

IBM WebSphere Application Server V4.0

WebSphere in the Enterprise



Topics



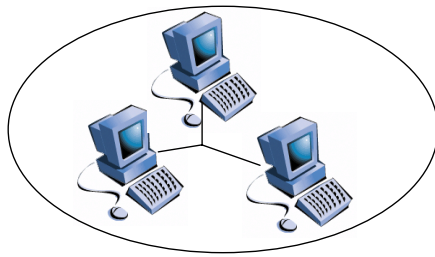
- J2EE Application Life Cycle
- WebSphere Tooling
- Deployment and Installation
- Administration
- WebSphere Version Features



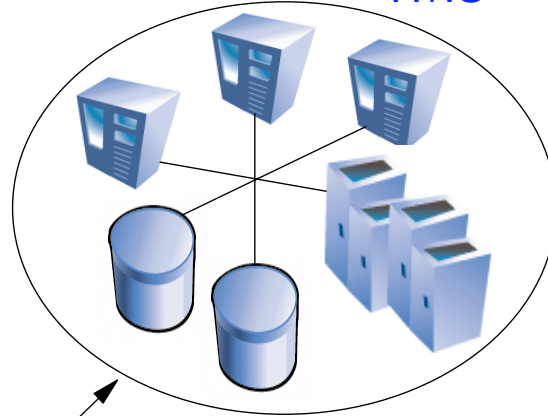
- ▶ With J2EE, notable changes have occurred in the way an application reaches a production server. The changes start with the development tools used to create the JSPs, servlets, and EJBs. Today's development tools as well as the upcoming WebSphere Studio Advanced Developer, provide the function to create fully J2EE compatible applications.
- ▶ Once they are created, the different components are packaged and deployed, then installed into a production environment to be managed and monitored by the System Administrator.
- ▶ Coinciding with the J2EE specification, there are specific tools to aid system administrators in managing the different nodes and applications, as well as tools for monitoring performance of specific modules in WebSphere.
- ▶ As in the past, there are different versions with different features for specific stages of the application life cycle. These different features help speed development time because they provide tools and functionality helpful at that point in the development process.

