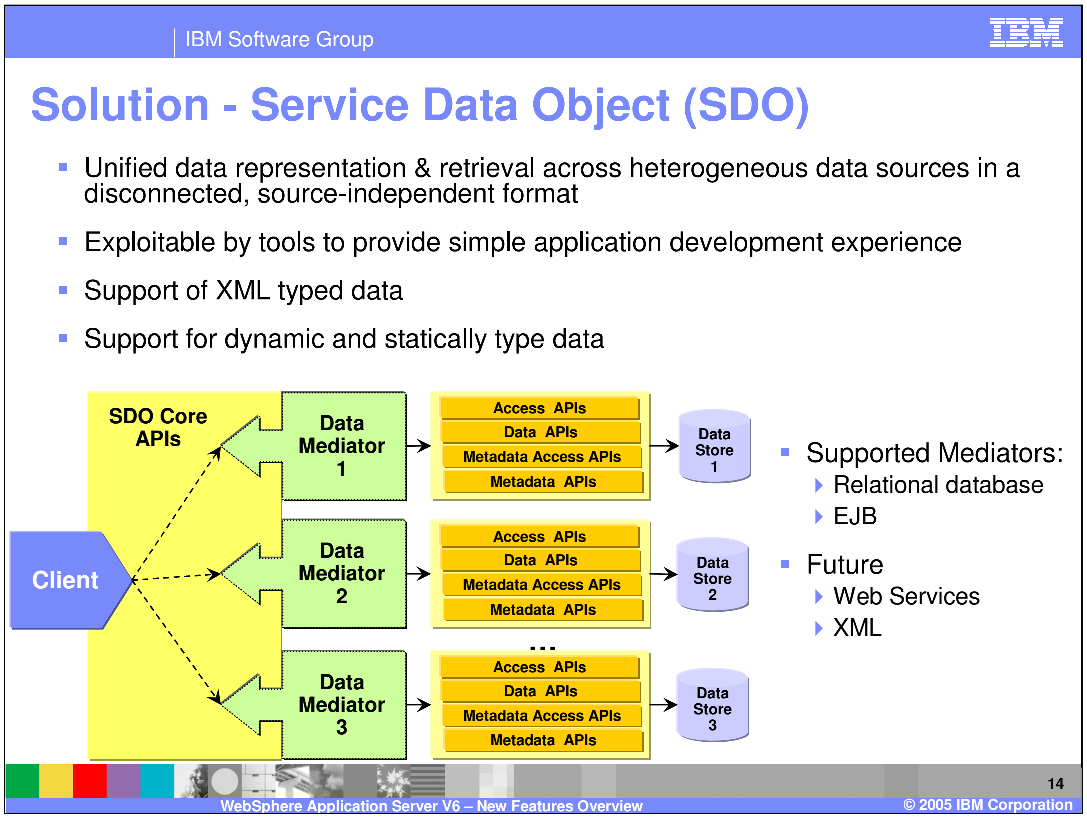


13

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

Several challenges exist today for programmers developing code to access data. There are different APIs and data models to learn and understand, and there is a lack of support for standard application patterns. This results in programmers being forced to focus on the technology rather than on the business problem and requires them to code more low-level programming. This also makes it difficult for development tools to integrate and provide a higher-level task orientation data abstraction.



IBM Software Group

Solution - Service Data Object (SDO)

- Unified data representation & retrieval across heterogeneous data sources in a disconnected, source-independent format
- Exploitable by tools to provide simple application development experience
- Support of XML typed data
- Support for dynamic and statically type data

SDO Core APIs
Client
Data Mediator 1
Data Mediator 2
Data Mediator 3
Access APIs
Data APIs
Metadata Access APIs
Metadata APIs
...
Data Store 1
Data Store 2
Data Store 3

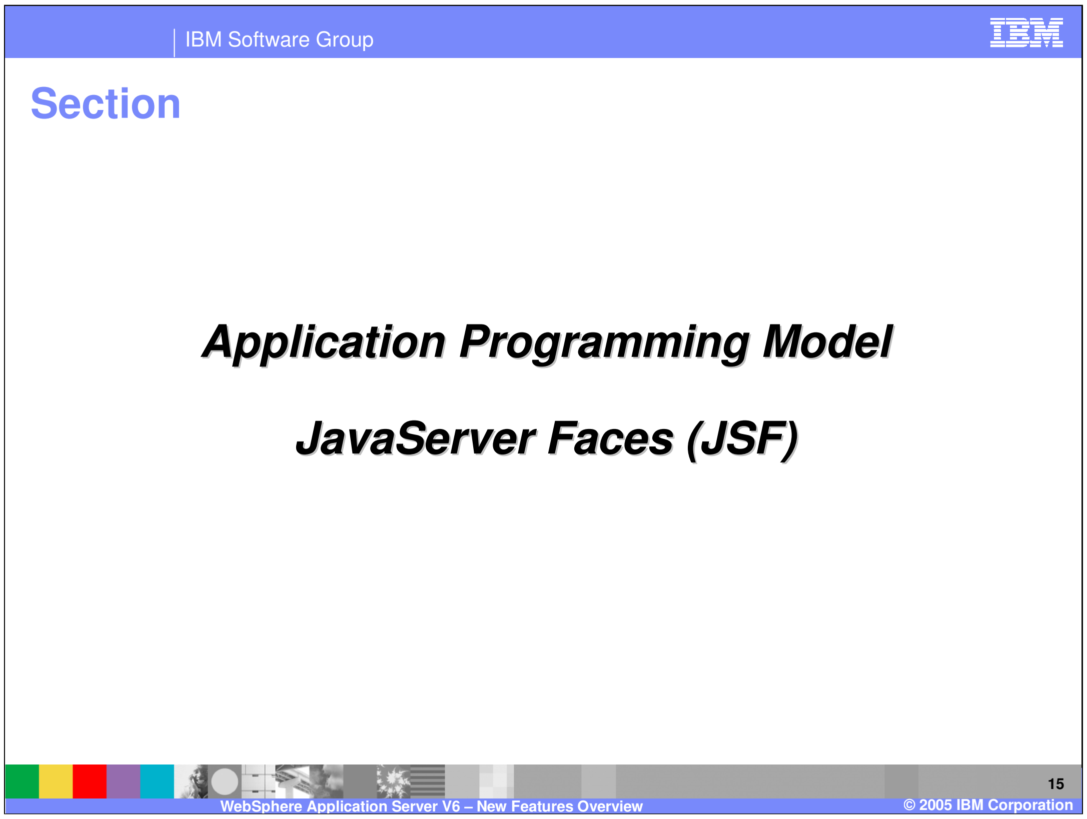
- Supported Mediators:
  - Relational database
  - EJB
- Future
  - Web Services
  - XML

14

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

A solution to the data access challenges is Service Data Object (SDO). The primary goal of the SDO architecture is to make it easier for application and tools developers to create, view, update, and delete data that is stored in a variety of backend data stores. The SDO architecture provides disconnected, uniform data access and representation across a wide variety of data sources as well as support for many common application patterns that are encountered in J2EE application development. This is accomplished by providing a core set of APIs that in turn access mediators for different data models. The client programmer can now use a simpler, common set of SDO APIs to read and write data.

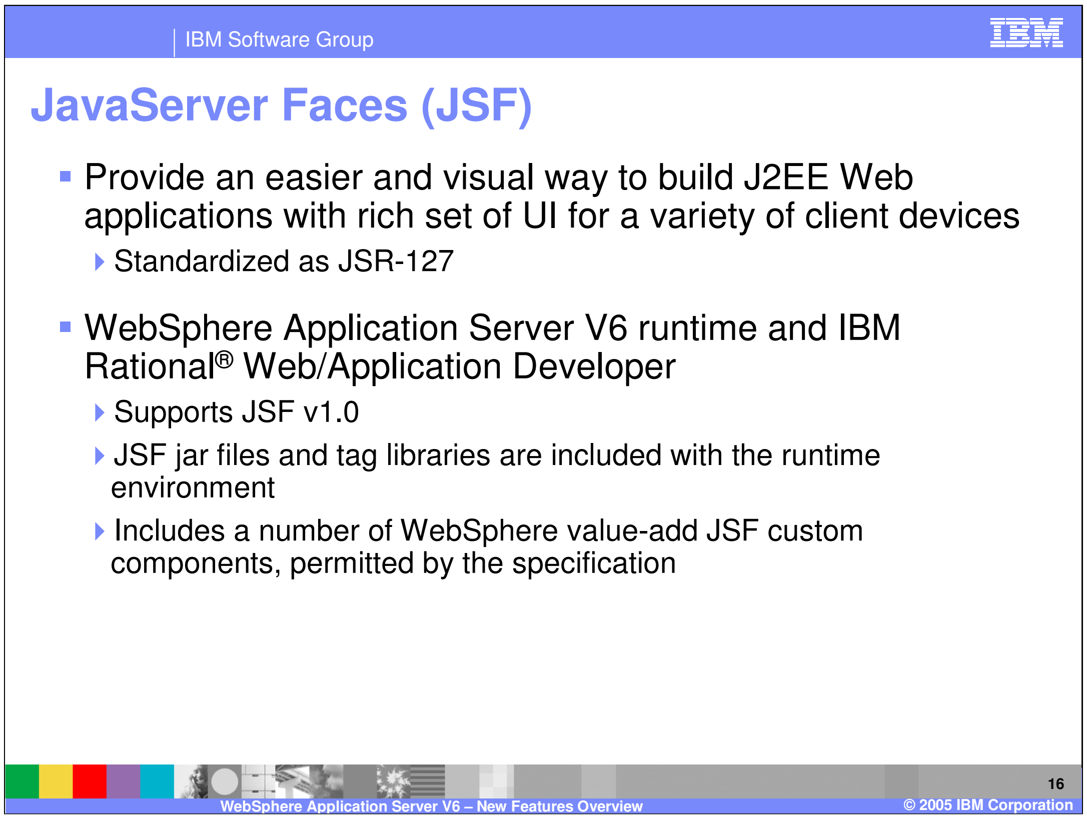


Section

Application Programming Model

JavaServer Faces (JSF)

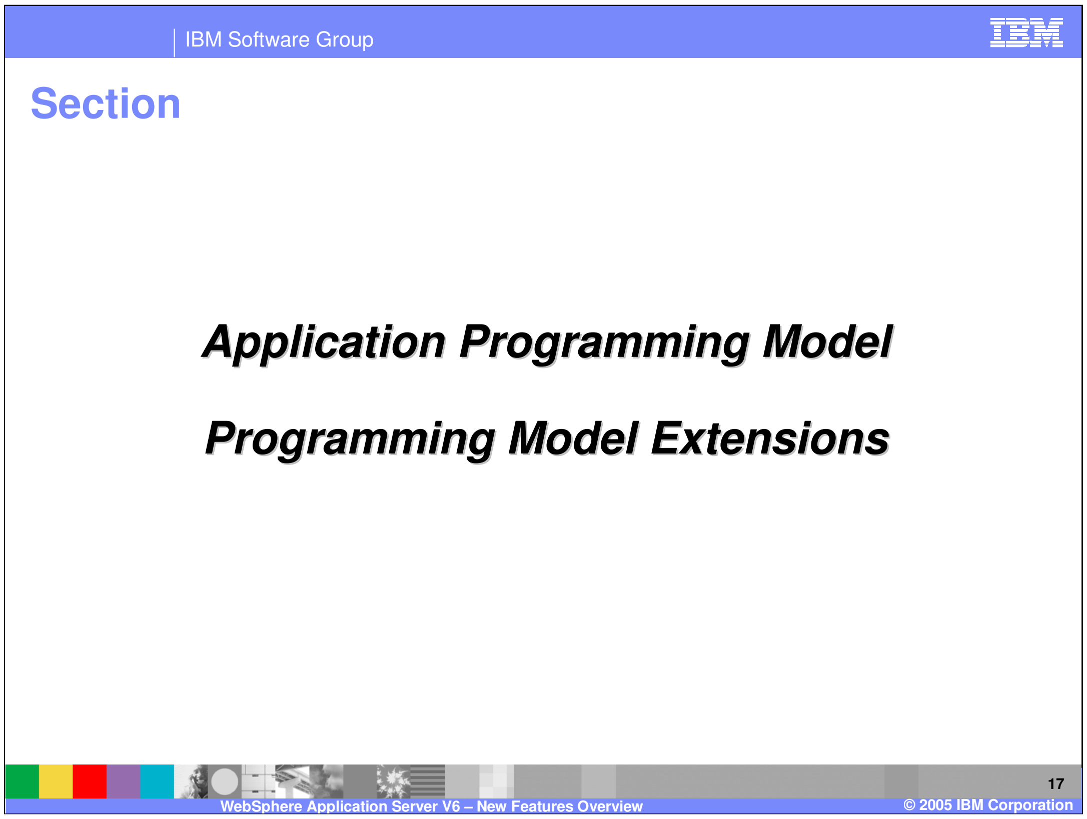
The next section will discuss the JavaServer Faces (JSF) API.



