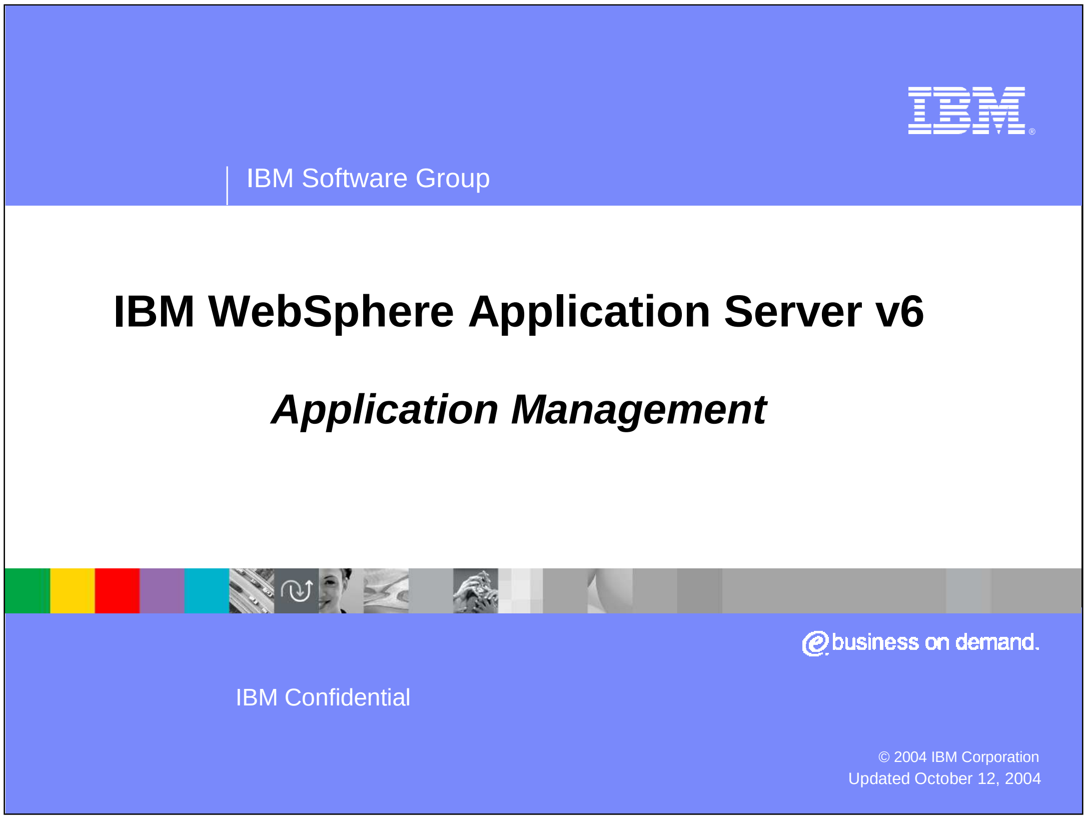


IBM Software Group

IBM WebSphere Application Server v6

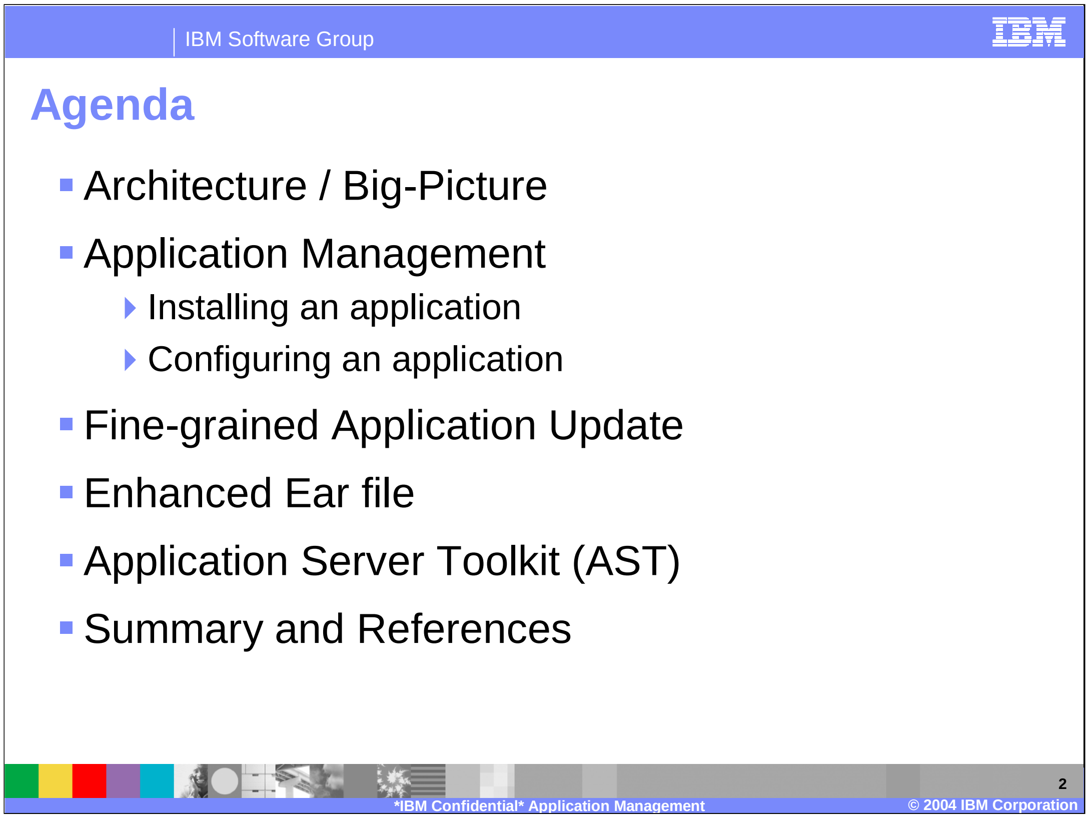
Application Management

@business on demand.

IBM Confidential

© 2004 IBM Corporation

Updated October 12, 2004



Agenda

- Architecture / Big-Picture
- Application Management
    - Installing an application
    - Configuring an application
- Fine-grained Application Update
- Enhanced Ear file
- Application Server Toolkit (AST)
- Summary and References

© 2004 IBM Corporation

The agenda for this presentation applies to both the Network Deployment packaging and the Express packaging of WebSphere Application Server. We will give a brief overview of the architecture and big picture, we will talk about installing an application, configuring an application, about fine grained application update, the enhanced ear file, and then application server toolkit.

IBM Software Group

Section

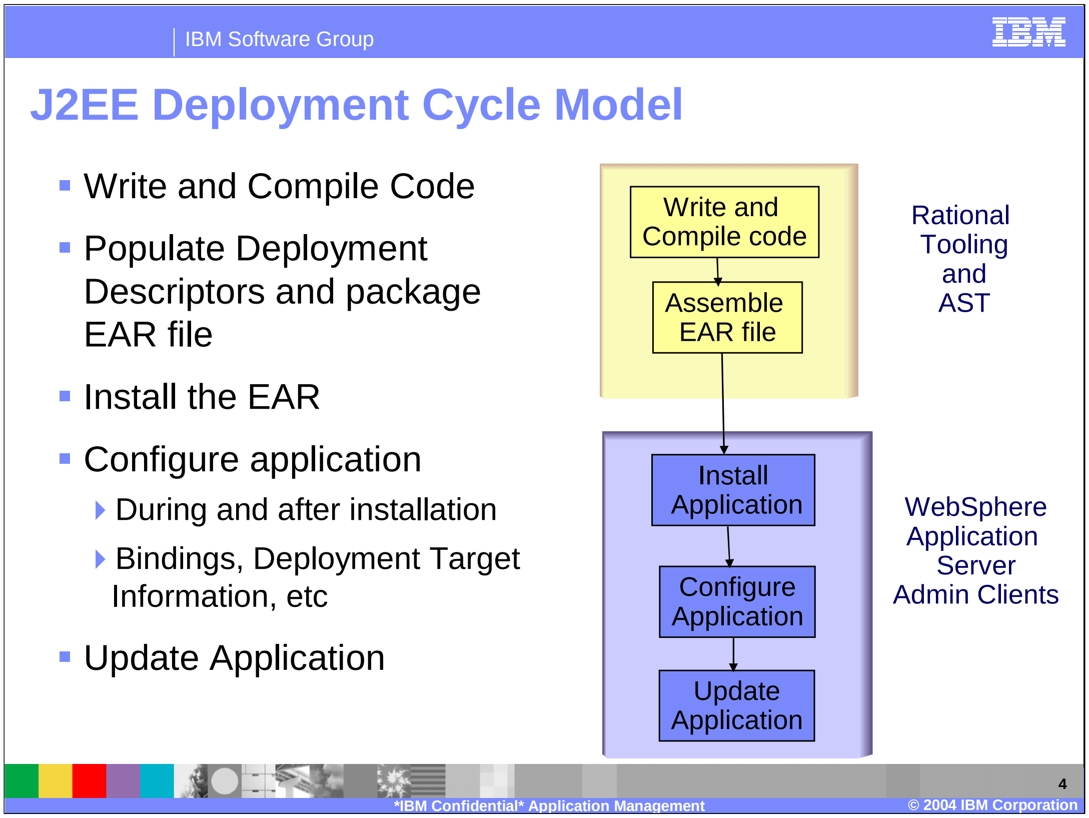
Architecture / Big-Picture

IBM Confidential Application Management

© 2004 IBM Corporation

3

Let’s now look at the application architecture and big picture.



IBM Software Group

IBM

J2EE Deployment Cycle Model

- Write and Compile Code
- Populate Deployment Descriptors and package EAR file
- Install the EAR
- Configure application
  - During and after installation
  - Bindings, Deployment Target Information, etc
- Update Application

Write and Compile code

Assemble EAR file

Rational Tooling and AST

Install Application

Configure Application

Update Application

WebSphere Application Server Admin Clients

4

IBM Confidential Application Management

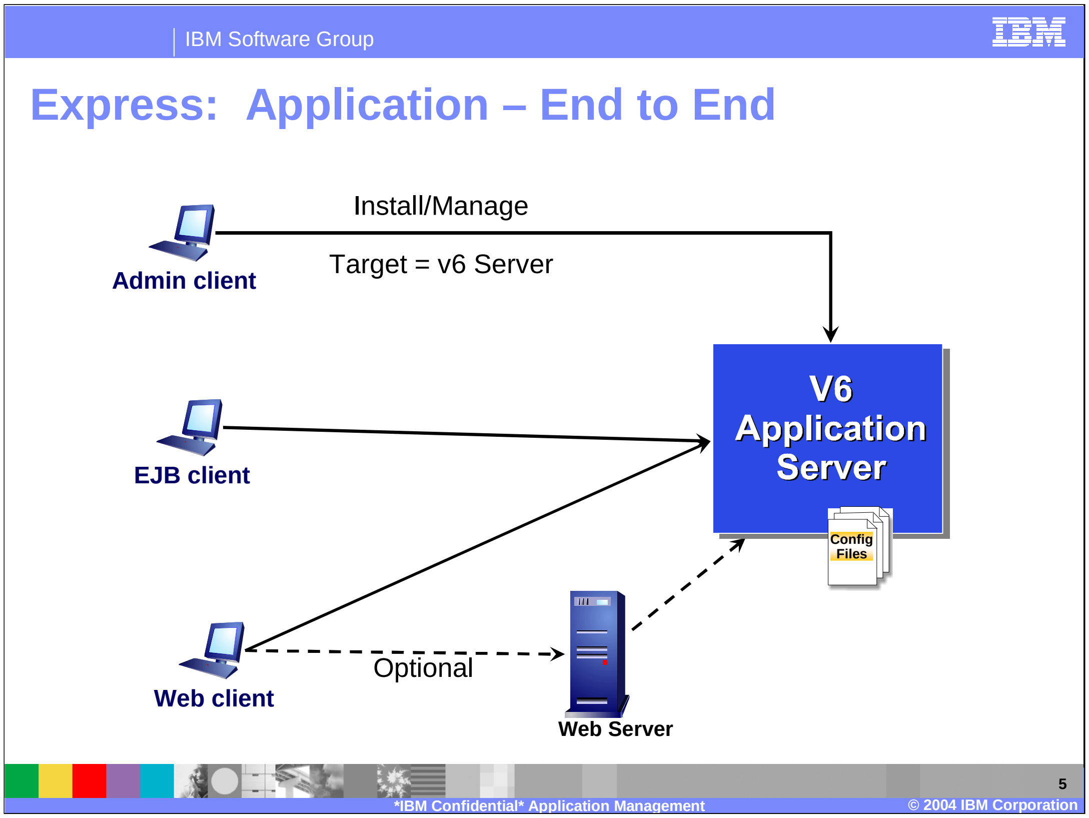
© 2004 IBM Corporation

The developer’s tasks should focus on developing code, but there are many other responsibilities that are the result of the current J2EE deployment model. For example, packaging code for distribution should be a simple step, aided by tooling - but in reality, it is just one more skill that the developer must maintain, one more set of files that needs to be updated.