Ideal J2EE Enterprise

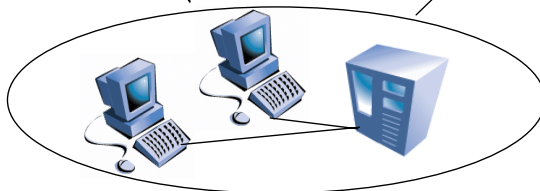
Development



IT/IS



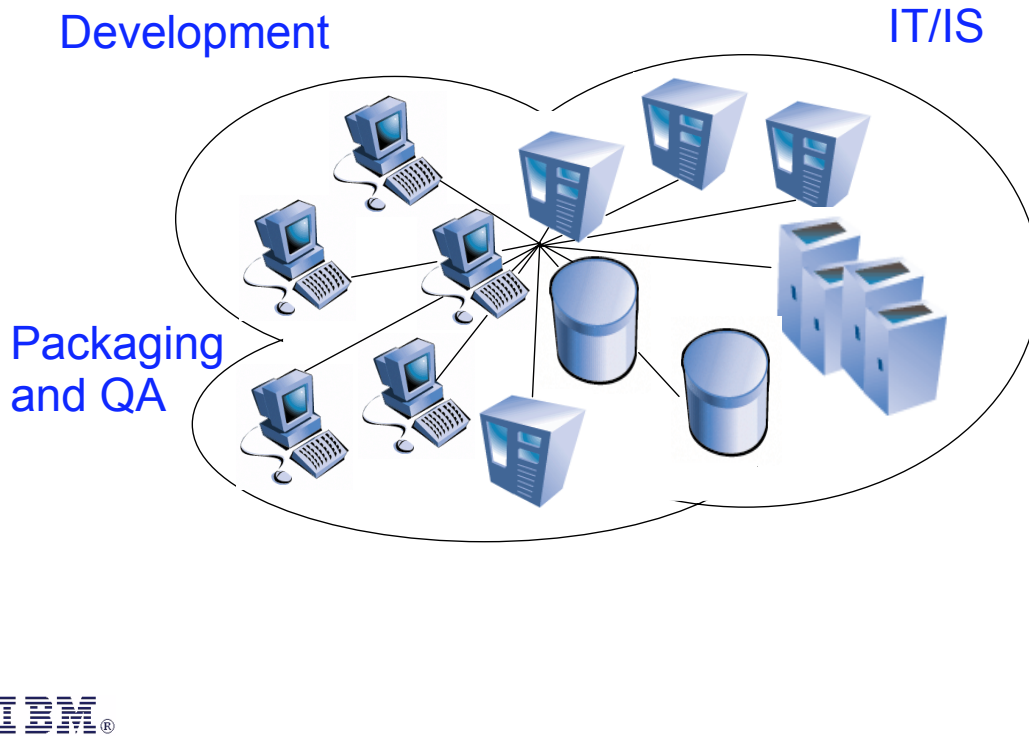
Packaging and QA



IBM®

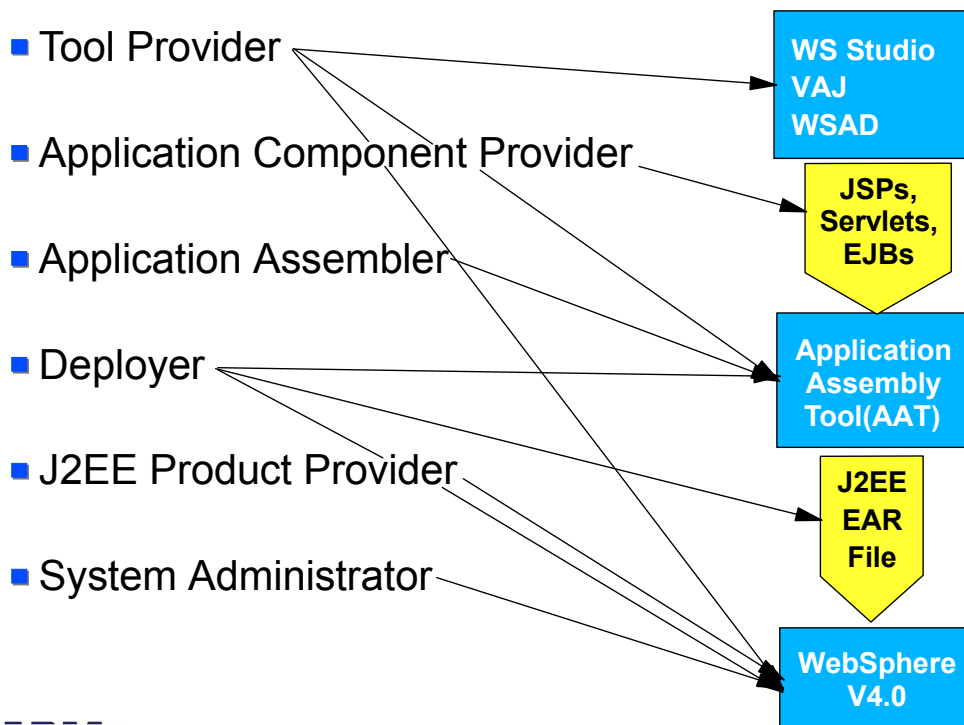
- ▶ In an ideal enterprise, the application life cycle can be separated into three groups, Development, Packaging and QA (testing), and Information Technology/Information Systems (IT/IS).
- ▶ Each group has separate tasks and responsibilities, virtually independent of the other groups. With the separation of roles, each group can focus at the job at hand and not be distracted by problems they are not trained or ready to solve.
- ▶ Development can focus on the design and creation of the application. Packaging and QA focuses on combining the different components created by Development into a unit that can be installed easily into production, reducing the bring up time.
- ▶ Finally IT/IS is responsible for overseeing the daily operation of the application, insuring that performance is acceptable and the application is available.