JavaServer Faces (JSF)

- Provide an easier and visual way to build J2EE Web applications with rich set of UI for a variety of client devices
  - Standardized as JSR-127
- WebSphere Application Server V6 runtime and IBM Rational® Web/Application Developer
  - Supports JSF v1.0
  - JSF jar files and tag libraries are included with the runtime environment
  - Includes a number of WebSphere value-add JSF custom components, permitted by the specification

JavaServer Faces allows for an easier way to build J2EE Web applications. Provided in the Rational Web and Rational Application Developer tools, is a rich set of visual components that you can select from a palette and include in your application.

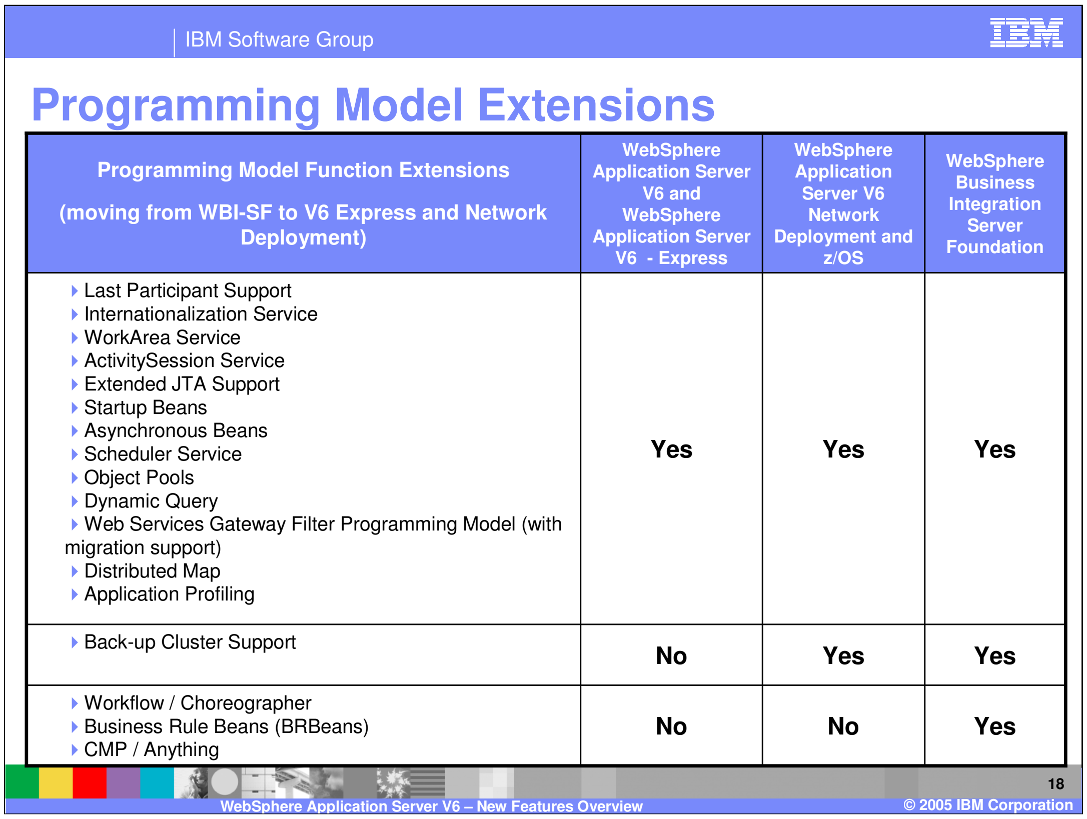


Section

Application Programming Model

Programming Model Extensions

The next section will discuss the Programming Model Extensions APIs.



IBM Software Group

Programming Model Extensions

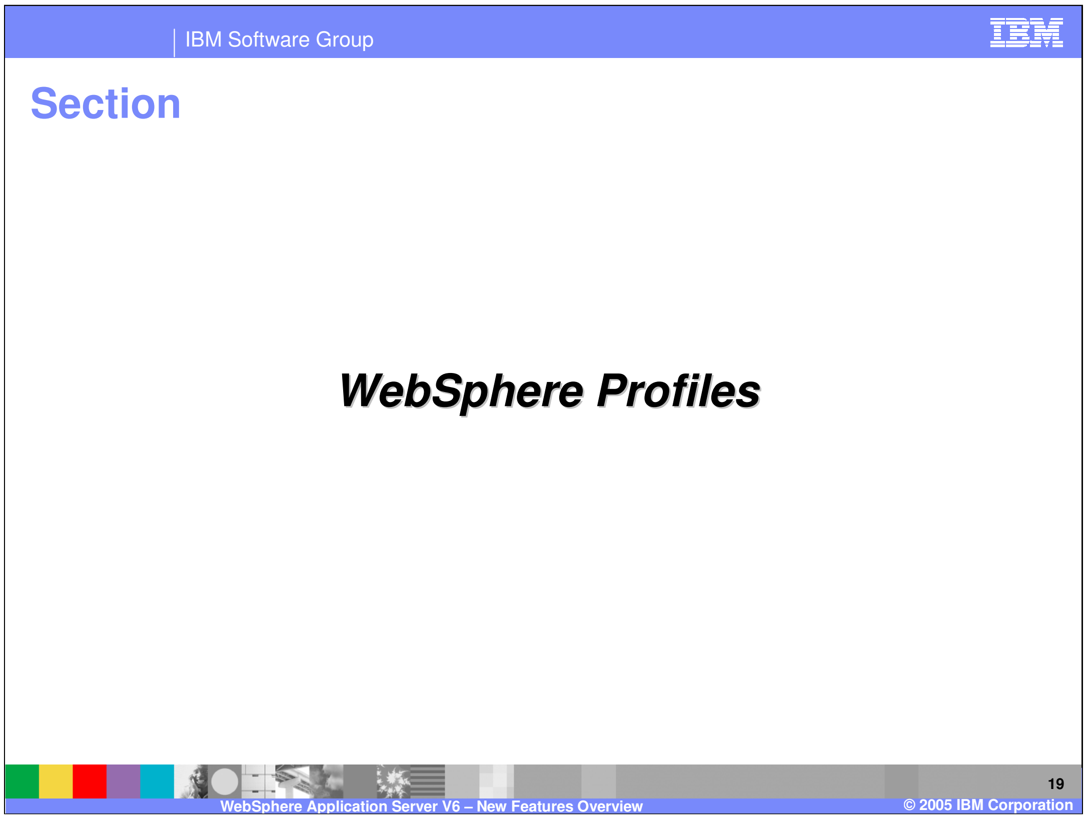
Programming Model Function Extensions (moving from WBI-SF to V6 Express and Network Deployment)	WebSphere Application Server V6 and WebSphere Application Server V6 - Express	WebSphere Application Server V6 Network Deployment and z/OS	WebSphere Business Integration Server Foundation
▸ Last Participant Support ▸ Internationalization Service ▸ WorkArea Service ▸ ActivitySession Service ▸ Extended JTA Support ▸ Startup Beans ▸ Asynchronous Beans ▸ Scheduler Service ▸ Object Pools ▸ Dynamic Query ▸ Web Services Gateway Filter Programming Model (with migration support) ▸ Distributed Map ▸ Application Profiling	**Yes**	**Yes**	**Yes**
▸ Back-up Cluster Support	**No**	**Yes**	**Yes**
▸ Workflow / Choreographer ▸ Business Rule Beans (BRBeans) ▸ CMP / Anything	**No**	**No**	**Yes**

18

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

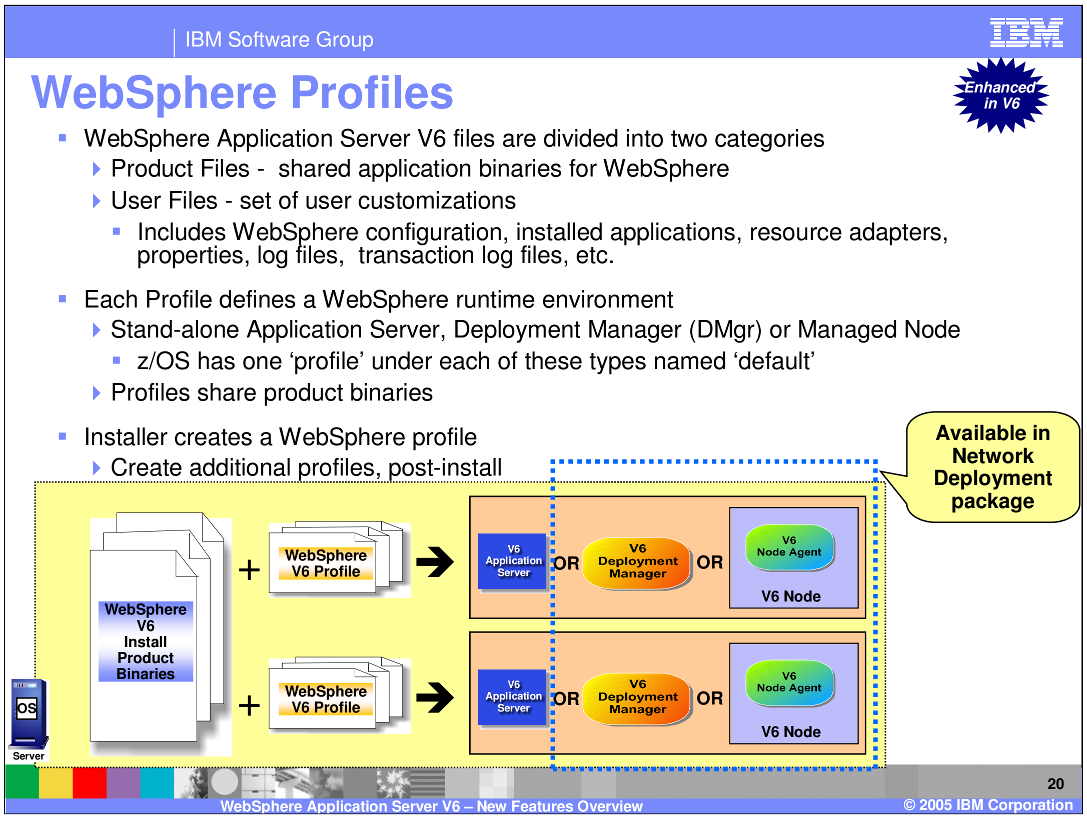
This chart provides the list of Programming Model Extensions that are now included in WebSphere Application Server V6. Previously, these were available only in the WebSphere Business Integration Server Foundation product.



Section

WebSphere Profiles

The next section will discuss WebSphere Profiles.