In this slide we want to differentiate between the roles of a developer and an administrator, and the different tools to that are available in WebSphere for these roles.

For a developer, we have the Rational tooling - that is, IBM Rational Application Developer, and Application Server Toolkit. The Application Server Toolkit is our assembly tool which is based on Eclipse. You can write and compile your J2EE applications using the Rational tooling, you can assemble the enterprise application, that is, populate the deployment descriptors and package the EAR file using the application server toolkit. Once you are done with that, and you have the enterprise application ready for installation, you can move over to the WebSphere Application Server administrative clients. That can be either the administrative console, the Wsadmin command line utility, or a pure Java client that uses the JMX programming model. The configuration of the application, which includes binding information and deployment target information, may be configured during, or after installation.

Once you have your application up and running, at some point in the future, you might want to update the application. It is in this application update step that we have some major changes in version six. The rest of it is fairly similar to version five, except, of course, that we have the Rational Application Developer tooling and J2EE is now at the 1.4 level.



IBM Software Group

Express: Application – End to End

Install/Manage

Target = v6 Server

Admin client

EJB client

Web client

Optional

Web Server

V6 Application Server

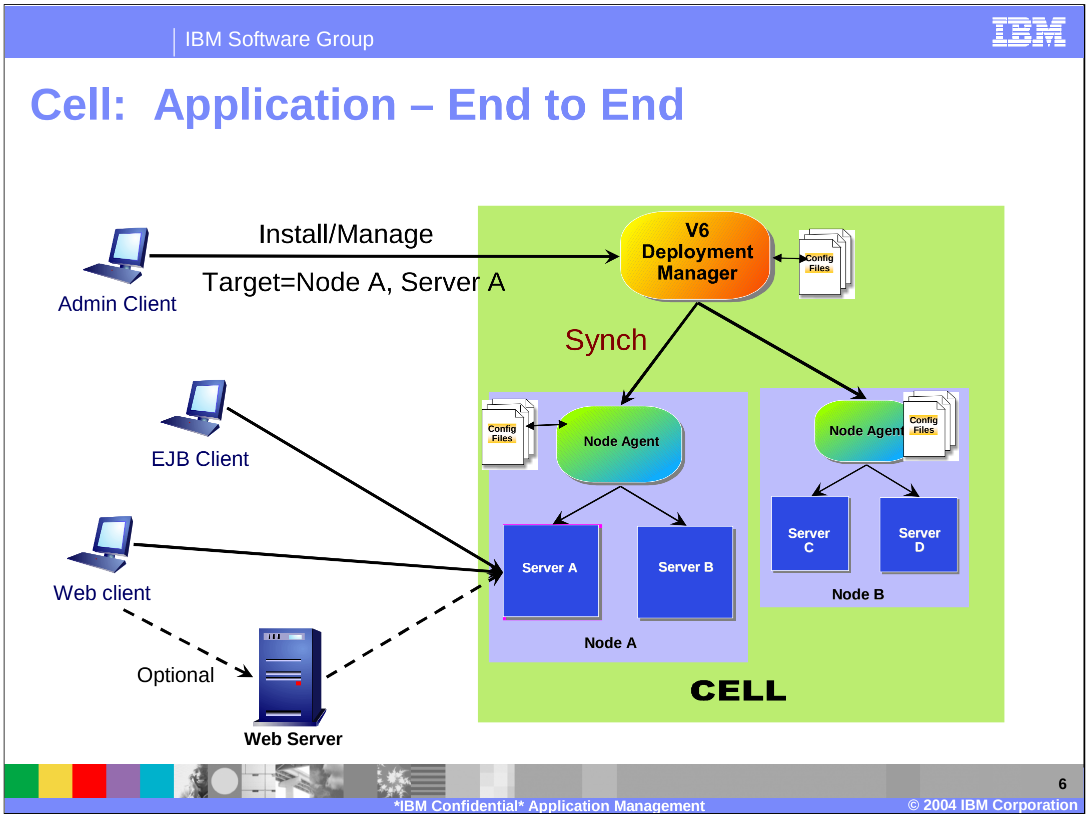
Config Files

5

IBM Confidential Application Management

© 2004 IBM Corporation

This is a view of how to install and manage an application on WebSphere application server express, which is our single stand alone application server. We show the version six application server and the XML configuration files. Any application server client, such as the admin console, wsadmin, or a custom MBean client , can be used to install an application under the version six application server. Once you have the application installed on the application server, then you can use the application clients to access the application.



IBM Software Group

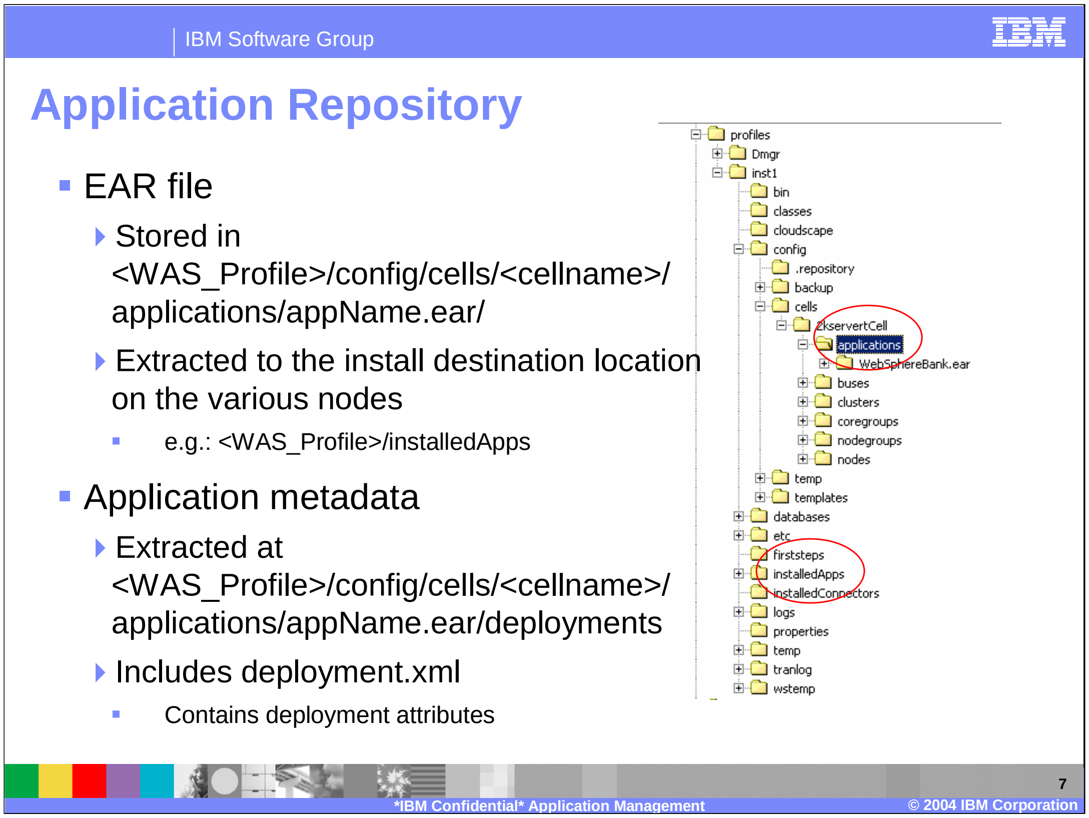
Cell: Application – End to End

6

IBM Confidential Application Management

© 2004 IBM Corporation

The Network Deployment picture is different, with the Deployment Manager and the Node Agents. Briefly, all the administration is done through the deployment manager. You install your application by connecting the administrative client to the deployment manager, and the nodes where the application is expected to run, pull the configuration from deployment manager. Your EJB and web clients, like in the case of Express, would be directly connecting to the application server.



IBM Software Group

Application Repository

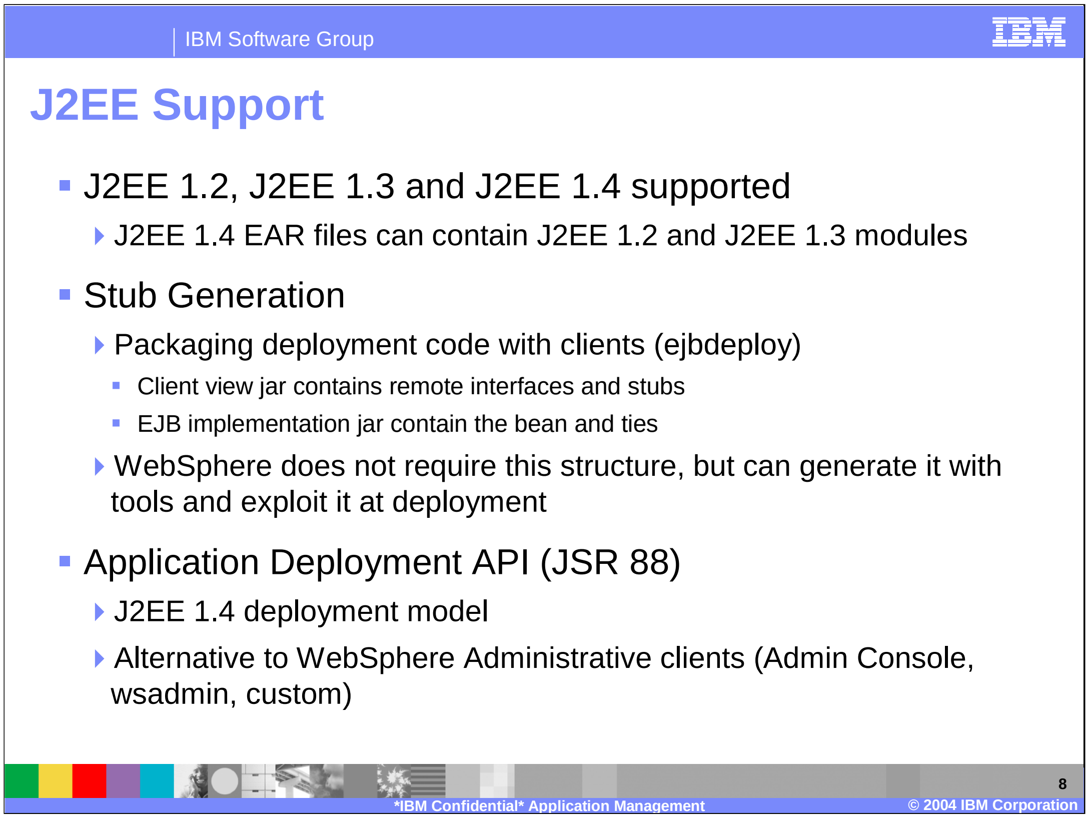
- EAR file
  - Stored in <WAS_Profile>/config/cells/<cellname>/ applications/appName.ear/
  - Extracted to the install destination location on the various nodes
    - e.g.: <WAS_Profile>/installedApps
- Application metadata
  - Extracted at <WAS_Profile>/config/cells/<cellname>/ applications/appName.ear/deployments
  - Includes deployment.xml
    - Contains deployment attributes

7

IBM Confidential Application Management

© 2004 IBM Corporation

The enterprise application, or EAR file, is stored under WebSphere profiles, profile name, config, cells, cell name, applications. The application is also extracted to the install destination location on the various nodes. By default, the location where it is extracted all is the installedApps directory. The metadata, that is your deployment descriptor files, is extracted under config/cells/cellname/applications/appName.ear (which would be a folder)/Deployments. In this folder would be a file called deployment.xml, which contains deployment attributes, such as module to deployment target relations, Classloader settings and so on.