Typical Enterprise (Pre J2EE)



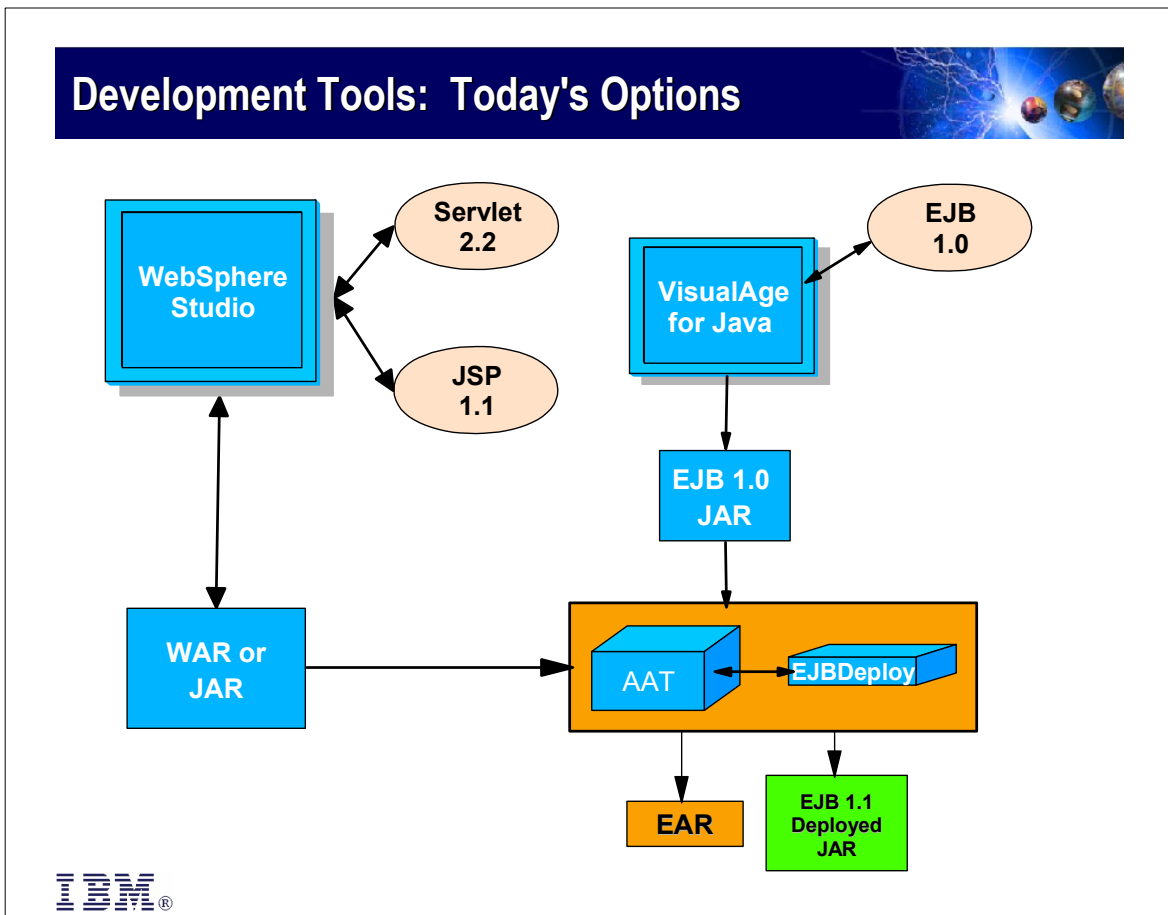
- ▶ Unfortunately, groups in an enterprise having specific roles and responsibilities defined seldom happens. Many times Development, while very busy programming, is responsible for packaging and testing the application throughout the development process. Development can also be tasked to resolve implementation problems and is forced to spend time setting up the application and troubleshooting configuration errors.
- ▶ Many times changes from Development skip Packaging and QA and go straight into the production environment. Skipping testing and packaging can cause more down time if errors occur in the production environment. More time is spent with configuration issues. Worse yet, Packaging and QA may take place in the production environment, severely hindering performance and availability.
- ▶ IT/IS tends to spend too much time working with development on these installation and configuration problems, neglecting the daily operations of the servers and network. Performance problems and availability can quickly become a problem if not carefully monitored.