IBM Software Group

WebSphere Profiles

Enhanced in V6

- WebSphere Application Server V6 files are divided into two categories
  - Product Files - shared application binaries for WebSphere
  - User Files - set of user customizations
    - Includes WebSphere configuration, installed applications, resource adapters, properties, log files, transaction log files, etc.
- Each Profile defines a WebSphere runtime environment
  - Stand-alone Application Server, Deployment Manager (DMgr) or Managed Node
    - z/OS has one 'profile' under each of these types named 'default'
  - Profiles share product binaries
- Installer creates a WebSphere profile
  - Create additional profiles, post-install

Available in Network Deployment package

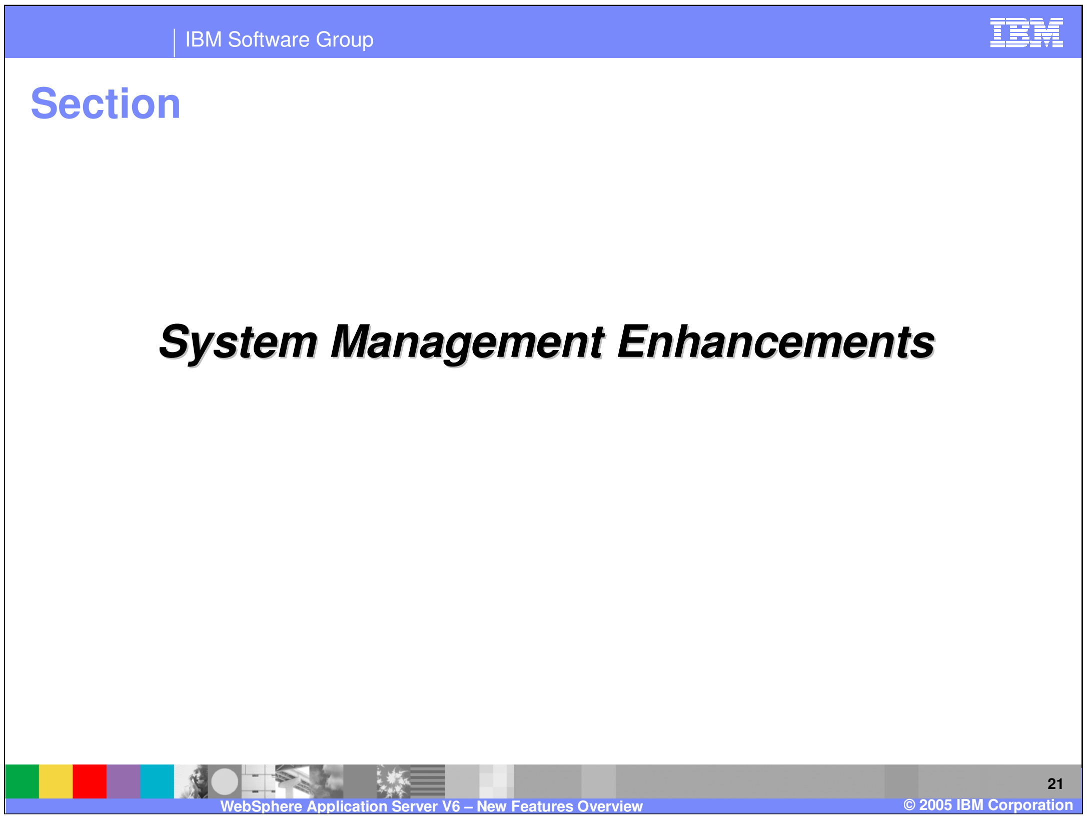
20

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

Consider the files that make up the WebSphere Application Server. There are two categories of files, product files and user files. The product files include the application binaries needed to run the Application Server. The user files contain information used by the Application Server. The user files are where variables are defined, resources are configured, and log files are stored. A profile is a collection of these files, creating a WebSphere Application Server runtime environment. When combined with the shared binaries, a profile becomes a complete WebSphere Application Server installation.

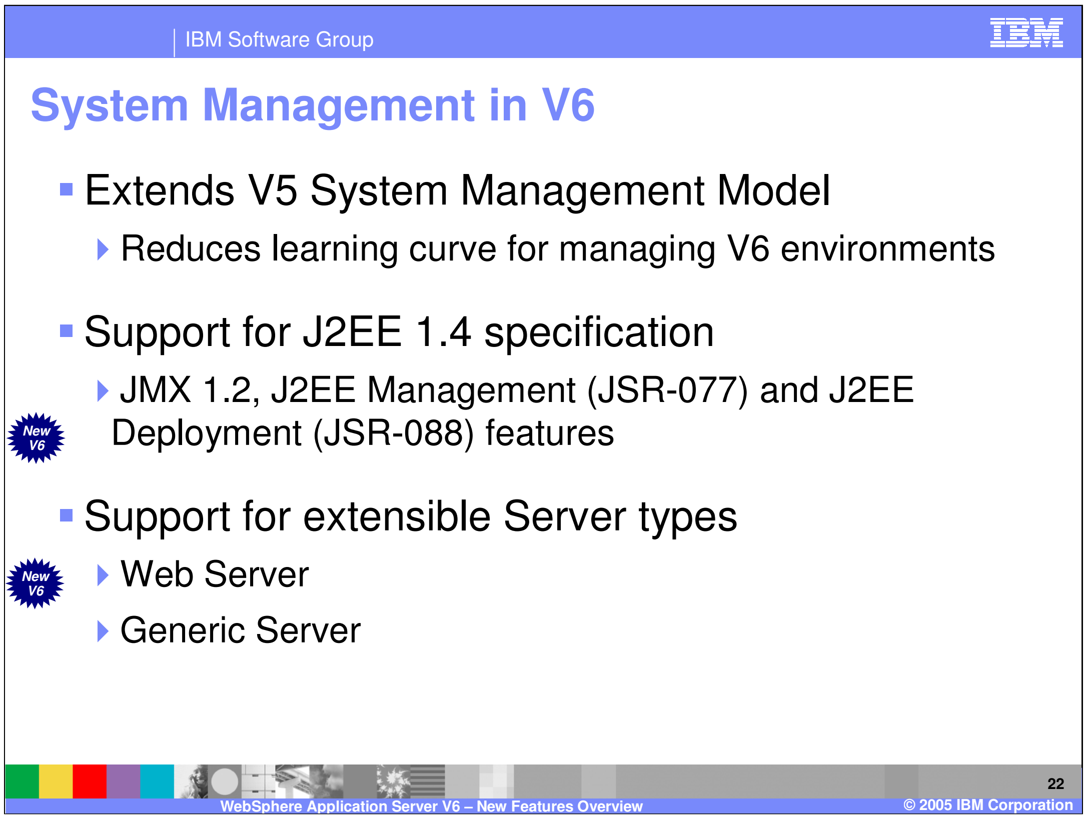
This sharing of product binary files and the separation of configuration files is an efficient use of disk space when creating multiple configurations. In addition, updates to the binary files are more easily applied as they reside in one location per physical machine, even when multiple profiles are configured.



Section

System Management Enhancements

The next section will discuss system management enhancements.



IBM Software Group

System Management in V6

- Extends V5 System Management Model
  - Reduces learning curve for managing V6 environments
- Support for J2EE 1.4 specification
  - JMX 1.2, J2EE Management (JSR-077) and J2EE Deployment (JSR-088) features
- Support for extensible Server types
  - Web Server
  - Generic Server

22

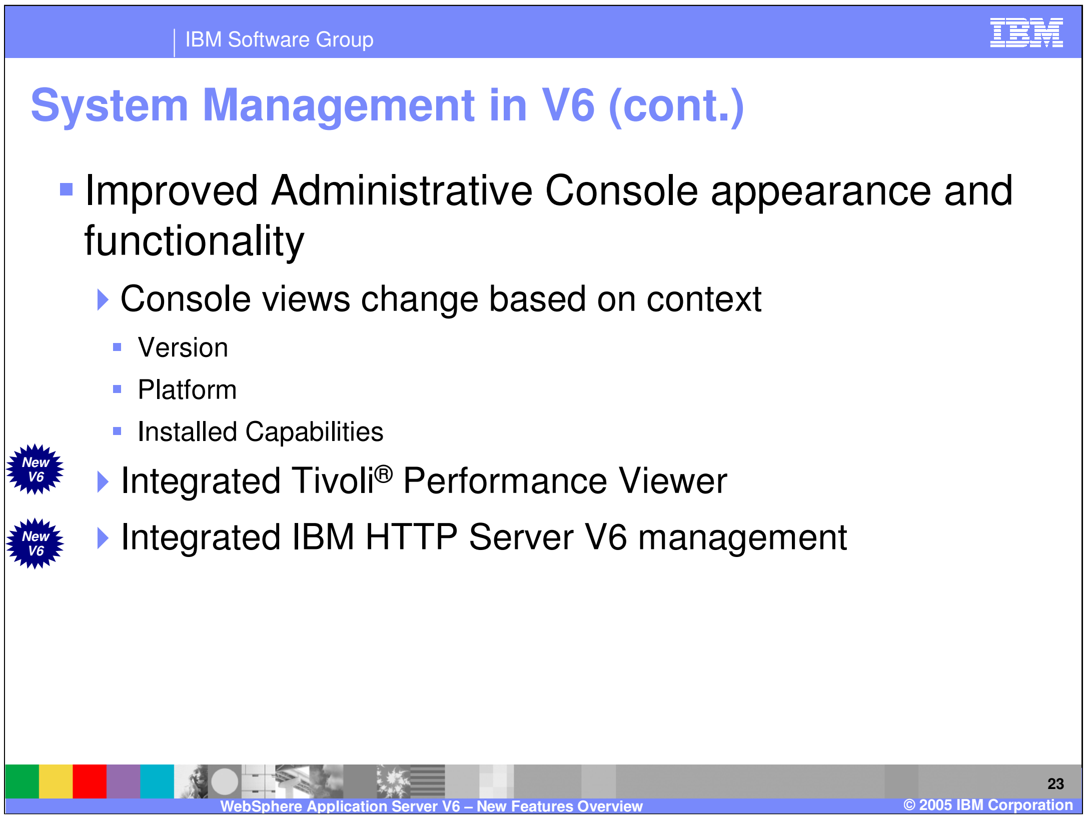
WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

System Management functionality in V6 builds upon and extends the V5 model. The configuration files continue to be stored in XML format. By extending the V5 model, the learning curve for managing V6 environments is reduced. Enhancements include J2EE 1.4 specification support as well as support for extensible server types.

The Web server feature allows you to associate a Web server with a previously defined managed or unmanaged node. After you define the Web server to a node, you can use the administrative console to perform functions for that Web server such as check the status of the Web server and start and stop the server.

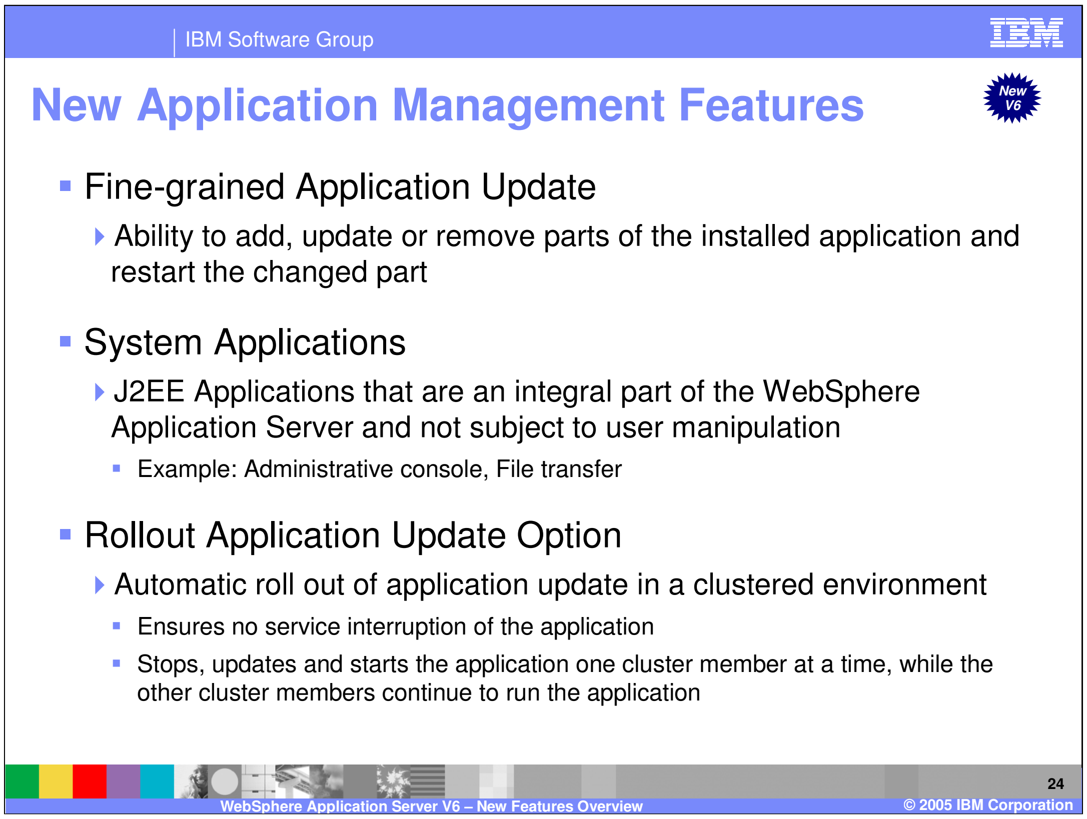
The Generic server feature allows you create a generic server as an application server instance within the WebSphere Application Server administration, and associate it with a non-WebSphere server or process. The generic server can be associated with any server or process necessary to support the application server environment, including a Java server, a C or C++ server or process, a CORBA server, or a Remote Method Invocation server. After you define a generic server, you can use the Application Server administrative console to start, stop, and monitor the associated non-WebSphere server or process when stopping or starting the applications that rely on them.