IBM Software Group

J2EE Support

- J2EE 1.2, J2EE 1.3 and J2EE 1.4 supported
  - J2EE 1.4 EAR files can contain J2EE 1.2 and J2EE 1.3 modules
- Stub Generation
  - Packaging deployment code with clients (ejbdeploy)
    - Client view jar contains remote interfaces and stubs
    - EJB implementation jar contain the bean and ties
  - WebSphere does not require this structure, but can generate it with tools and exploit it at deployment
- Application Deployment API (JSR 88)
  - J2EE 1.4 deployment model
  - Alternative to WebSphere Administrative clients (Admin Console, wsadmin, custom)

8

IBM Confidential Application Management

© 2004 IBM Corporation

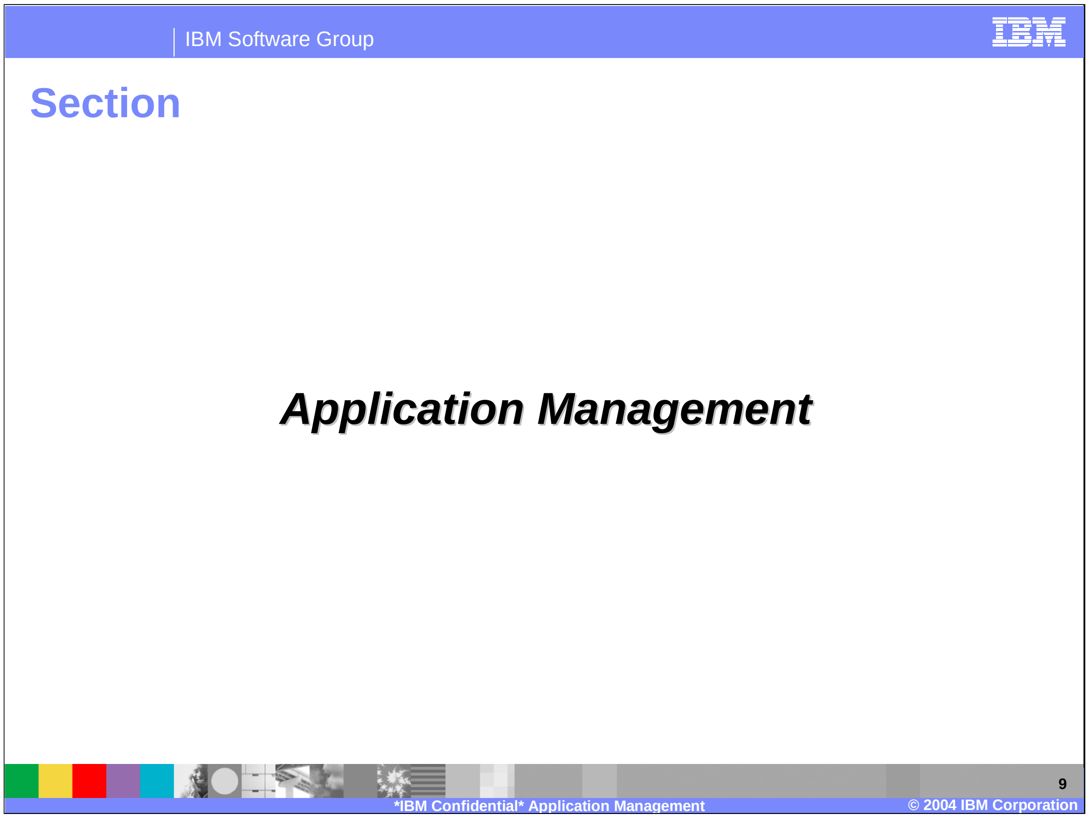
J2EE 1.2, 1.3, and 1.4 are supported as part of the downward compatibility requirements of the J2EE 1.4 specification.

Stub generation is actually not new in version six. The ejbdeploy tool creates and EJB jar file and a client view jar file. This client view jar file will contain the remote interfaces and the stubs. The implementation jar file will only contain the bean and the ties. The client view char file would be referenced by the EJB implementation jar file, as well as any EJB client jars. This structure is not really required by WebSphere, but the tooling can create it, and WebSphere can exploit it during deployment.

The third bullet is about JSR 88, which is a new application deployment API that is part of the J2EE 1.4 specification.

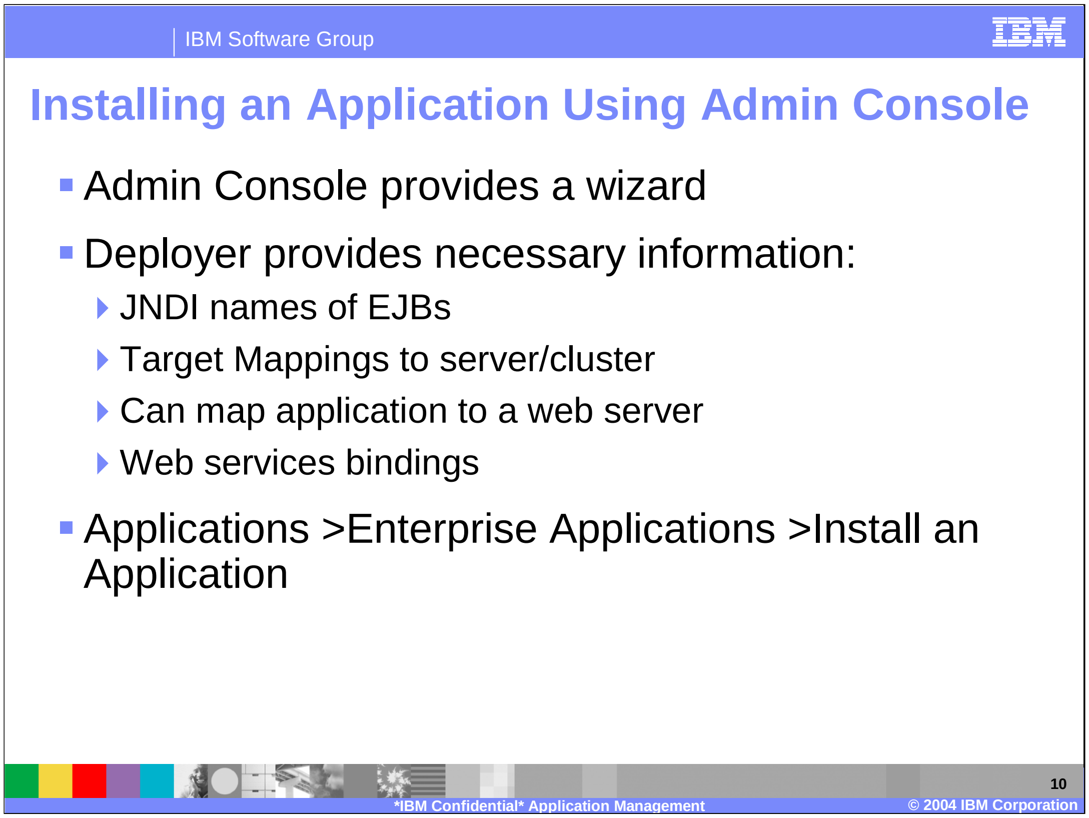
JSR-88 (Application Deployment API) is a contract between a tool vendor and platform. It was the only deployment vehicle for Component Test Suite 1.4. It provides yet another deployment model for WAS in addition to wsadmin, Admin Console, MBean.

Any of the non- j2ee functionality, such as PME, cannot be done using JSR 88. The JSR 88 spec doesn't have extensibility. It is therefore not a recommended deployment tool. Instead, it is recommended that you use the WebSphere administrative clients for your application deployment.



Section

Application Management



IBM Software Group

Installing an Application Using Admin Console

- Admin Console provides a wizard
- Deployer provides necessary information:
  - JNDI names of EJBs
  - Target Mappings to server/cluster
  - Can map application to a web server
  - Web services bindings
- Applications >Enterprise Applications >Install an Application

10

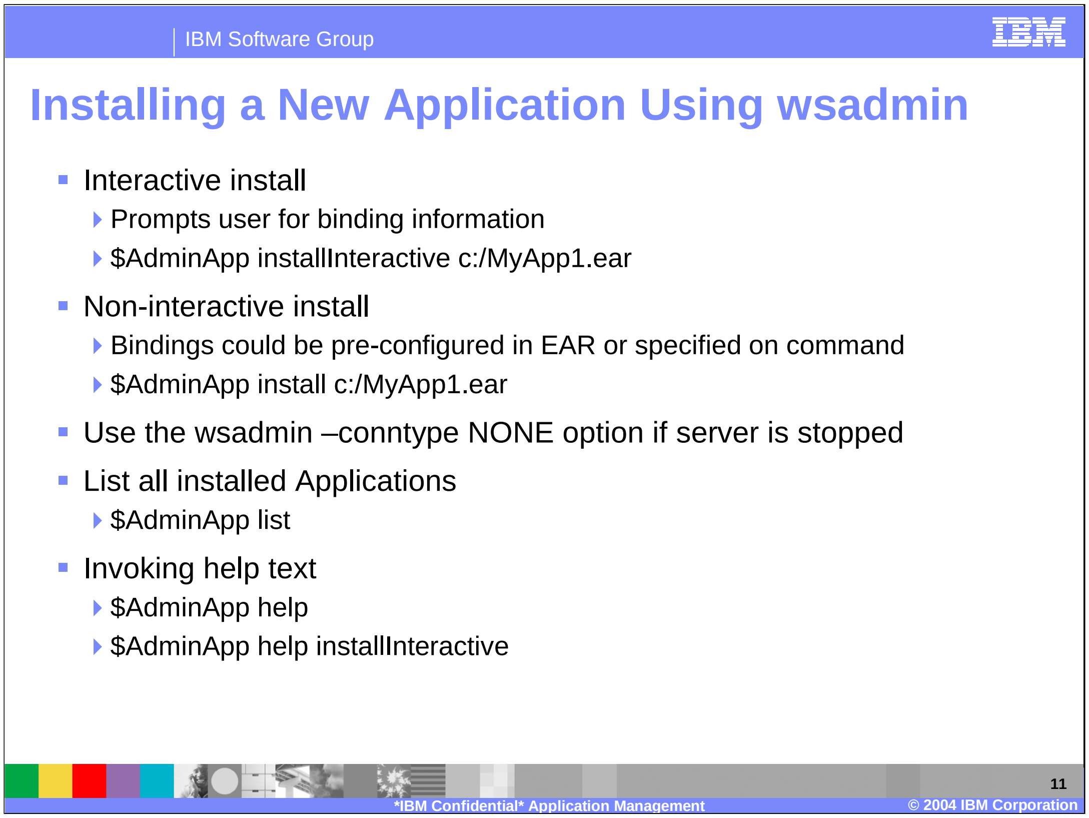
IBM Confidential Application Management

© 2004 IBM Corporation

The administrative console provides a wizard for installing the application. This is very similar to version five. The deployer will provide necessary information during the installation of the application, such as JNDI names of an EJB, bindings, and target mappings to servers or clusters, which means which servers the application will run on.

Something new in version six is the ability to specify a web server as a target. So in addition to associating an application with a specific application server, we can also associate an application with a specific web server. Generating the plug in for the http server in version six is application centric, meaning that WebSphere will create a plug-in file for each web server based on the list of applications that are deployed on that web server.

Finally, Web Services bindings can be specified during the installation of an application. To install application using the administrative console, under Applications click Enterprise Applications then Install an Application.



IBM Software Group

Installing a New Application Using wsadmin

- Interactive install
  - Prompts user for binding information
  - $AdminApp installInteractive c:/MyApp1.ear
- Non-interactive install
  - Bindings could be pre-configured in EAR or specified on command
  - $AdminApp install c:/MyApp1.ear
- Use the wsadmin –conntype NONE option if server is stopped
- List all installed Applications
  - $AdminApp list
- Invoking help text
  - $AdminApp help
  - $AdminApp help installInteractive

11

IBM Confidential Application Management

© 2004 IBM Corporation

To install an application using wsadmin you can either do in interactive install or a non interactive install. When you do an interactive install, the command line installer will prompt you to specify any binding information. You will need to specify, for example, the JNDI names for your EJBs, your resource reference bindings, and so on during installation. The method for doing an interactive install is the command “dollar AdminApp installInteractive” followed by the path to your EAR file.

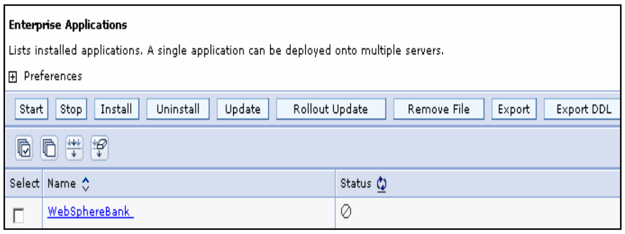
A non-interactive install is available if your application contains all the binding information. Packaging tools like the Application Server Toolkit or IBM Rational Application Developer, allow you to provide all the bindings within the EAR file. You may also specify binding information on the install command.

It is worth noting that once you run an interactive install, the non interactive command for the exact same thing is placed in the logs, which can be very helpful in automating your future installs.

Using the AdminApp command within wsadmin is available in even if the server is not running, using the command line option conntype none. Using this connection type places wsadmin in local mode where it interacts directly with the XML files that are the repository. Not all of the commands in wsadmin are available in local mode however. One example is the start/stop Application commands.

In order to see all the commands that are available for the AdminApp object, enter “AdminApp help”. For specific command help, enter “AdminApp help” and then the command.