J2EE Application Life Cycle



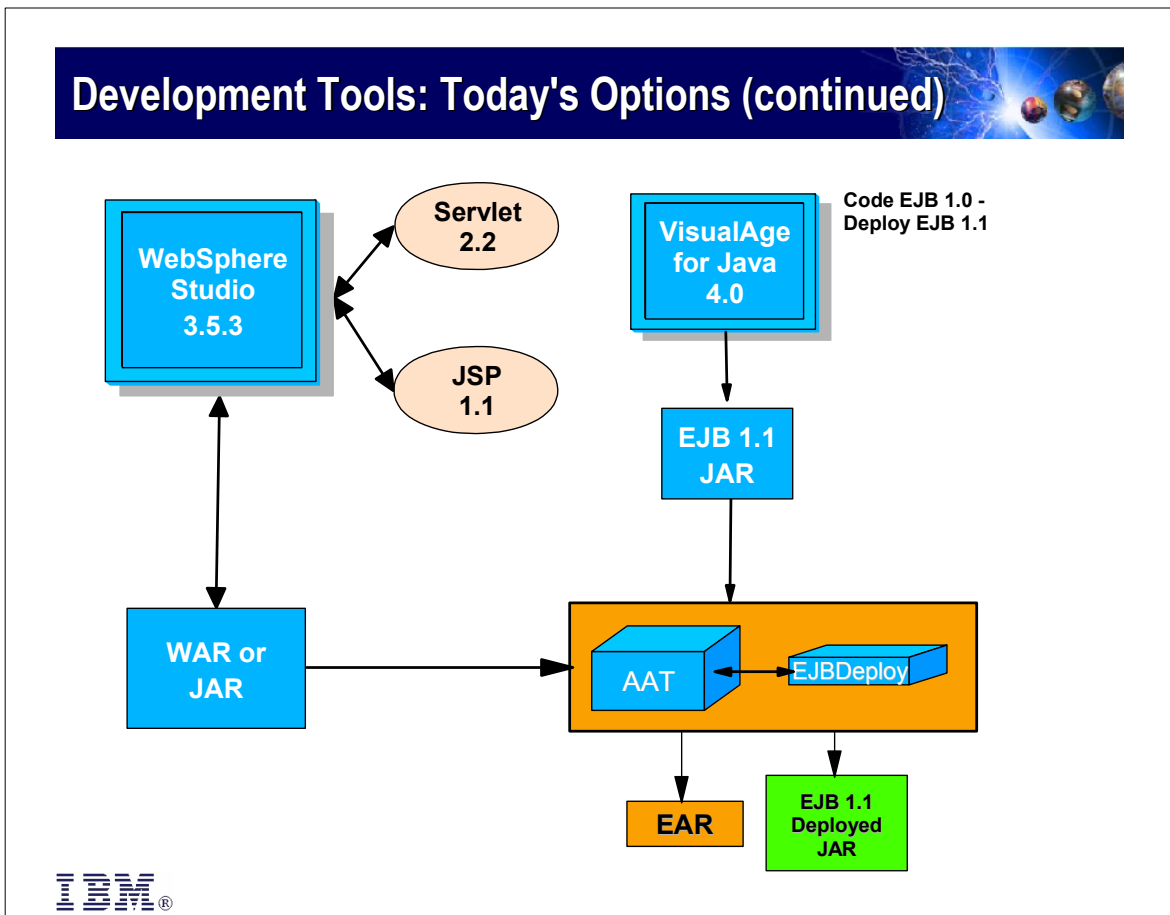
- ▶ Separation of roles, part of the J2EE specifications, brings the typical enterprise closer to an ideal state where certain people have specific tasks and responsibilities in the application life cycle.
- ▶ Within WebSphere V4.0 there are specific tools, utilities, and applications to match each of the six J2EE roles.
- ▶ Starting with the Tool Provider Role, IBM provides those tools, such as Visual Age for Java and WebSphere Studio, for creating the JSPs, servlets, and EJBs used in the application. Tools, such as the Application Assembly Tool (AAT), are provided within WebSphere for packaging components into an installable EAR file. Tools are also provided for managing and monitoring the installed applications.
- ▶ The Application Component Provider is the developer who uses the tools available from the Tools Provider. Using the development tools, the Component Provider passes components such as JSPs, servlets, and EJBs to the Application Assembler for packaging.
- ▶ The Application Assembler uses AAT to form an Enterprise Archive or EAR file. The Application Assembler is also responsible for describing external dependencies of the application the Deployer must resolve in the deployment process.
- ▶ The Deployer is responsible for deploying Web applications and EJB components into a specific operating environment. The first of three steps the Deployer performs is installation of the different components and classes into the J2EE containers. In the next step, the Deployer resolves all external dependencies provided by the Application Component Provider and follows the assembly instructions defined by the Application Assembler. These instructions may include items such as adding the security roles to specific parts of the application. The last step is to actually start the application and verify it in an actual J2EE environment provided by the J2EE Product Provider.
- ▶ The J2EE Product Provider is IBM WebSphere. All versions of WebSphere V4.0 are J2EE compliant, and support all the functionality of a J2EE application.
- ▶ With the application installed into the J2EE product, the System Administrator can take over the daily management of the application. Specific tools, provided by the Tools provider, are used for managing and monitoring the application.