System Management in V6 (cont.)

- Improved Administrative Console appearance and functionality
  - Console views change based on context
    - Version
    - Platform
    - Installed Capabilities
  - New V6 Integrated Tivoli® Performance Viewer
  - New V6 Integrated IBM HTTP Server V6 management

The V6 administrative console appearance and functionality has been improved. The console views change based on the context being displayed. Tivoli Performance Viewer and IBM HTTP Server V6 management capabilities are now integrated into the console.



New Application Management Features

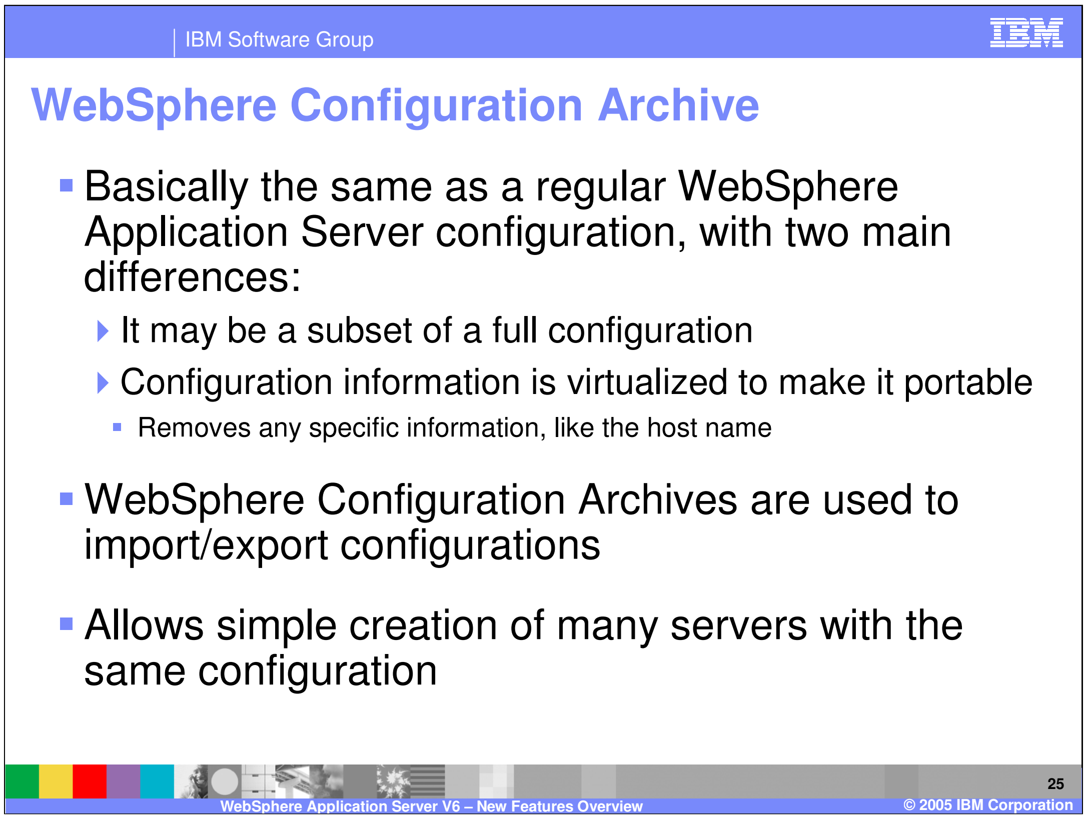
New V6

- Fine-grained Application Update
  - Ability to add, update or remove parts of the installed application and restart the changed part
- System Applications
  - J2EE Applications that are an integral part of the WebSphere Application Server and not subject to user manipulation
    - Example: Administrative console, File transfer
- Rollout Application Update Option
  - Automatic roll out of application update in a clustered environment
    - Ensures no service interruption of the application
    - Stops, updates and starts the application one cluster member at a time, while the other cluster members continue to run the application

Fine-grained application update is one of the new application management features of V6. When updating an application, the system needs to be made aware of only the portion of the code that actually changes. With application update, the application management logic determines the minimum action required in order to update the application. This action may involve stopping and restarting portions of the running application, or in some cases, the update can occur without stopping any portion of the running application.

System applications are another new application management feature. A system application is a J2EE enterprise application that is an integral part of the Application Server and not subject to user manipulation. System applications are not displayed in the list of installed applications on the administrative console, nor are they listed through wsadmin or Java APIs, to avoid an accidentally alteration of the system application. Examples of system applications are the administrative console and the file transfer applications.

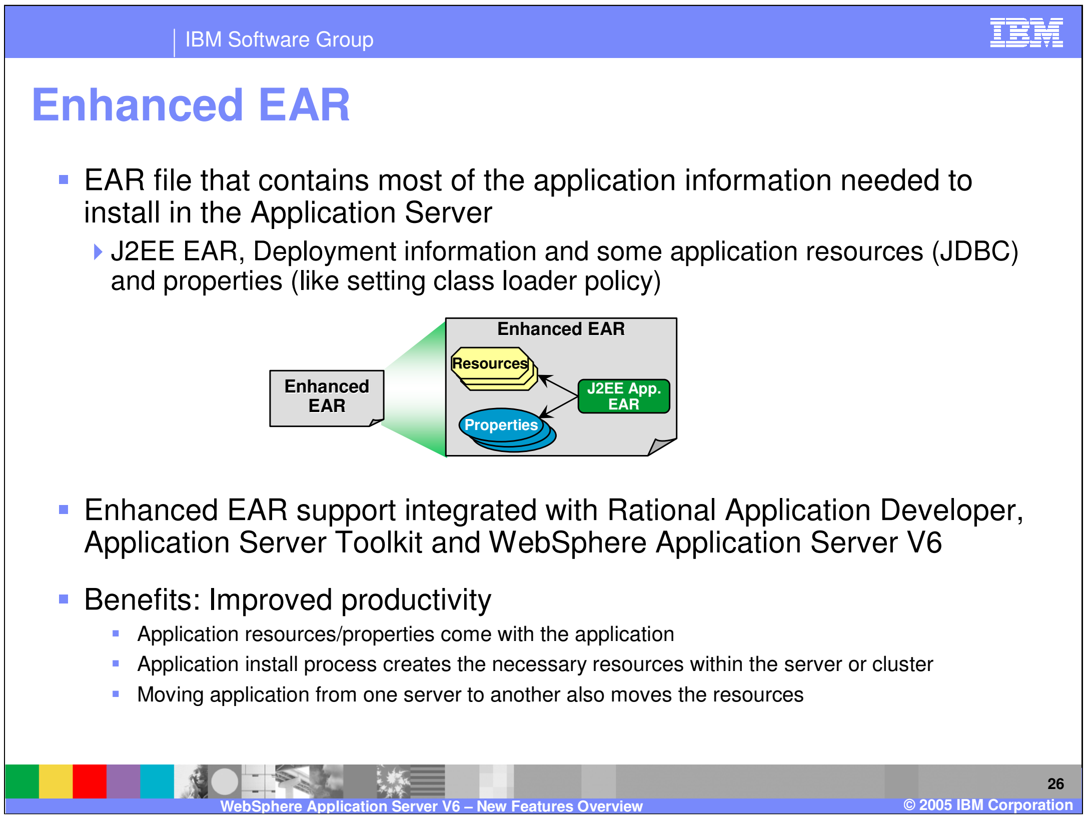
Rollout application update is another new feature that applies to updating applications. If the changed application or module is deployed on a cluster, click Rollout Update from the Enterprise Applications page of the administrative console to propagate the changed configuration on all members of the cluster on which the application or module is deployed. Rollout Update sequentially updates the configuration on the nodes that contain cluster members.



WebSphere Configuration Archive

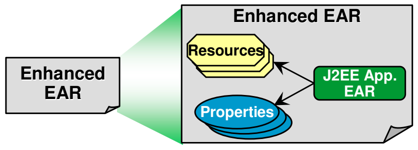
- Basically the same as a regular WebSphere Application Server configuration, with two main differences:
  - It may be a subset of a full configuration
  - Configuration information is virtualized to make it portable
    - Removes any specific information, like the host name
- WebSphere Configuration Archives are used to import/export configurations
- Allows simple creation of many servers with the same configuration

WebSphere configuration archive provides you with a way to import or export a full WebSphere Application Server configuration or a subset of the configuration, in the case of a single server. This is useful for propagating the configuration from one profile to another allowing for the simple creation of many servers with the same configuration. Information unique to a specific server is "virtualized" upon export, making the archive file portable. This information is then replaced upon the importing of the configuration to the new server. The wsadmin scripting tool is used to invoke this function.



Enhanced EAR

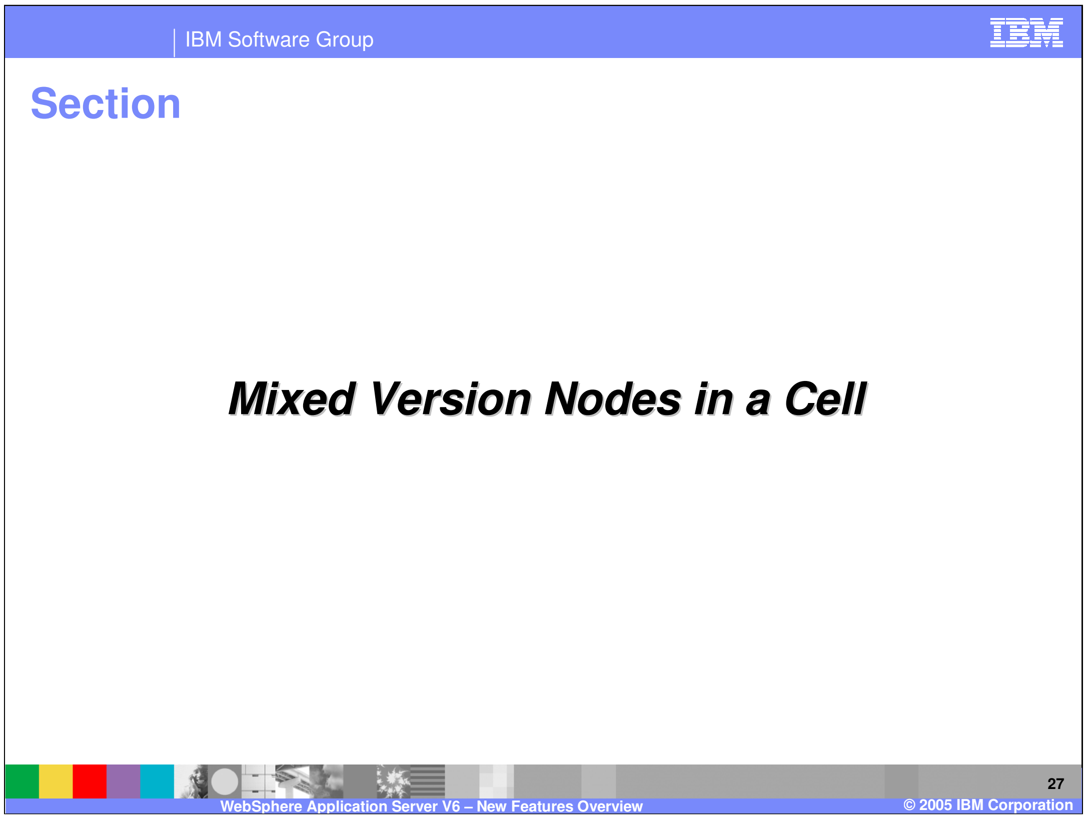
- EAR file that contains most of the application information needed to install in the Application Server
  - J2EE EAR, Deployment information and some application resources (JDBC) and properties (like setting class loader policy)



- Enhanced EAR support integrated with Rational Application Developer, Application Server Toolkit and WebSphere Application Server V6
- Benefits: Improved productivity
  - Application resources/properties come with the application
  - Application install process creates the necessary resources within the server or cluster
  - Moving application from one server to another also moves the resources

Using the enhanced EAR editor from Application Developer or Application Server Toolkit, you can define resources and properties for the application, embed those definitions within the application resulting in an Enhanced EAR, and then export that application to be installed by your system administrator. The system administrator no longer needs to define this deployment information, as it is already included. This improves productivity in application installation and in moving applications from one server to another.