Other Operations



Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

- Admin Console has an uninstall function
- Resources associated with the app are not removed
- Uninstalling apps using wsadmin
  - wsadmin –c “$AdminApp uninstall WebSphereBank.ear”
- Applications can also be started and stopped
- Export applications to file system
- Export table ddl file

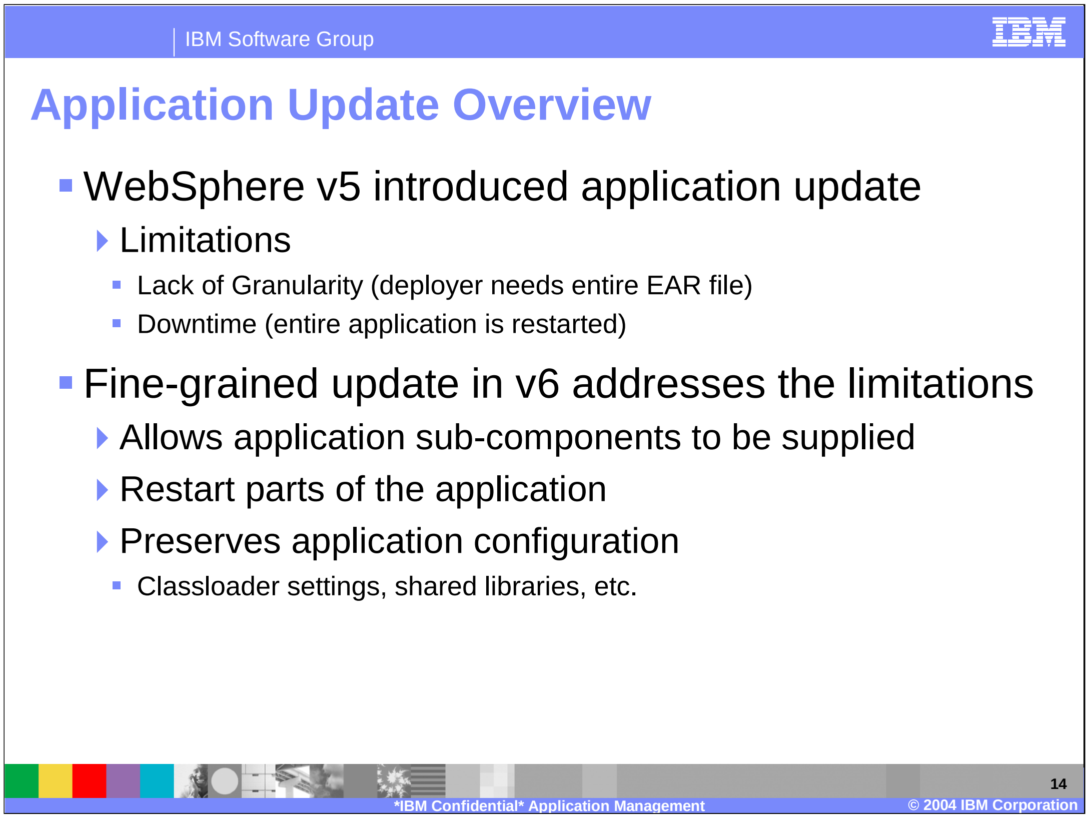
Any of the functions that are available in the Administrative Console are also available in wsadmin. One example is the uninstall function in the admin console. Keep in mind that the resources that are associated with the application are not removed, only the application configuration itself. To uninstall the WebSphereBank application using wsadmin, you would use the command wsadmin -C “$AdminApp uninstall WebSphereBank.EAR”. Applications can also be started and stopped using either the administrative console or wsadmin, applications can be exported to the file system, and it is possible to export the table data definition language file. The table DDL file describes to a database the structure of the data used by the application.

But there are two other buttons illustrated here that we haven’t mentioned yet: the Update, and the Rollout Update buttons. The update button is used to do fine grained application update, which we will talk about in the last part of this presentation. The Rollout Update button is used to rollout the update to one node at a time, thus keeping the application running on at least some of the nodes at any given time.

Section

Fine-grained Application Update

© 2004 IBM Corporation



IBM Software Group

Application Update Overview

- WebSphere v5 introduced application update
  - Limitations
    - Lack of Granularity (deployer needs entire EAR file)
    - Downtime (entire application is restarted)
- Fine-grained update in v6 addresses the limitations
  - Allows application sub-components to be supplied
  - Restart parts of the application
  - Preserves application configuration
    - Classloader settings, shared libraries, etc.

14

IBM Confidential Application Management

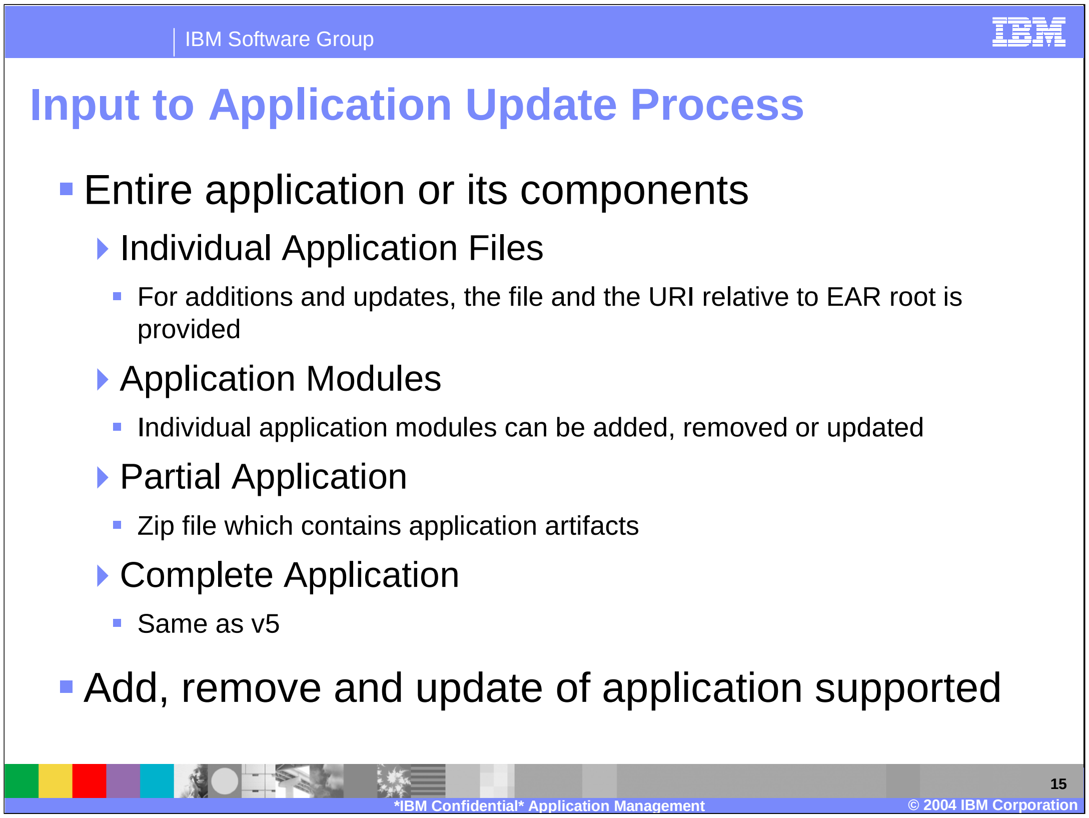
© 2004 IBM Corporation

The application management framework in WebSphere 5.0 introduced an *application update* function which was exposed via various administrative programs such as web-based administrative console, wsadmin scripting tool as well as the programmatic MBean interface. This update function took a new application EAR file for an application deployed on the WebSphere platform and performed the necessary deployment steps to replace the existing application configuration with the new one. To make the new application configuration take effect the WebSphere application management logic stopped the running application, replaced the application files and restarted the application. There were two main problems with the update function –

Granularity of update input (or the lack thereof) - The update could only be initiated if the deployer had the entire EAR file that was to be used to update the deployed application. It was not possible to *patch* a deployed application using partial contents; for example, a single file or a module.

Downtime during update - The entire application had to be recycled (stopped and restarted) in order to apply the updates. Extra steps needed to be taken to ensure continuous availability of application artifacts.

The fine-grained update feature in WebSphere 6.0 is intended to address the above problems. You can now apply a patch – for example, supply a single JSP and have WebSphere replace the same named JSP in the running application. It is now possible to restart parts of the application, and the application update process preserves configuration settings such as bindings. Classloader settings, shared libraries, and so on.



IBM Software Group

Input to Application Update Process

- Entire application or its components
  - Individual Application Files
    - For additions and updates, the file and the URI relative to EAR root is provided
  - Application Modules
    - Individual application modules can be added, removed or updated
  - Partial Application
    - Zip file which contains application artifacts
  - Complete Application
    - Same as v5
- Add, remove and update of application supported

15

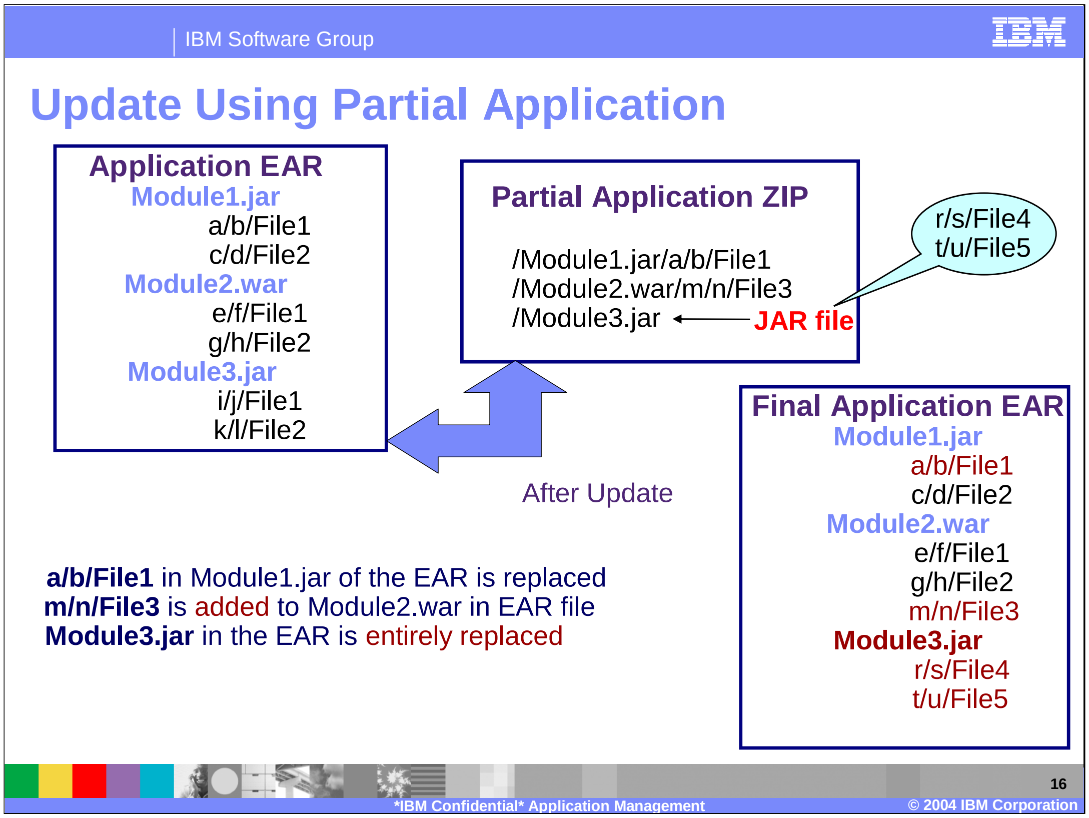
IBM Confidential Application Management

© 2004 IBM Corporation

The input to the Application update process can either be an entire EAR file, or individual components. Those components may be individual files, such as a JSP or a servlet. The administrator would also need the URI relative to the EAR root. You can also update complete web modules, or EJB Jar files in an application.

A third option is to provide a partial application, which would be a .zip file which contains application artifacts. This would be a likely input for updating multiple files in the application.

Finally, the three options for changing an application are Add, Remove, and Update.