Not all resources are capable of being defined in this manner at this point in time. For example, JMS and JavaMail resources are not currently included in this support.


IBM Software Group

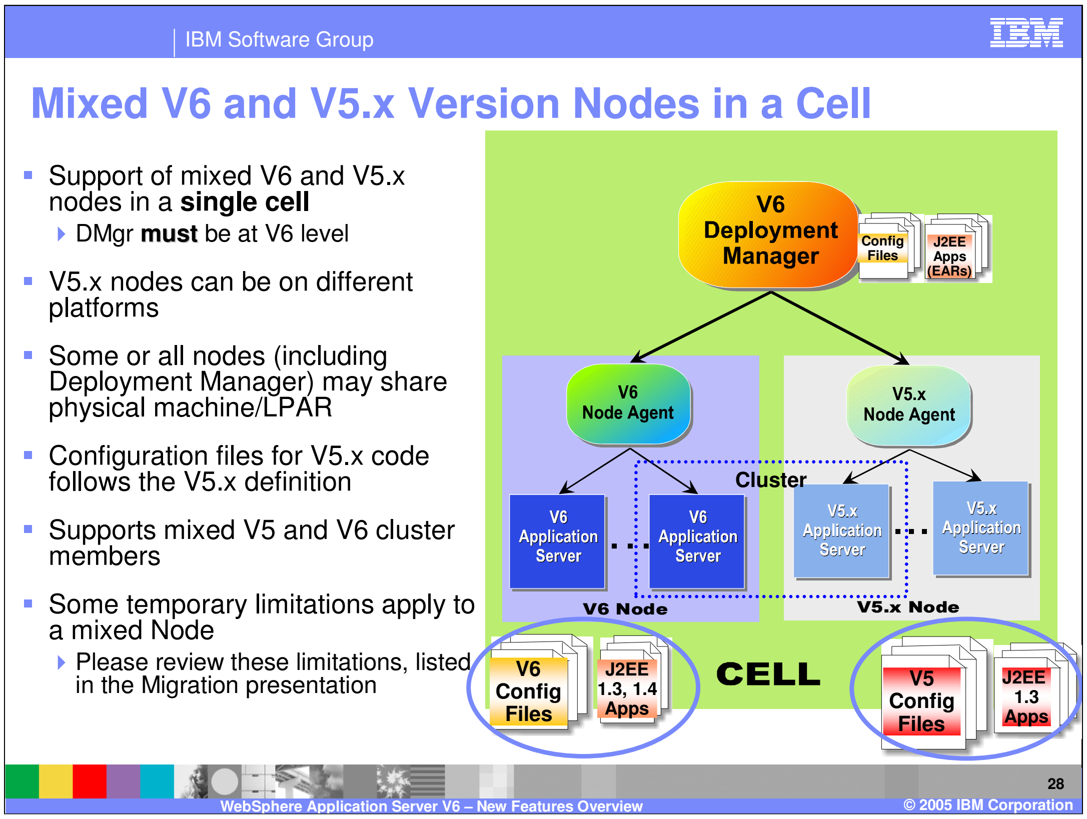
Section

Mixed Version Nodes in a Cell

27

WebSphere Application Server V6 – New Features Overview

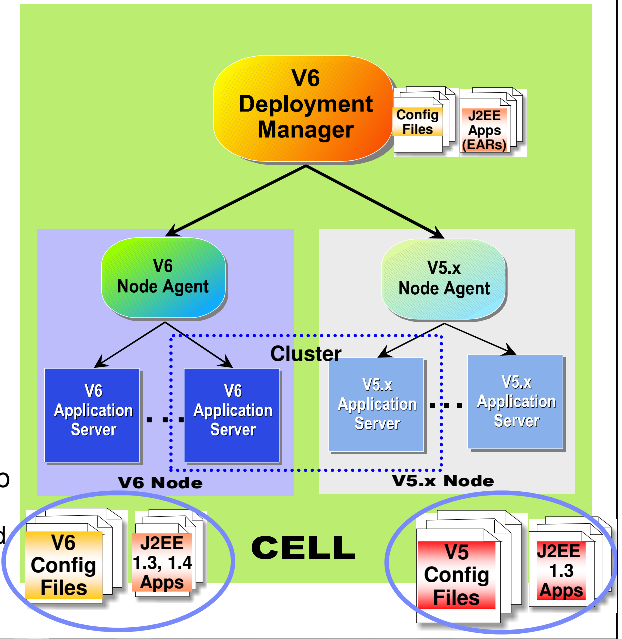
The next section will discuss Mixed Version Nodes in a Cell.



IBM Software Group

Mixed V6 and V5.x Version Nodes in a Cell

- Support of mixed V6 and V5.x nodes in a **single cell**
  - DMgr **must** be at V6 level
- V5.x nodes can be on different platforms
- Some or all nodes (including Deployment Manager) may share physical machine/LPAR
- Configuration files for V5.x code follows the V5.x definition
- Supports mixed V5 and V6 cluster members
- Some temporary limitations apply to a mixed Node
  - Please review these limitations, listed in the Migration presentation



28

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

There is support for mixed V6 and V5.x nodes in a single cell, if the Deployment Manager is at the V6 level. You may take advantage of this support, for example, after migrating a V5 Deployment Manager to a V6 Deployment Manager. The V6 Deployment Manager runs in compatibility mode by default, where it can manage both V5 nodes and V6 nodes.

IBM Software Group

Section

New V6

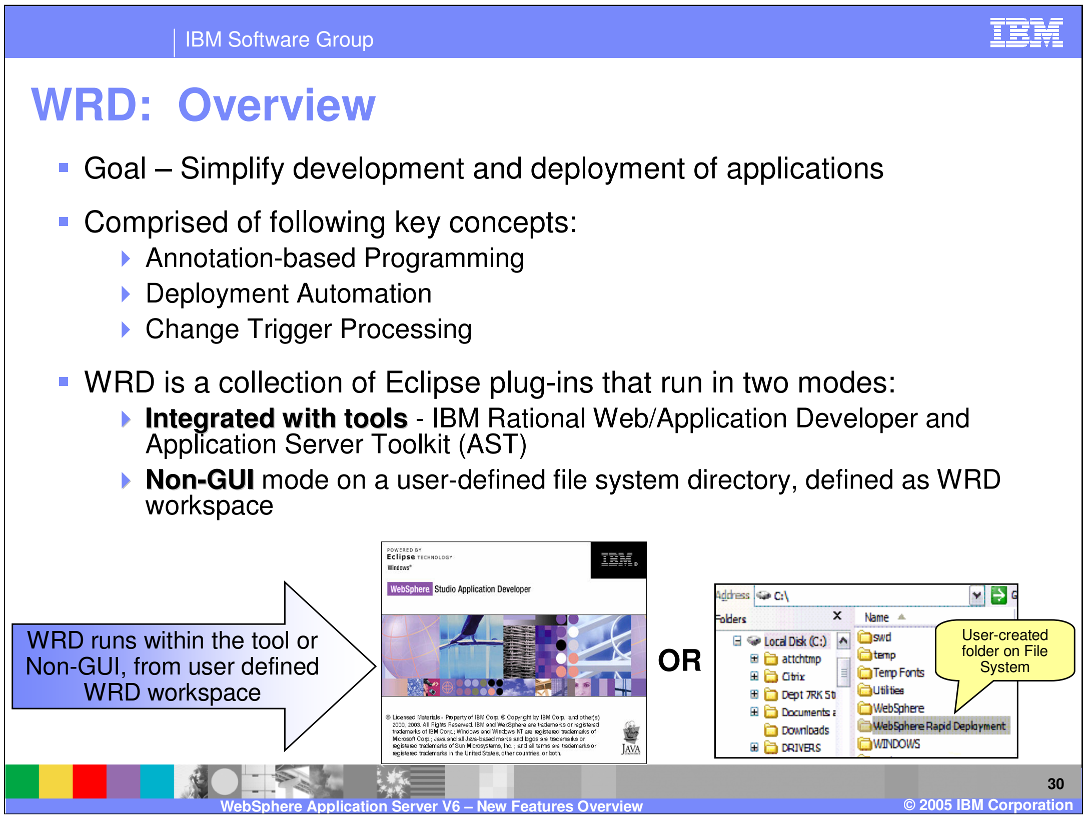
WebSphere Rapid Deployment (WRD)

29

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

The next section will discuss WebSphere Rapid Deployment.



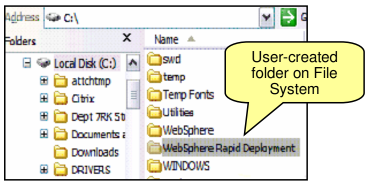
WRD: Overview

- Goal – Simplify development and deployment of applications
- Comprised of following key concepts:
  - Annotation-based Programming
  - Deployment Automation
  - Change Trigger Processing
- WRD is a collection of Eclipse plug-ins that run in two modes:
  - **Integrated with tools** - IBM Rational Web/Application Developer and Application Server Toolkit (AST)
  - **Non-GUI** mode on a user-defined file system directory, defined as WRD workspace

WRD runs within the tool or Non-GUI, from user defined WRD workspace

POWERED BY Eclipse TECHNOLOGY
Windows
IBM
WebSphere Studio Application Developer

OR



WebSphere Rapid Deployment (WRD) simplifies the development and deployment of applications. It's capabilities include annotation-based programming, deployment automation, and change-triggered processing. To use WRD functionality, no changes are required on the Application Server. It uses existing Application Server administration function to deploy and control applications.

Annotation-based programming allows the developer to add metadata tags into application source code. WRD uses the metadata to generate additional J2EE artifacts needed to run the application on the Application Server.

Deployment Automation allows for automatic deployment of applications from a working directory to a test Application Server environment.

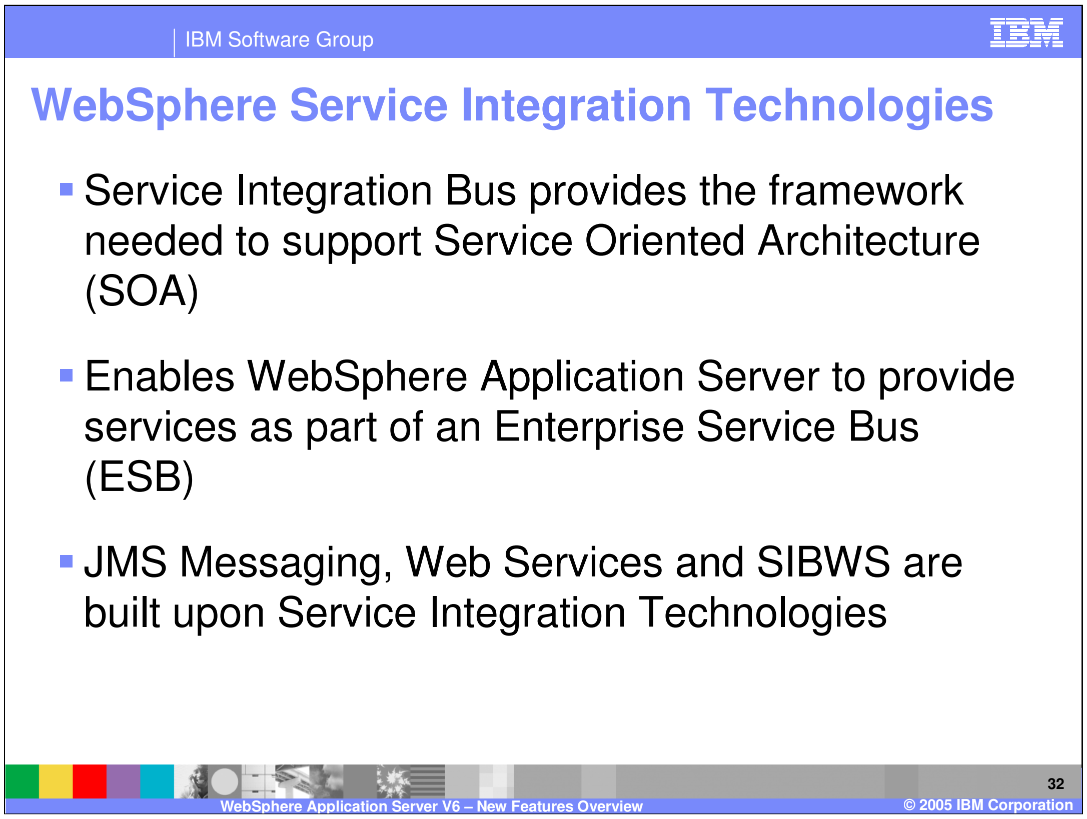
Change Trigger processing provides automatic monitoring of changes in the WRD user workspace. Changes trigger the automatic generation of code and deployment of the application to the Application Server.

Section

New V6

WebSphere Service Integration Technologies

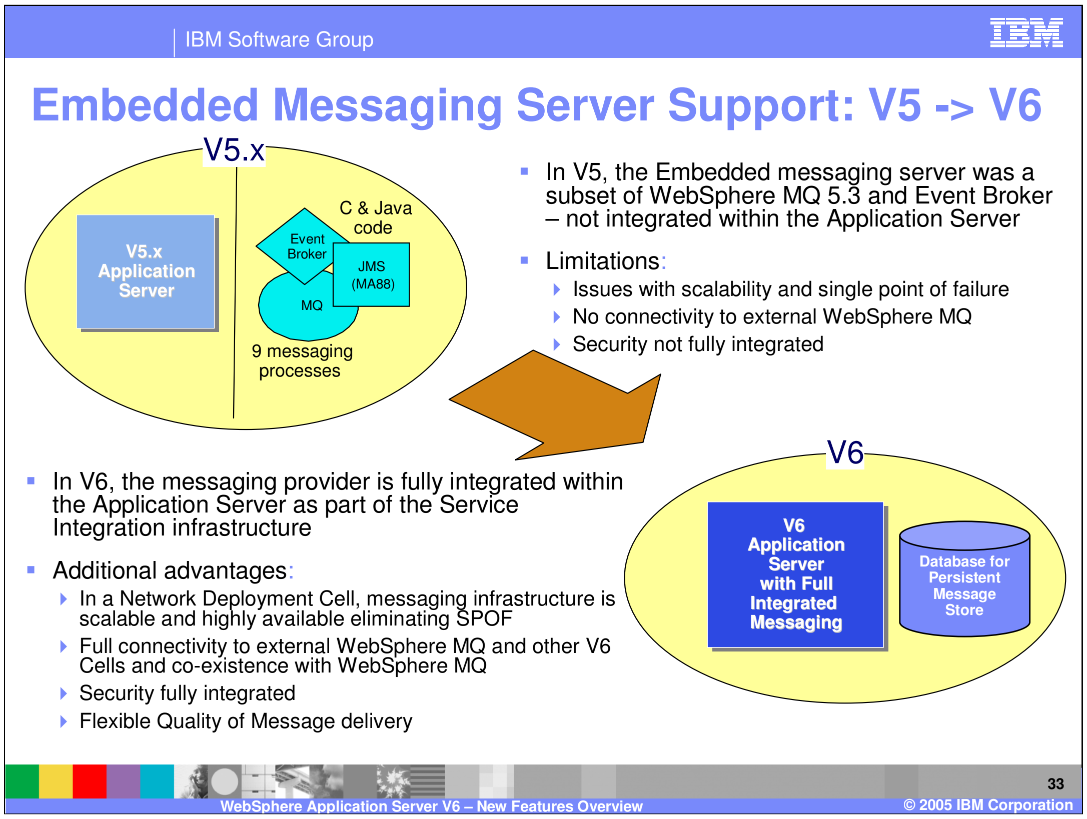
The next section will discuss WebSphere Service Integration Technologies.



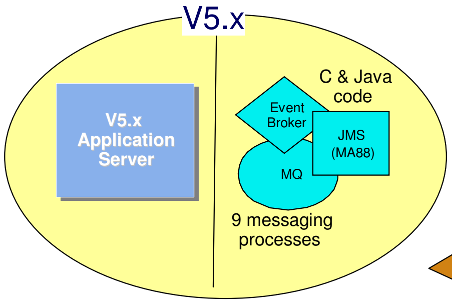
WebSphere Service Integration Technologies

- Service Integration Bus provides the framework needed to support Service Oriented Architecture (SOA)
- Enables WebSphere Application Server to provide services as part of an Enterprise Service Bus (ESB)
- JMS Messaging, Web Services and SIBWS are built upon Service Integration Technologies

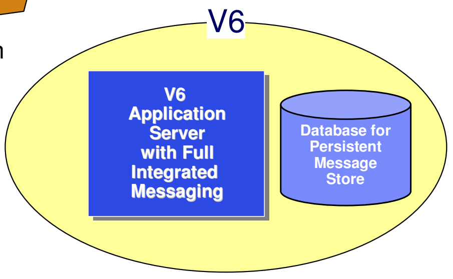
The service integration functionality within WebSphere Application Server provides a highly-flexible messaging fabric that supports a service-oriented architecture with a wide spectrum of quality of service options, supported protocols, and messaging patterns. It supports both message-oriented and service-oriented applications.



Embedded Messaging Server Support: V5 -> V6

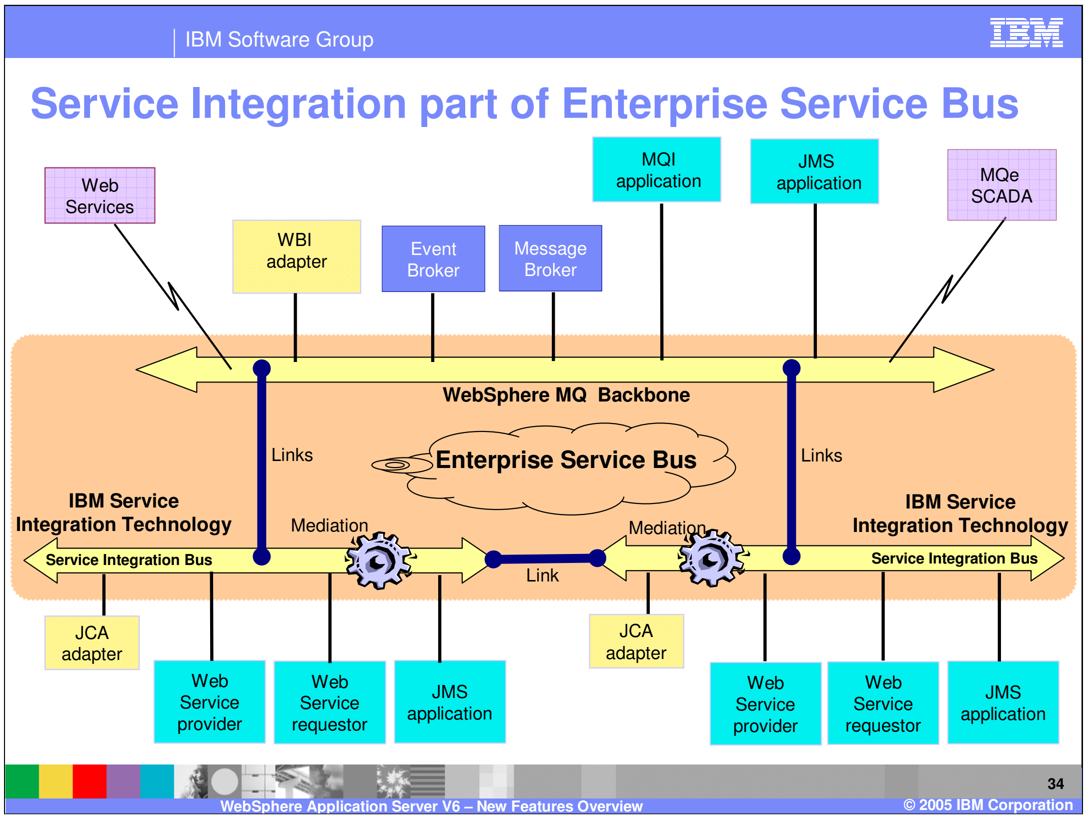


- In V5, the Embedded messaging server was a subset of WebSphere MQ 5.3 and Event Broker – not integrated within the Application Server
- Limitations:
  - Issues with scalability and single point of failure
  - No connectivity to external WebSphere MQ
  - Security not fully integrated


V6
V6 Application Server with Full Integrated Messaging
Database for Persistent Message Store

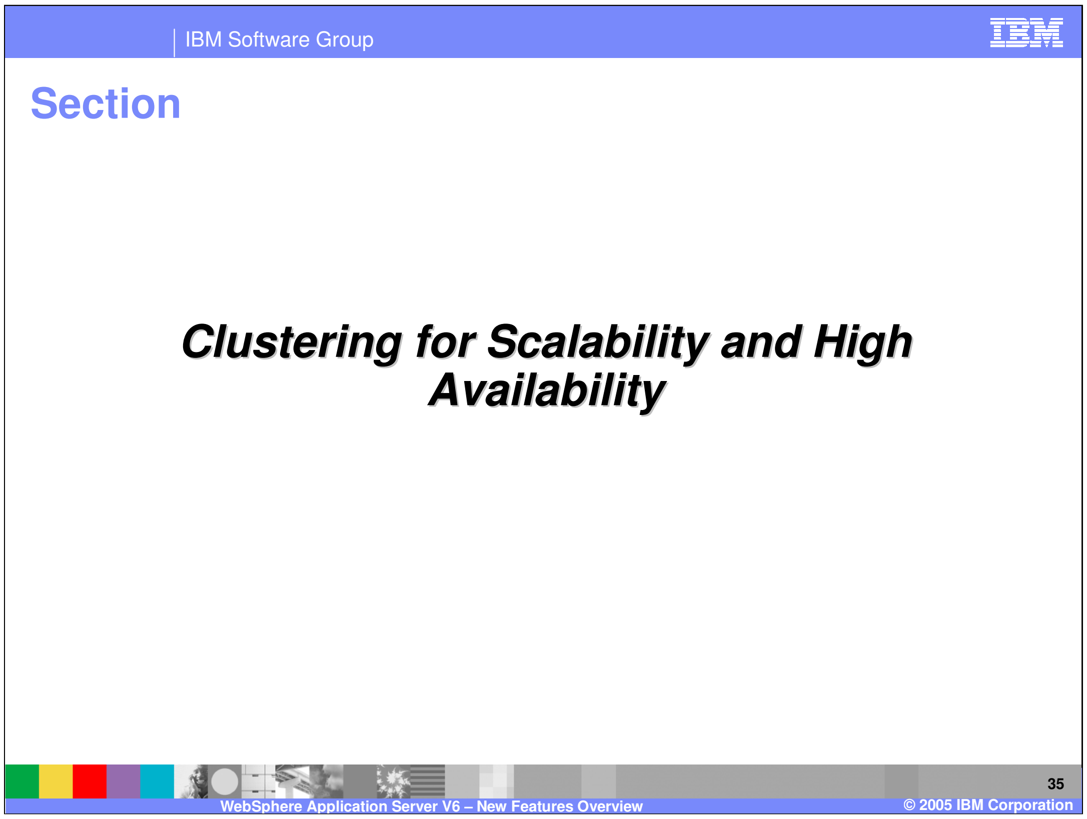
- In V6, the messaging provider is fully integrated within the Application Server as part of the Service Integration infrastructure
- Additional advantages:
  - In a Network Deployment Cell, messaging infrastructure is scalable and highly available eliminating SPOF
  - Full connectivity to external WebSphere MQ and other V6 Cells and co-existence with WebSphere MQ
  - Security fully integrated
  - Flexible Quality of Message delivery

WebSphere Application Server V6 provides a pure Java JMS 1.1 provider that is installed as part of the Service Integration Technology infrastructure, during the server installation. This support is fully integrated within the Application Server JVM. V6 uses the WebSphere Application Server supported databases with JDBC for the support of persistent messages. As this functionality is now fully integrated within the Application Server, it is able to take advantage of the same underlying infrastructure and capabilities provided by the Application Server. This includes functionality provided by systems management, logging, security, performance monitoring, and RAS. Each server can have its own interconnected messaging engine. This messaging support is capable of interoperating with WebSphere MQSeries®.