IBM Software Group

Update Using Partial Application

Application EAR
Module1.jar
a/b/File1
c/d/File2
Module2.war
e/f/File1
g/h/File2
Module3.jar
i/j/File1
k/l/File2

Partial Application ZIP

/Module1.jar/a/b/File1
/Module2.war/m/n/File3
/Module3.jar ← **JAR file**

r/s/File4
t/u/File5

After Update

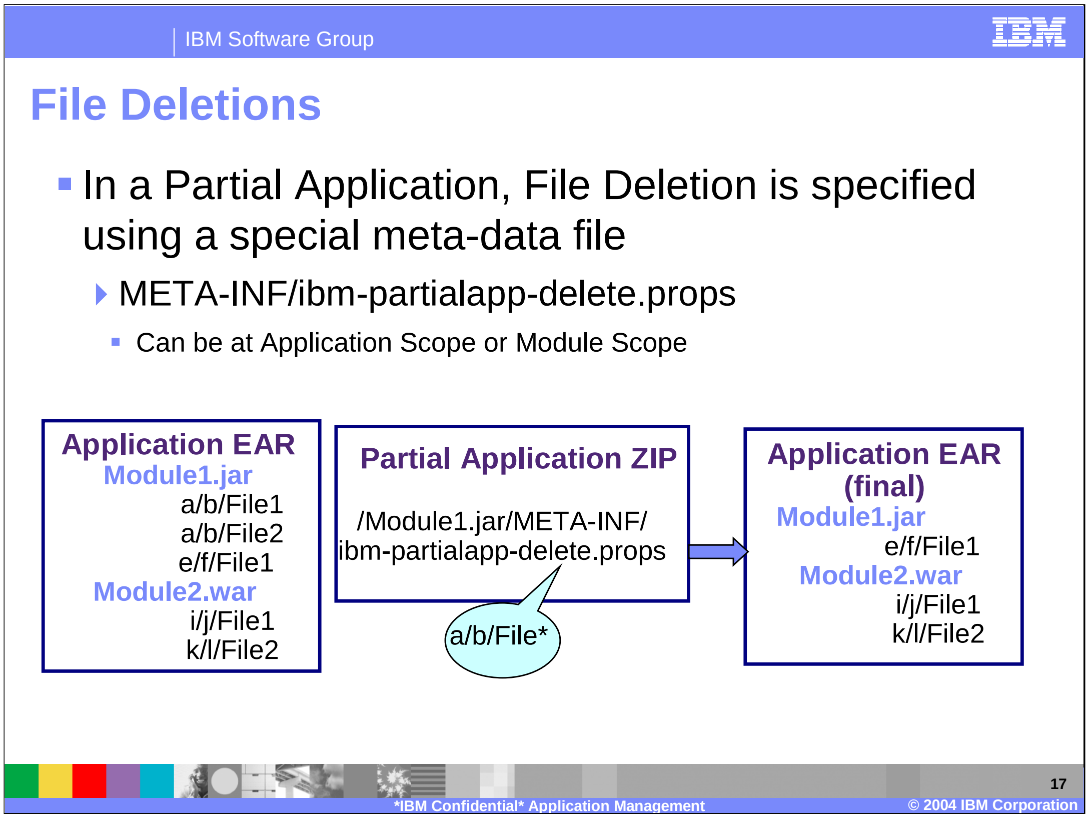
Final Application EAR
Module1.jar
a/b/File1
c/d/File2
Module2.war
e/f/File1
g/h/File2
m/n/File3
Module3.jar
r/s/File4
t/u/File5

a/b/File1 in Module1.jar of the EAR is replaced
m/n/File3 is added to Module2.war in EAR file
Module3.jar in the EAR is entirely replaced

16
IBM Confidential Application Management
© 2004 IBM Corporation

Here we have a client application EAR file which is currently running in WebSphere. The application EAR file have three modules. A Module1.jar which contains File 1 and File 2 in different directories, a Module2.war which contains a File 1 and File 2 in different directories, and Module 3, which contains File 1 and File 2 in still other directories. In this example, we need to update the application with a new version of the code, so we are providing a partial application. The partial application zip file contains the following entries: /Module1.jar /a/b/file1, which will replace the a/b/file1 in the application, and Module2.war/m/n/File3, which will be added to the application since it does not already exist, and Module3.jar, which will replace the entire contents of the existing Module3.jar in the enterprise application.

So the final application EAR file will look like the one that you see at the lower right. In Module1, file a/b/File1 is replaced with the one from the partial application zip file. In Module2.war, there is a new file: m/n/File3. And Module3.jar has been replaced, so i/j/File1 and k/l/File2 are removed from the application and r/s/File4 and t/u/File5 are added.



File Deletions

- In a Partial Application, File Deletion is specified using a special meta-data file
  - META-INF/ibm-partialapp-delete.props
    - Can be at Application Scope or Module Scope

Application EAR
Module1.jar
a/b/File1
a/b/File2
e/f/File1
Module2.war
i/j/File1
k/l/File2

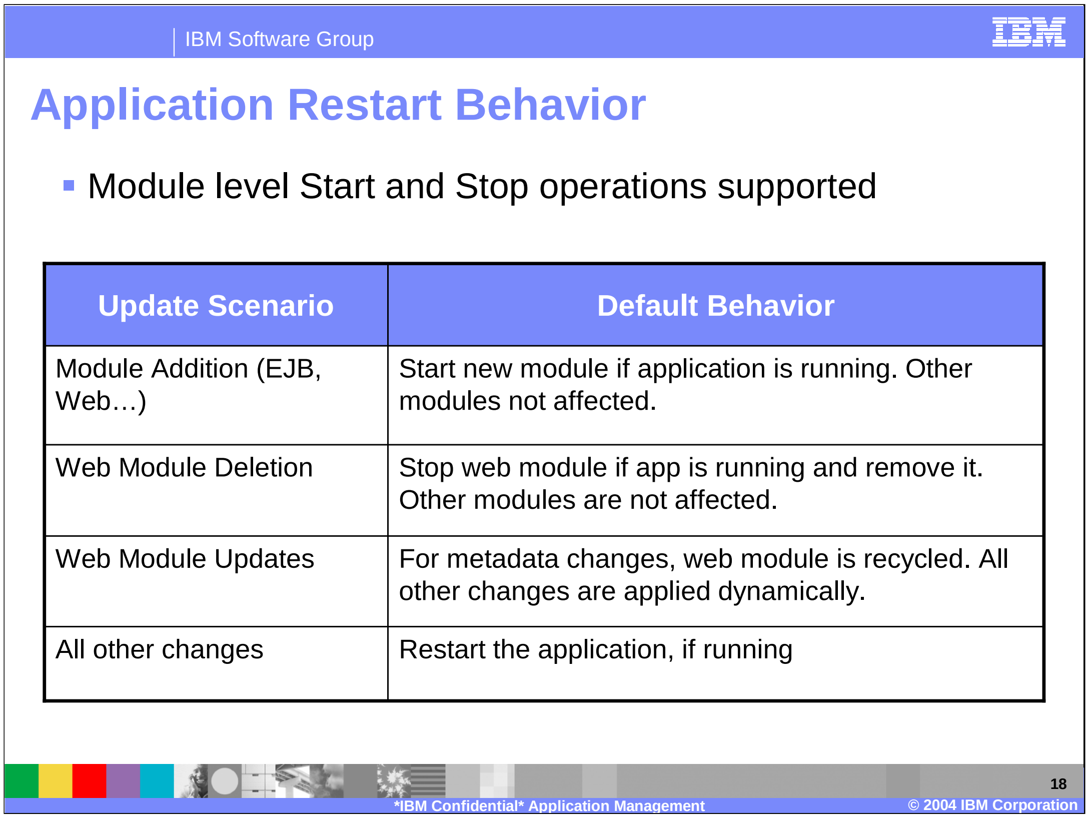
Partial Application ZIP
/Module1.jar/META-INF/
ibm-partialapp-delete.props

a/b/File*

Application EAR (final)
Module1.jar
e/f/File1
Module2.war
i/j/File1
k/l/File2

© 2004 IBM Corporation

In a partial application, file deletions can be specified using a special metadata file META-INF/ibm-partialapp-delete.props at application or module scope in the hierarchical structure. Each line in the file represents a pattern of application or module scoped URIs that are to be deleted. The pattern can be specified as a regular expression using constructs defined in java.util.regex.Pattern.



Application Restart Behavior

- Module level Start and Stop operations supported

Update Scenario	Default Behavior
Module Addition (EJB, Web…)	Start new module if application is running. Other modules not affected.
Web Module Deletion	Stop web module if app is running and remove it. Other modules are not affected.
Web Module Updates	For metadata changes, web module is recycled. All other changes are applied dynamically.
All other changes	Restart the application, if running

© 2004 IBM Corporation

The WebSphere 5.x application server only allowed starting and stopping of the entire application. As a part of fine-grained application update support, the WebSphere server runtime is enhanced to support module level start and stop operations, making it possible to recycle only parts of a running application when it is updated.

The following rules apply to module level start/stop operations:

Starting a module makes module components accessible to clients. A module can be started only if its parent application is running. Starting an application, starts all the modules deployed on that server, but the converse is not true.

Module start operation is supported on all module types such as EJB, Web and Connector.

Stopping an application makes module components inaccessible to clients. Stopping the last running application module on an application server does not stop the parent application on that server.

Module stop operation is supported only for Web modules, and *only* if the WAR classloader policy for the deployed application is MODULE (i.e. there is a classloader per WAR module). If the WAR classloader policy is APPLICATION (i.e. all WARs of the application share a single classloader which is the application classloader) then the entire application needs to be recycled (stopped and re-started).

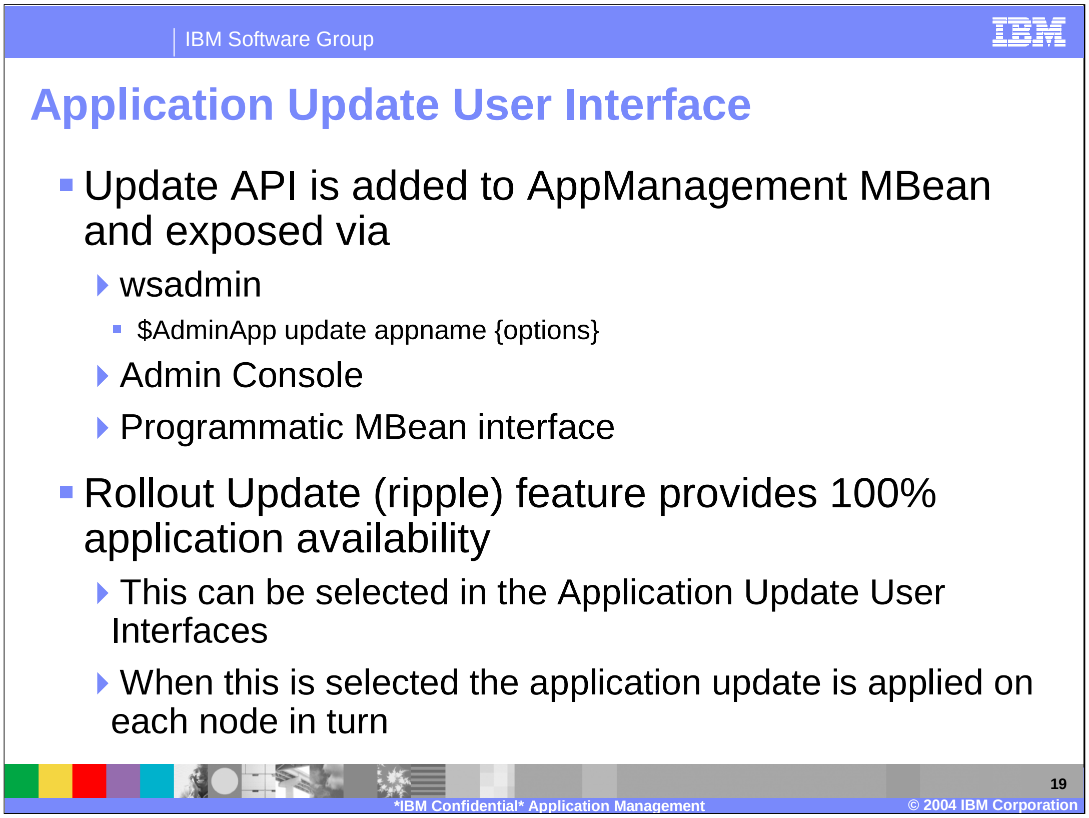
Please note that you can not start or stop a module manually. It only happens as part of update operation.

For Module Additions, modules can be added on the fly without stopping the running application. However, if there are security roles defined in the DD for the module, and security provider does not support dynamic updates, then the application is recycled.

For Web Module Deletions, the module is stopped and the module files are deleted. For other modules (EJB, connector), the application is recycled.

For Updates of Web modules, if meta-data is changed, then the module is recycled. If only non-metadata artifacts (servlets, JSPs) are updated, nothing is recycled.

For Update of other modules, the entire application is recycled.