Service Integration part of Enterprise Service Bus

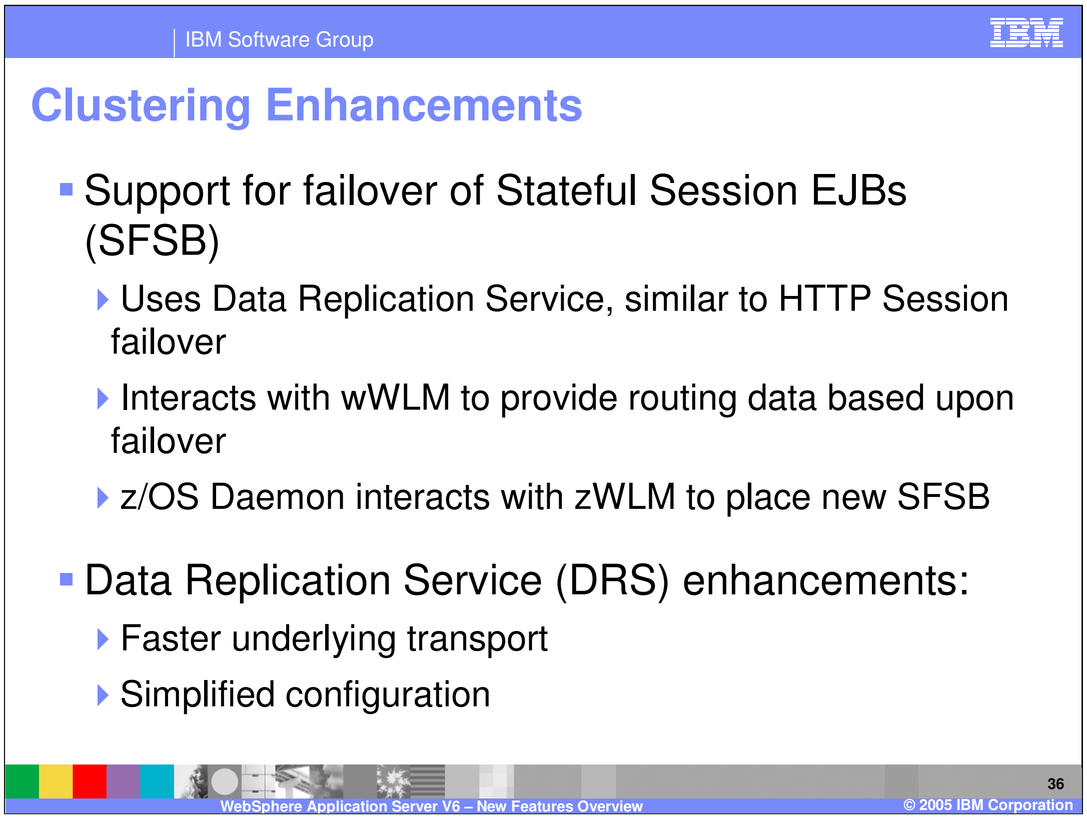
The existing WebSphere messaging products provide a comprehensive range of functions which enable you to build an Enterprise Service Bus (ESB) today. This is augmented by the new Service Integration Technology functions which provide greater support for J2EE and Web Services standards. Service Integration provides a more flexible and integrated messaging solution for the Application Server with connectivity onto the ESB.



Section

Clustering for Scalability and High Availability

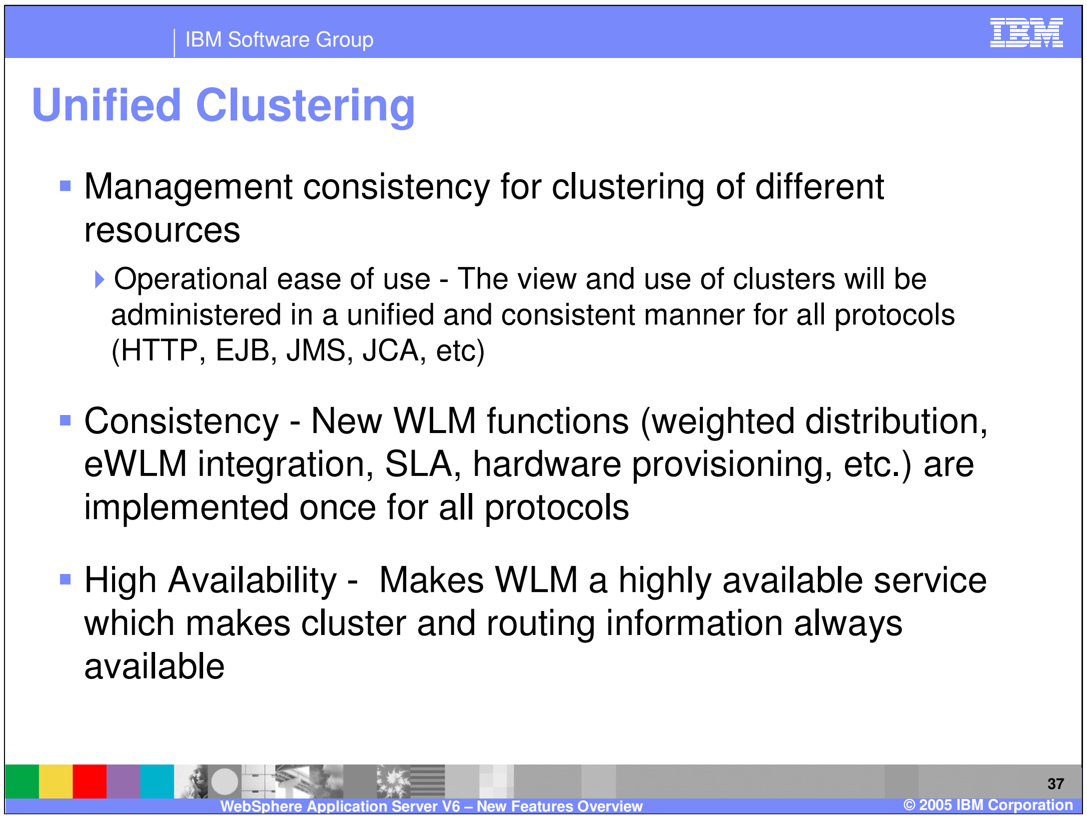
The next section will discuss Clustering for Scalability and High Availability.



Clustering Enhancements

- Support for failover of Stateful Session EJBs (SFSB)
  - Uses Data Replication Service, similar to HTTP Session failover
  - Interacts with wWLM to provide routing data based upon failover
  - z/OS Daemon interacts with zWLM to place new SFSB
- Data Replication Service (DRS) enhancements:
  - Faster underlying transport
  - Simplified configuration

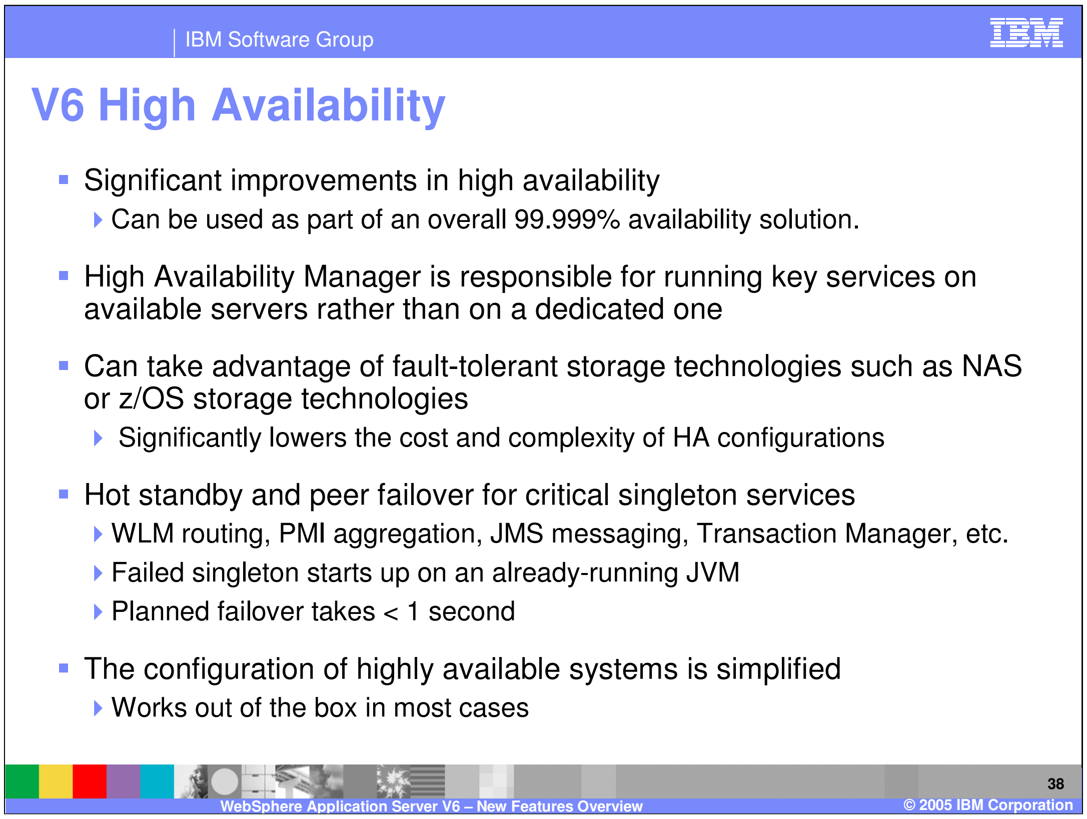
Support for failover of stateful Session EJBs is now available in V6. This support relies upon the Data Replication Service (DRS) and WebSphere Workload Management (wWLM) services. Enhancements have also been made to DRS in the areas of providing a faster underlying transport and simplified configuration.



Unified Clustering

- Management consistency for clustering of different resources
  - Operational ease of use - The view and use of clusters will be administered in a unified and consistent manner for all protocols (HTTP, EJB, JMS, JCA, etc)
- Consistency - New WLM functions (weighted distribution, eWLM integration, SLA, hardware provisioning, etc.) are implemented once for all protocols
- High Availability - Makes WLM a highly available service which makes cluster and routing information always available

The unified clustering framework standardizes how the cluster data is collected, propagated, and routed using a standard consistent architecture. As new technologies are introduced into WebSphere Application Server, they will also be able to take advantage of the framework. WLM is now a highly available service, which allows clustering and routing information to always be available.


IBM Software Group

V6 High Availability

- Significant improvements in high availability
  - Can be used as part of an overall 99.999% availability solution.
- High Availability Manager is responsible for running key services on available servers rather than on a dedicated one
- Can take advantage of fault-tolerant storage technologies such as NAS or z/OS storage technologies
  - Significantly lowers the cost and complexity of HA configurations
- Hot standby and peer failover for critical singleton services
  - WLM routing, PMI aggregation, JMS messaging, Transaction Manager, etc.
  - Failed singleton starts up on an already-running JVM
  - Planned failover takes < 1 second
- The configuration of highly available systems is simplified
  - Works out of the box in most cases

38

WebSphere Application Server V6 – New Features Overview

© 2005 IBM Corporation

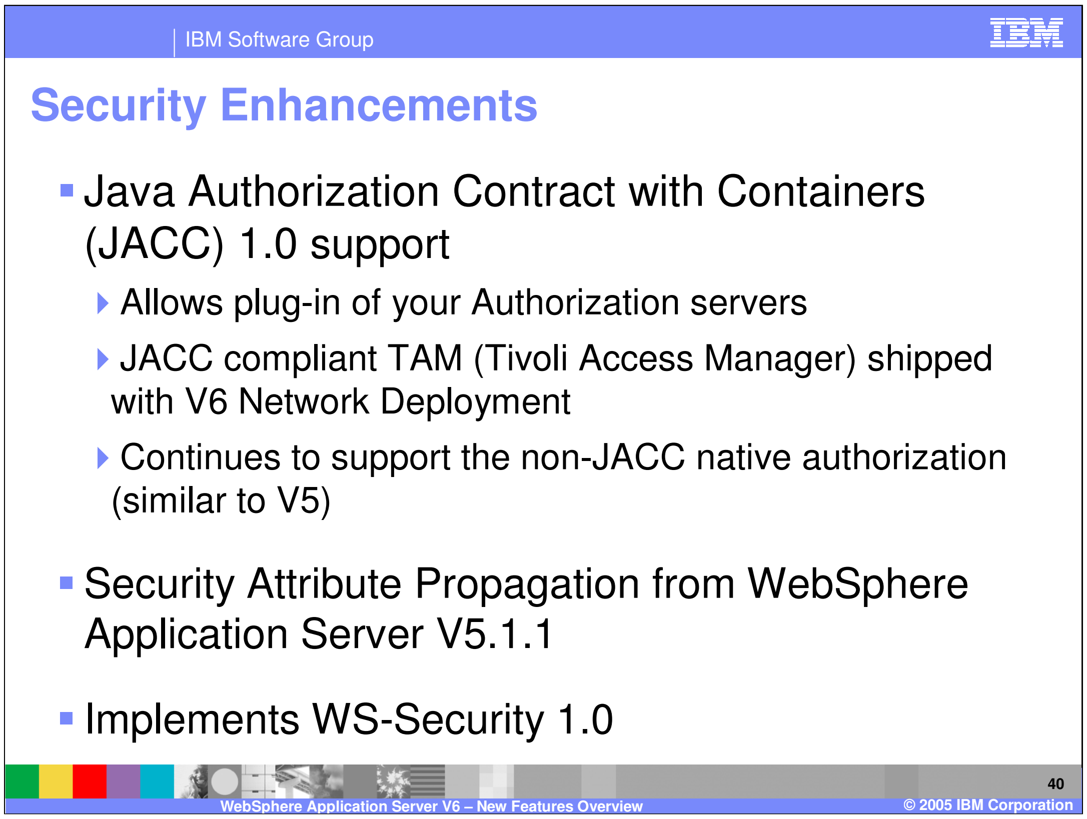
Significant improvements in V6 have been made in the area of high availability. WebSphere Application Server can now be used as part of an overall 99.999% availability solution as a result of the functionality provided by the new High Availability Manager (HA Manager). The HA Manager runs key services on any Application Server that is available, rather than using only a dedicated server, such as the Deployment Manager. The HA Manager keeps track of the status of all of your servers and the services they are running, ensuring that all services remain continuously available. When a failure is detected, the failed service can be started in another already-running JVM, potentially on another physical machine, in very little time. Planned failover takes less than a second. On z/OS, Peer Restart and Recovery (PRR) is still available in V6 but it is no longer being improved. Using the HA Manager instead provides for a faster recovery.

The configuration of highly available systems is greatly simplified. In most cases, this capability will work out of the box, with no configuration required.

Section

Security Enhancements

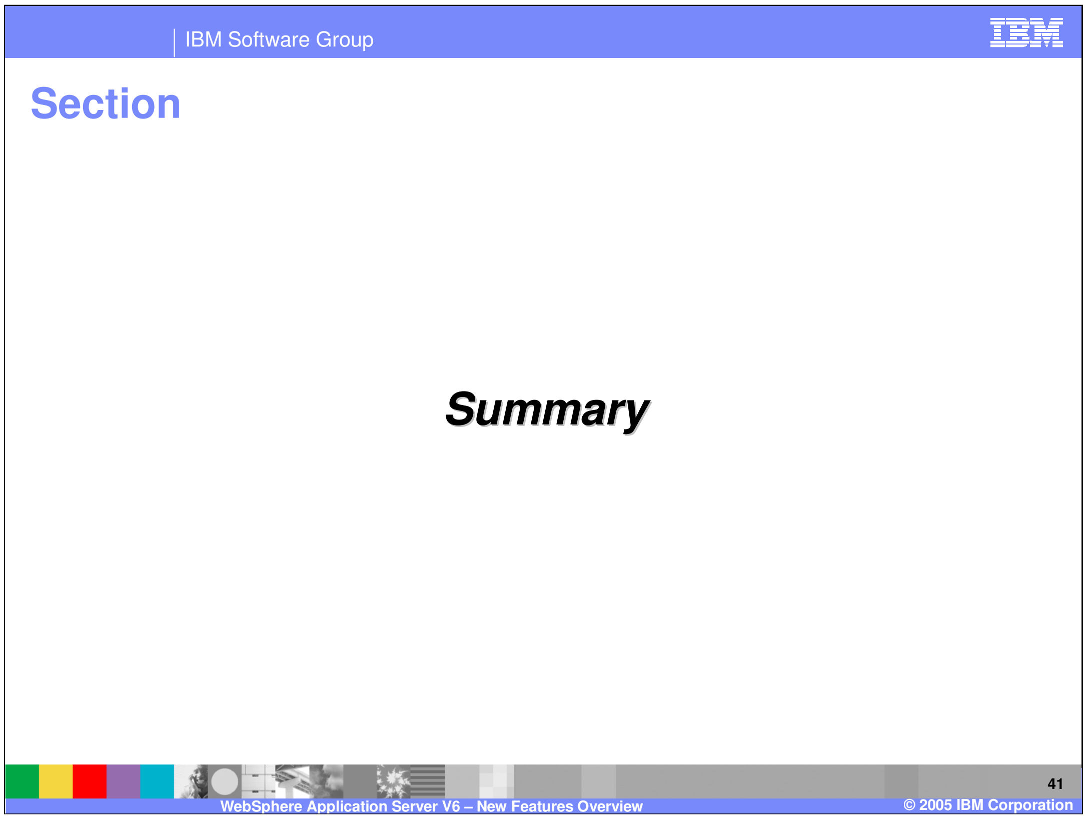
The next section will discuss Security enhancements.



Security Enhancements

- Java Authorization Contract with Containers (JACC) 1.0 support
  - Allows plug-in of your Authorization servers
  - JACC compliant TAM (Tivoli Access Manager) shipped with V6 Network Deployment
  - Continues to support the non-JACC native authorization (similar to V5)
- Security Attribute Propagation from WebSphere Application Server V5.1.1
- Implements WS-Security 1.0

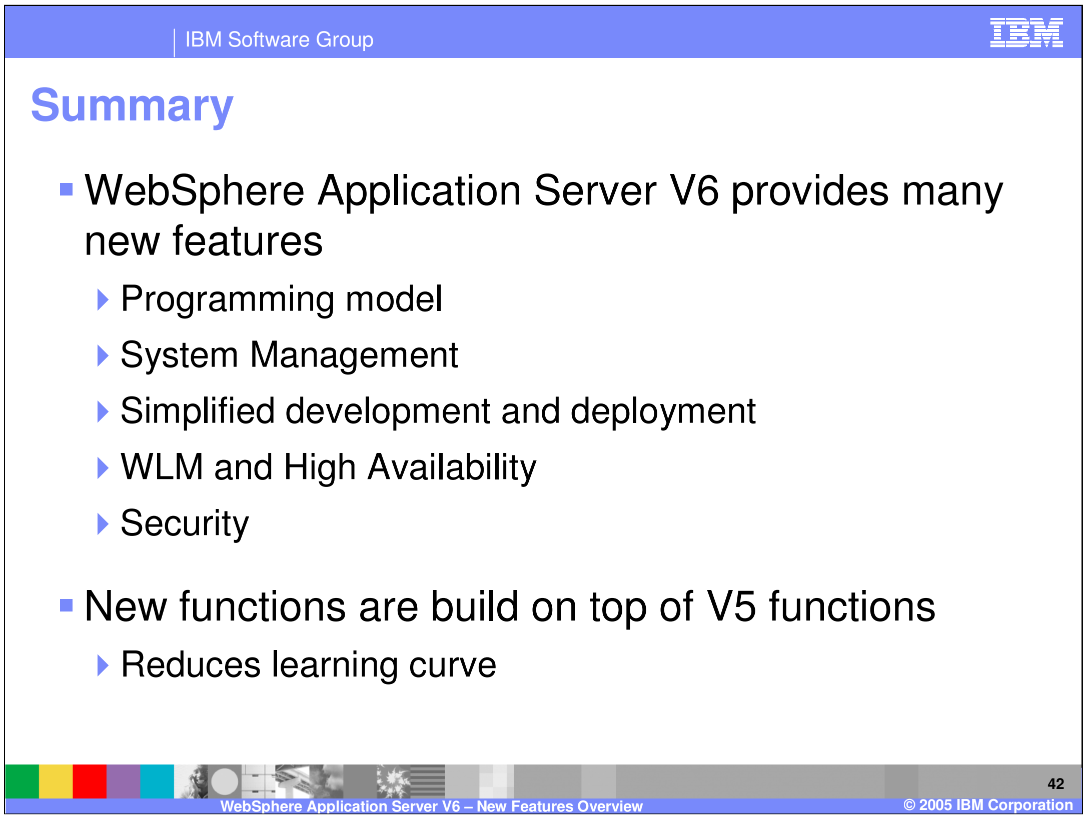
As part of the support of J2EE 1.4, WebSphere Application Server V6 provides support for Java Authorization Contract with Containers (JAAC). This support allows you to plug in your own authorization server, for example the Tivoli Access Manager which is shipped with V6 Network Deployment. Support for security attribute propagation as well as WS-Security 1.0 functionality is also included.



Section

Summary

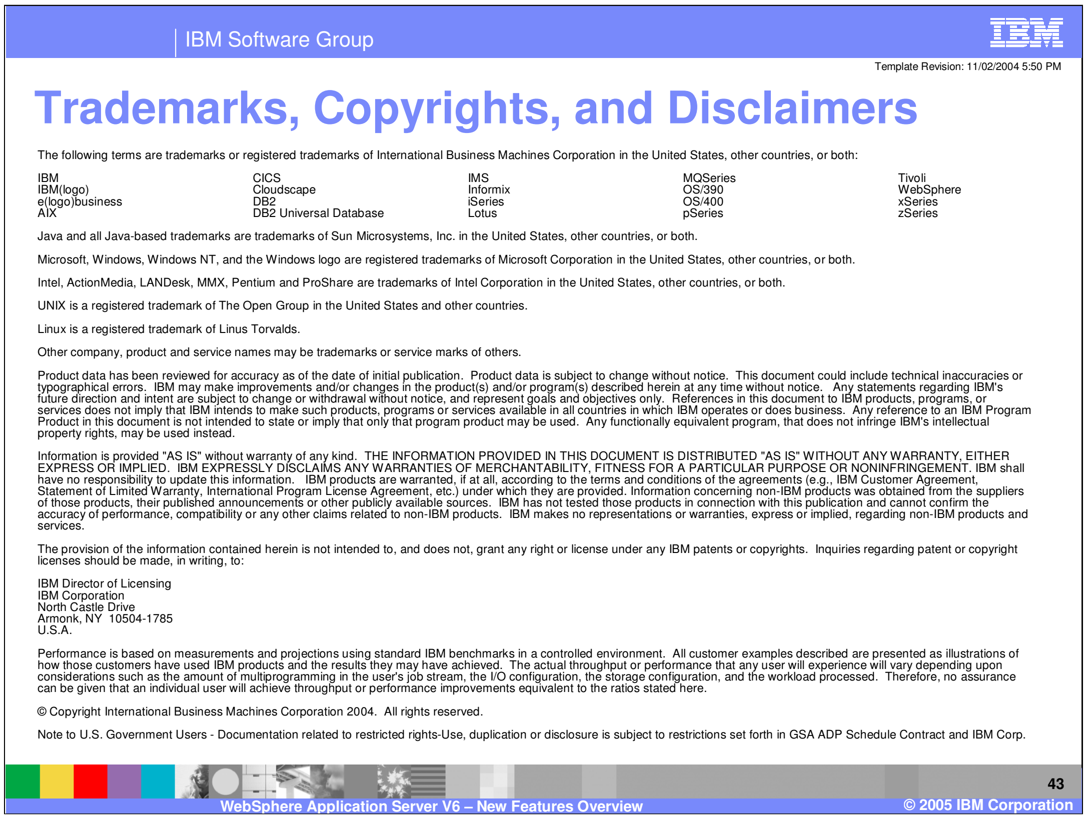
The next section will provide a summary.



Summary

- WebSphere Application Server V6 provides many new features
  - Programming model
  - System Management
  - Simplified development and deployment
  - WLM and High Availability
  - Security
- New functions are build on top of V5 functions
  - Reduces learning curve

In summary, you have learned about many of the new enhancements provided in WebSphere Application Server V6. These enhancements are in the areas of Programming Model APIs, System Management, simplified development and deployment using WebSphere Rapid Deployment, WLM and High Availability, and Security.



Trademarks, Copyrights, and Disclaimers