Application Update User Interface

- Update API is added to AppManagement MBean and exposed via
  - wsadmin
    - $AdminApp update appname {options}
  - Admin Console
  - Programmatic MBean interface
- Rollout Update (ripple) feature provides 100% application availability
  - This can be selected in the Application Update User Interfaces
  - When this is selected the application update is applied on each node in turn

The update function is added to the AdminApp object exposed by the wsadmin command line utility. It is also exposed to using the administrative console, and the wizard is very similar to the install wizard. You can also install applications using the programmatic MBean interface.

The command line syntax is **$AdminApp update appname contenttype {options}**

where contenttype can be file, module, partialapp, or application

and where options are

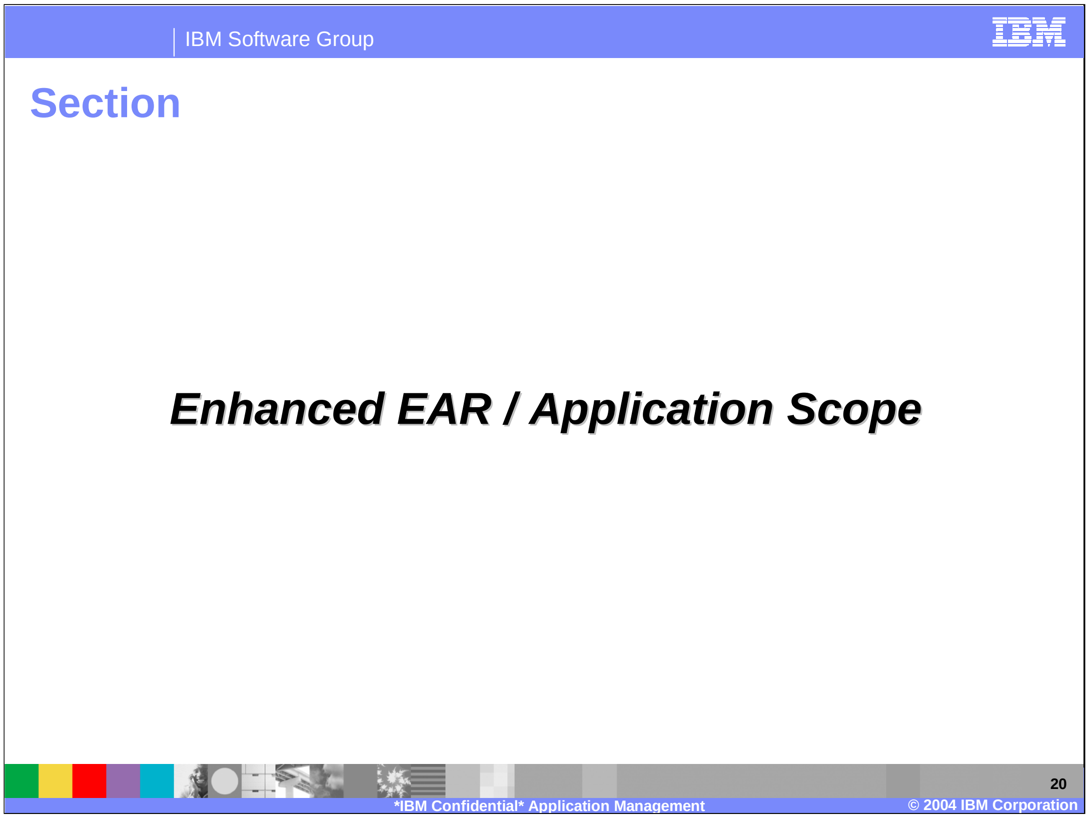
-contentURI : URI of the file/module

-contents : filename of the contents to be updated

-operation : add, delete, update, or addupdate

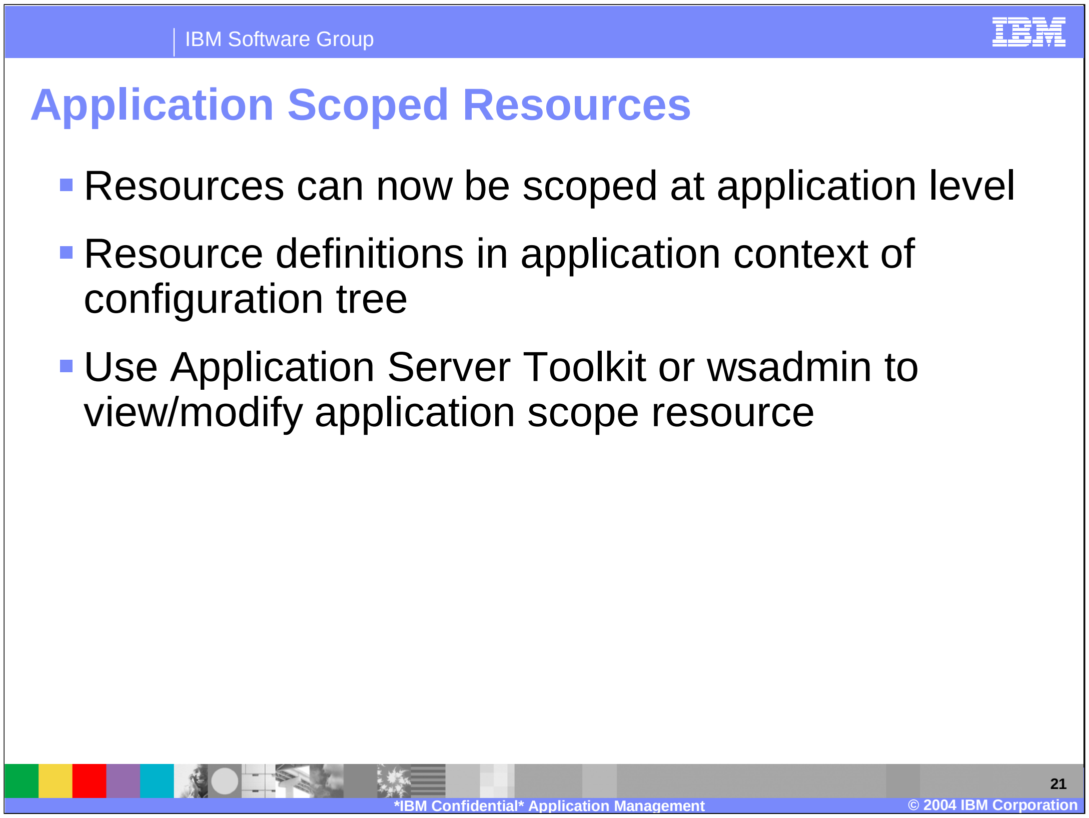
-recycle : can be default, all, none

The middle bullet point about Update rippling is for the Network Deployment package. With this option, the application updates are applied to one node at a time, so that the application is continuously available. This is the function provided by the “rollout update” button on the admin console, discussed earlier in this presentation.



Section

Enhanced EAR / Application Scope



Application Scoped Resources

- Resources can now be scoped at application level
- Resource definitions in application context of configuration tree
- Use Application Server Toolkit or wsadmin to view/modify application scope resource

© 2004 IBM Corporation

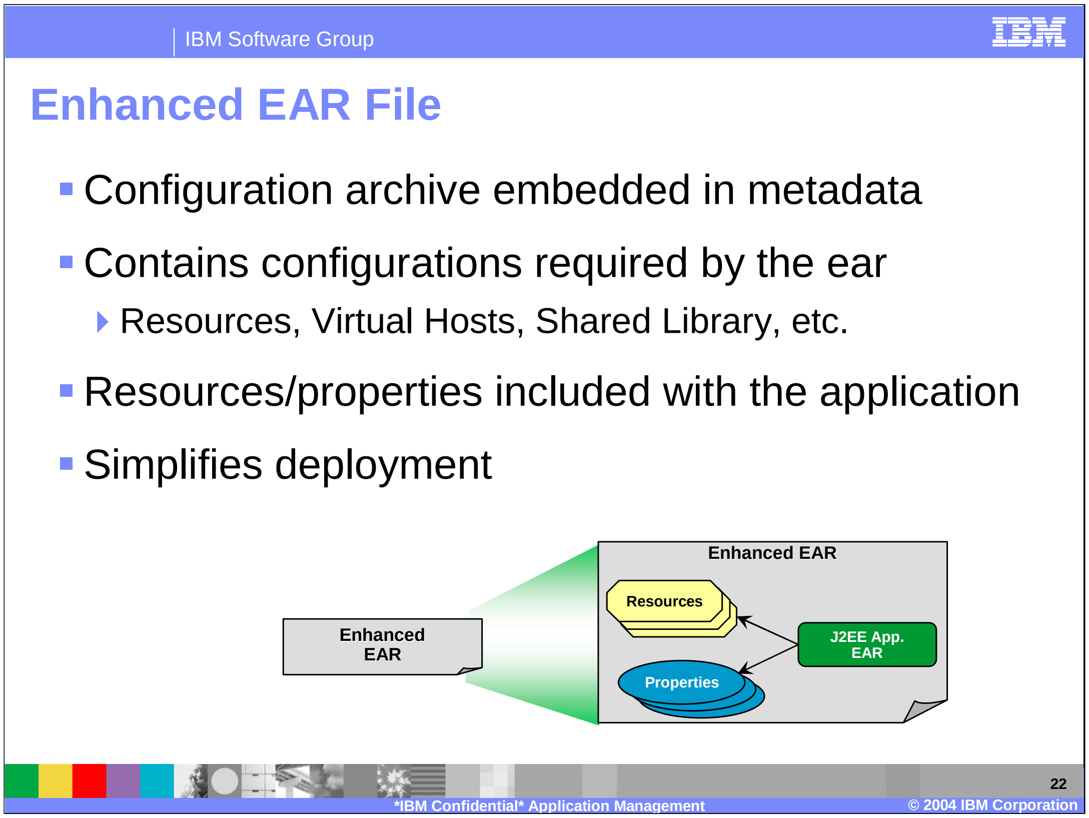
The current V5.x mechanism for storing resource definitions is topological(cell, node, etc). If an application is moved to a different server, the resource definitions must also be made available for the new runtime.

But it’s the application that needs the resources, not the server runtime. So, we need to be able to store resource definitions at the application level.

Application scope is a new scope for J2EE resource definitions.

It supports J2ee resource definitions in the application context of the configuration tree.

The Application Server Toolkit (AST) or wsadmin can be used to view/modify an application scope resource.



IBM Software Group

Enhanced EAR File

- Configuration archive embedded in metadata
- Contains configurations required by the ear
  - Resources, Virtual Hosts, Shared Library, etc.
- Resources/properties included with the application
- Simplifies deployment

Enhanced EAR
Enhanced EAR
Resources
J2EE App. EAR
Properties

22

IBM Confidential Application Management

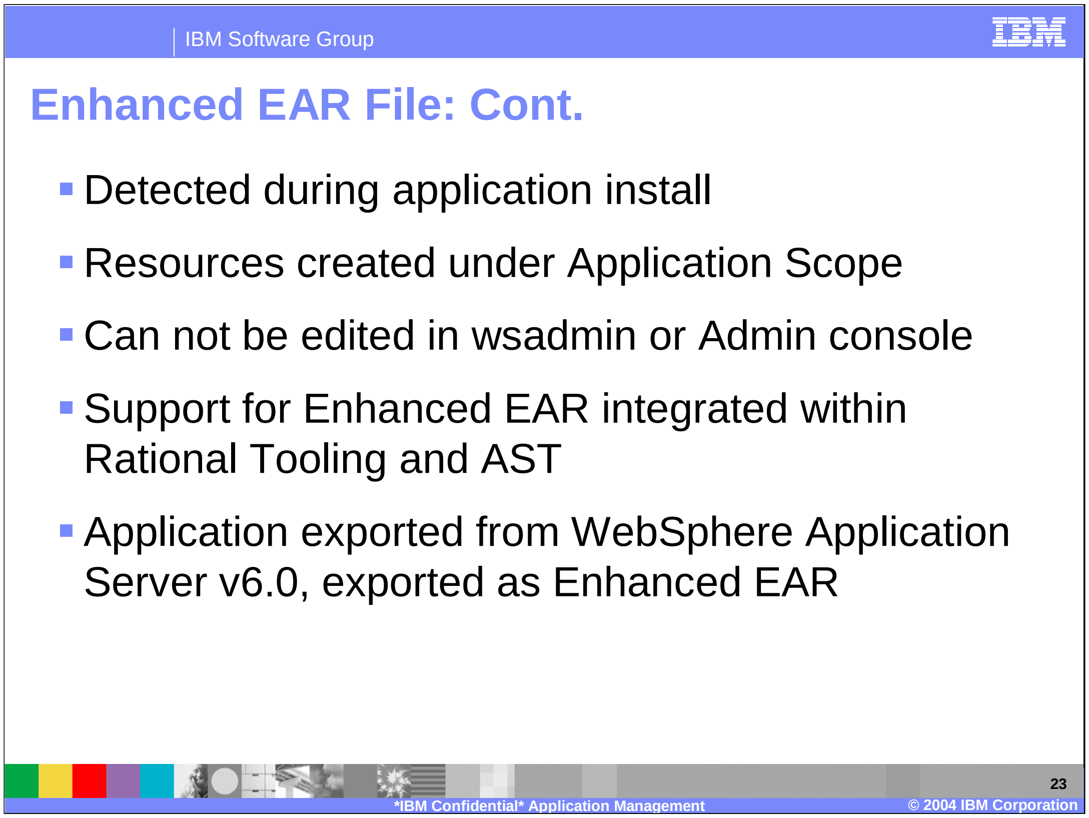
© 2004 IBM Corporation

An enhanced ear file contains a simple embedded configuration archive.

This is where users can define the required configurations for the application, such as, Resources, Virtual Hosts, Shared Library, etc.

The application resources and properties come with the application. All the application information needed to install the application on the Application Server.

In this way, users can deploy an application and create its required configurations all in one shot. Also, moving application from one server to another moves the resources as well.



IBM Software Group

Enhanced EAR File: Cont.

- Detected during application install
- Resources created under Application Scope
- Can not be edited in wsadmin or Admin console
- Support for Enhanced EAR integrated within Rational Tooling and AST
- Application exported from WebSphere Application Server v6.0, exported as Enhanced EAR

23

IBM Confidential Application Management

© 2004 IBM Corporation

During the application installation, the configuration archive is detected and the configurations defined within the configuration archive are created.

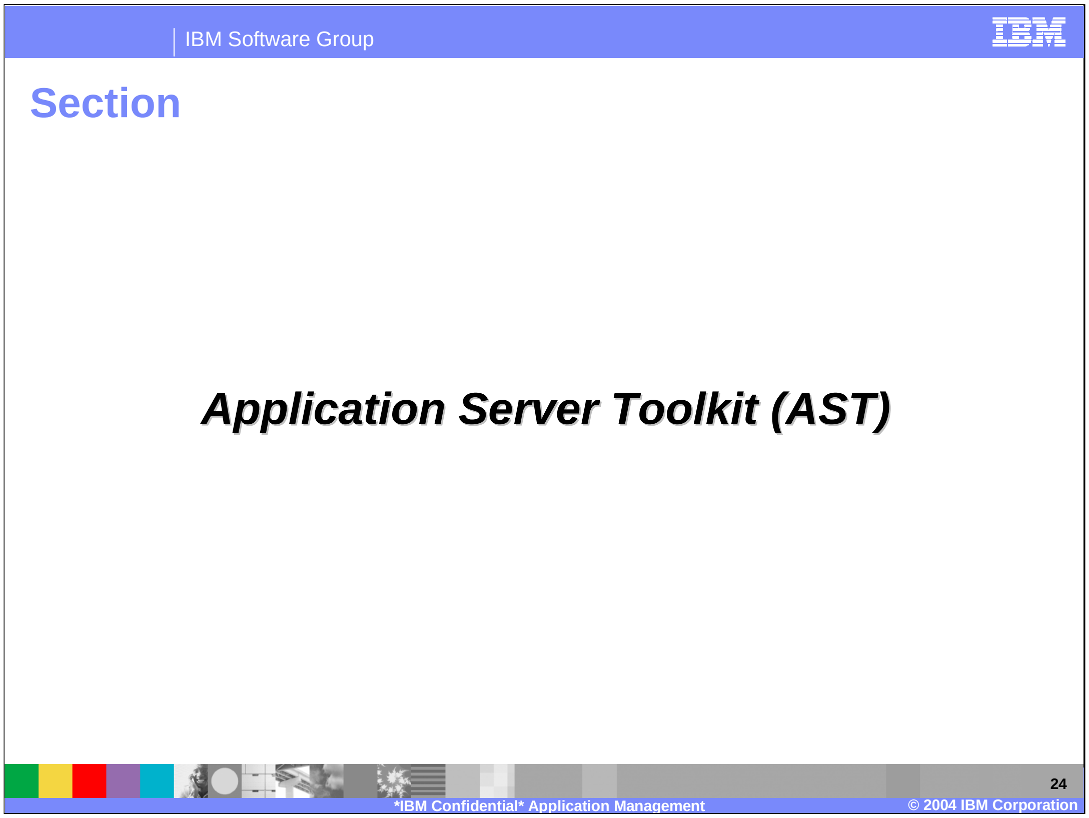
Resources are created under the Application Scope.

Application Scope resources can not be edited in wsadmin or the Admin console.

Support for the Enhanced EAR file is integrated within the Rational Tooling and Application Server Toolkit (AST).

Tooling allows defining of resources and properties on the application, and supports import and export of the Enhanced EAR file.

In WebSphere Application Server v5.x, when you exported an application, you lost any class loader or shared library information. In WebSphere Application Server v6.0, all applications are exported as Enhanced EAR files. So round trip (export, update, import) of an application is much easier, since we now save more of the information.



IBM Software Group

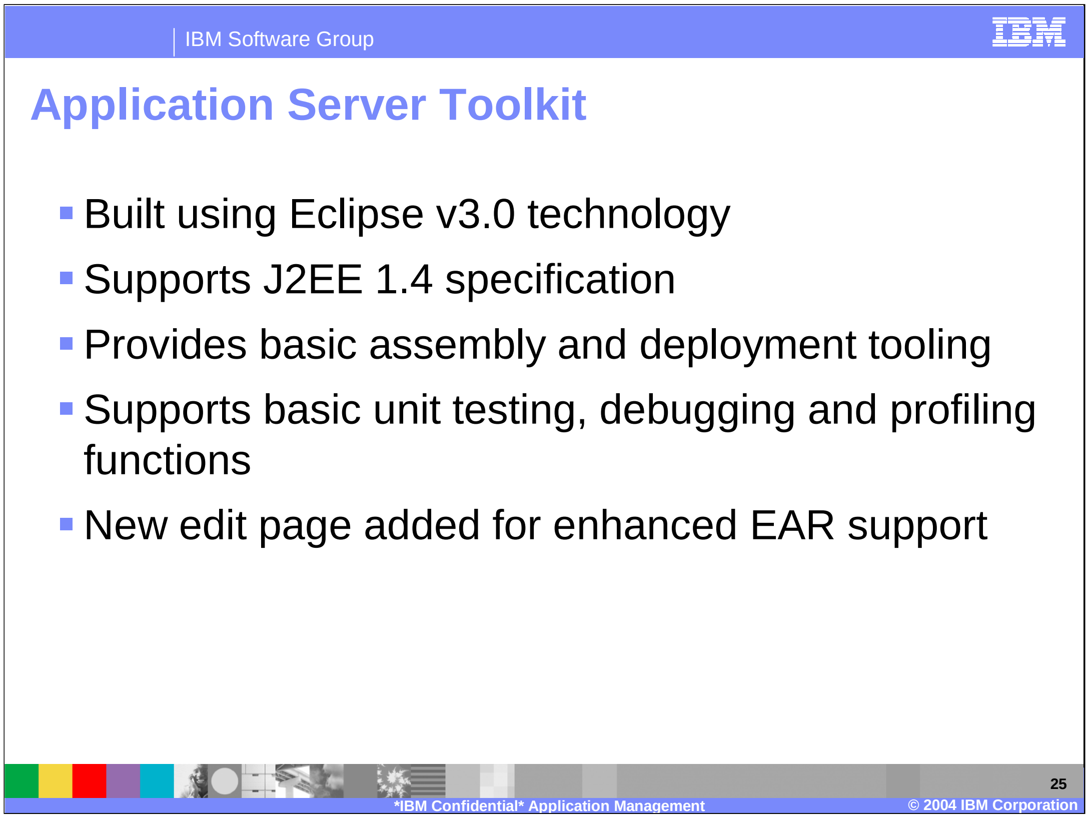
Section

Application Server Toolkit (AST)

24

IBM Confidential Application Management

© 2004 IBM Corporation



Application Server Toolkit

- Built using Eclipse v3.0 technology
- Supports J2EE 1.4 specification
- Provides basic assembly and deployment tooling
- Supports basic unit testing, debugging and profiling functions
- New edit page added for enhanced EAR support

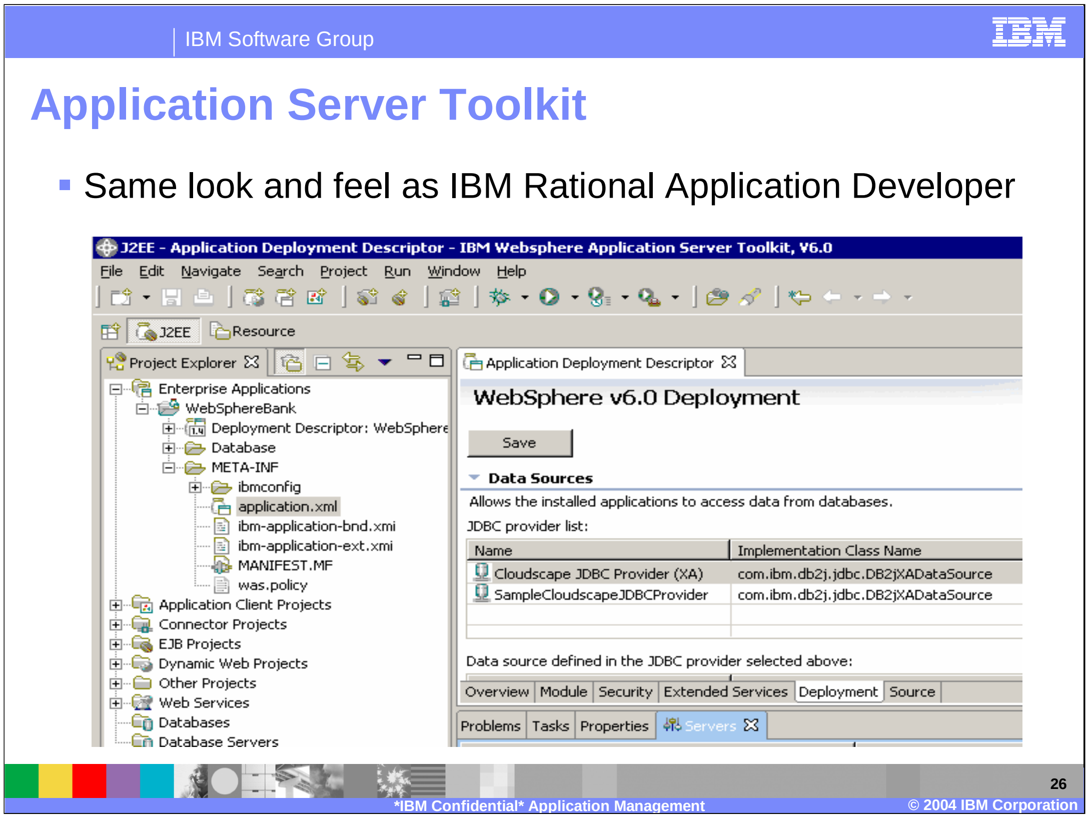
© 2004 IBM Corporation

The Application Server Toolkit is built using Eclipse v3.0 technology.

It supports new features that are defined in the Enterprise JavaBeans™ 2.1 specification and the J2EE 1.4 specification.

It provides basic tooling required for assembly, deployment, testing and debugging.

It also provides tooling necessary for enhanced ear support.



Application Server Toolkit

- Same look and feel as IBM Rational Application Developer

J2EE - Application Deployment Descriptor - IBM Websphere Application Server Toolkit, V6.0
File Edit Navigate Search Project Run Window Help
J2EE
Resource
Project Explorer
Enterprise Applications
WebSphereBank
Deployment Descriptor: WebSphere
Database
META-INF
ibmconfig
application.xml
ibm-application-bnd.xmi
ibm-application-ext.xmi
MANIFEST.MF
was.policy
Application Client Projects
Connector Projects
EJB Projects
Dynamic Web Projects
Other Projects
Web Services
Databases
Database Servers
Application Deployment Descriptor
Save
Overview Module Security Extended Services Deployment Source
Problems Tasks Properties Servers

WebSphere v6.0 Deployment

Data Sources

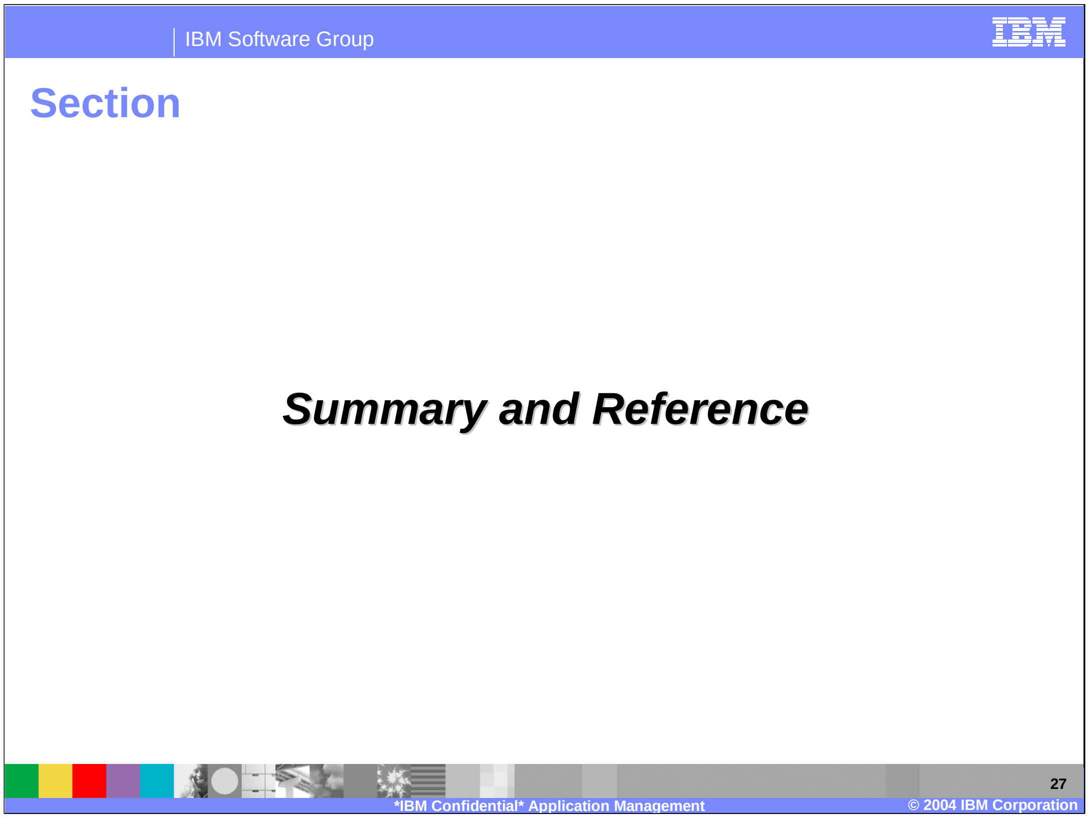
Allows the installed applications to access data from databases.

JDBC provider list:

Name	Implementation Class Name
Cloudscape JDBC Provider (XA)	com.ibm.db2j.jdbc.DB2jXADataSource
SampleCloudscapeJDBCProvider	com.ibm.db2j.jdbc.DB2jXADataSource

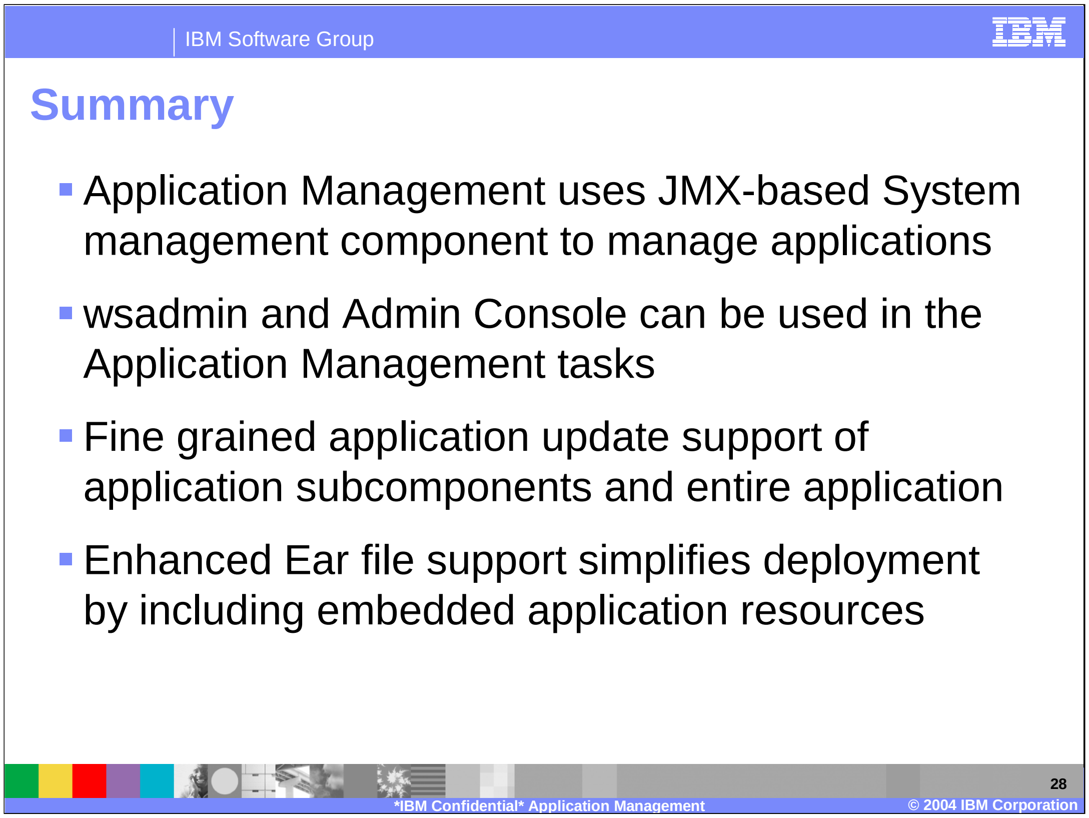
Data source defined in the JDBC provider selected above:

Although the Application Server Toolkit is not an application development tool, it does have the same look and feel as the IBM Rational Application Developer.



Section

Summary and Reference



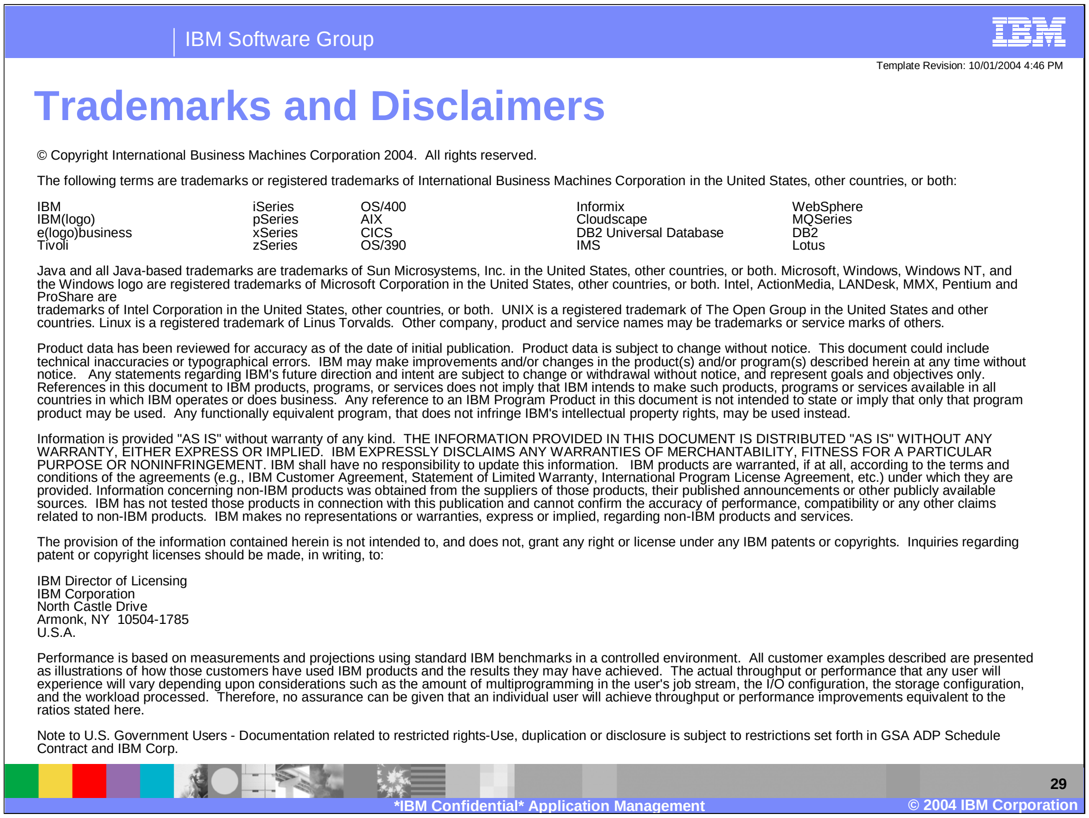
Summary

- Application Management uses JMX-based System management component to manage applications
- wsadmin and Admin Console can be used in the Application Management tasks
- Fine grained application update support of application subcomponents and entire application
- Enhanced Ear file support simplifies deployment by including embedded application resources

In version 6, as in version 5, WebSphere uses JMX based system management to manage applications within WebSphere. You can use the Admin Console, or wsadmin for system management.

Fine Grained Application Update allows updates of application subcomponents as well as updating the entire application.

And Enhanced Ear file support simplifies application deployment by including application resources within an embedded configuration archive.



Template Revision: 10/01/2004 4:46 PM

Trademarks and Disclaimers

© Copyright International Business Machines Corporation 2004. All rights reserved.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM(logo)
e(logo)business
Tivoli
iSeries
pSeries
xSeries
zSeries
OS/400
AIX
CICS
OS/390
Informix
Cloudscape
DB2 Universal Database
IMS
WebSphere
MQSeries
DB2
Lotus

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are
trademarks of Intel Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice.   Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

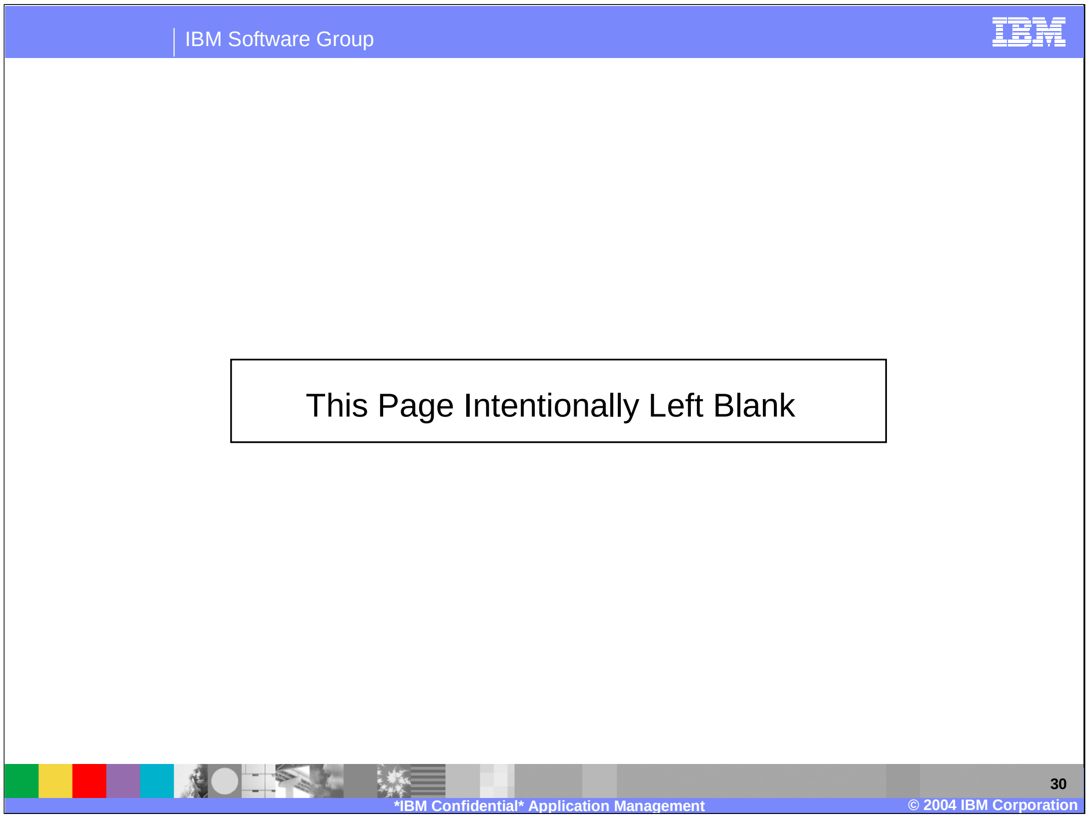
Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information.   IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.



This Page Intentionally Left Blank

© 2004 IBM Corporation