Development Tools: Today's Options



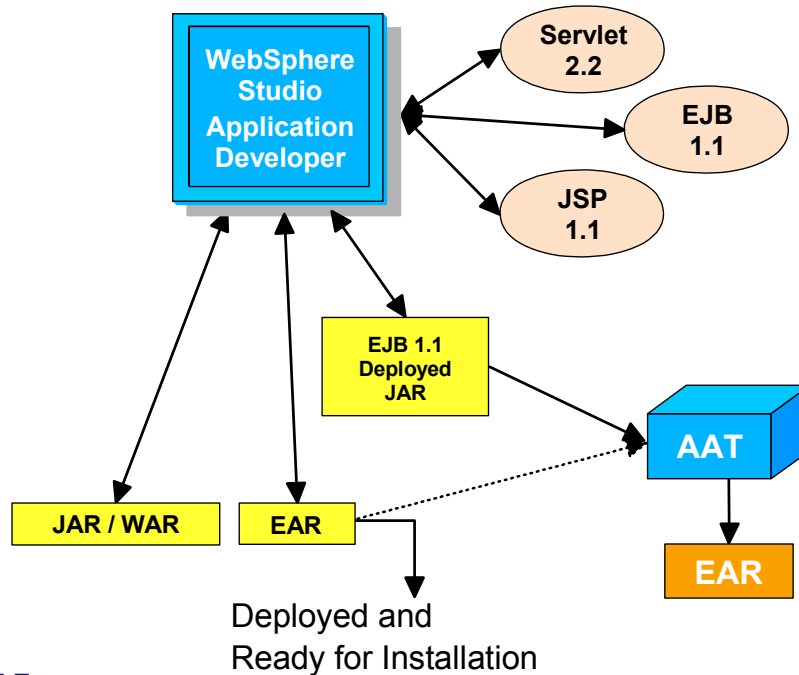
- ▶ Lets look at the different tools provided by IBM to create the components that make up typical J2EE applications as well as how AAT packages the components.
- ▶ Today we have two development tools, WebSphere Studio and Visual Age for Java. WebSphere Studio is an integrated toolset that provides wizards to help generate web application templates such as HTML, 1.1 JSPs, 2.2 Servlets.
- ▶ Visual Age for Java is a Java IDE. You can develop Enterprise Java Beans using Visual Age for Java. Although team development with EJBs is greatly improved, Visual age for Java V3.5.3 misses support for EJB1.1. You can use the EJB Development Environment in VisualAge for Java to create 1.0 EJBs, but you must use Application Assembly Tool to create the EJB 1.1 deployed JAR. In the past, the Class Editor in Visual Age for Java could be used to manually make changes to add 1.1 EJB method calls. Today with WebSphere 4.0, the AAT is the preferred method for converting to EJB1.1
- ▶ With the different parts of the J2EE application created and ready for packaging, the Application Assembler uses AAT, which implements the EJBDeploy tool, to convert the EJB1.0 deployment descriptors to an XML deployment descriptor and make it a true EJB 1.1 jar. The Application Assembler, using AAT, will then assemble the JAR and WAR files to create an Enterprise Application EAR file.

Development Tools: Today's Options (continued)



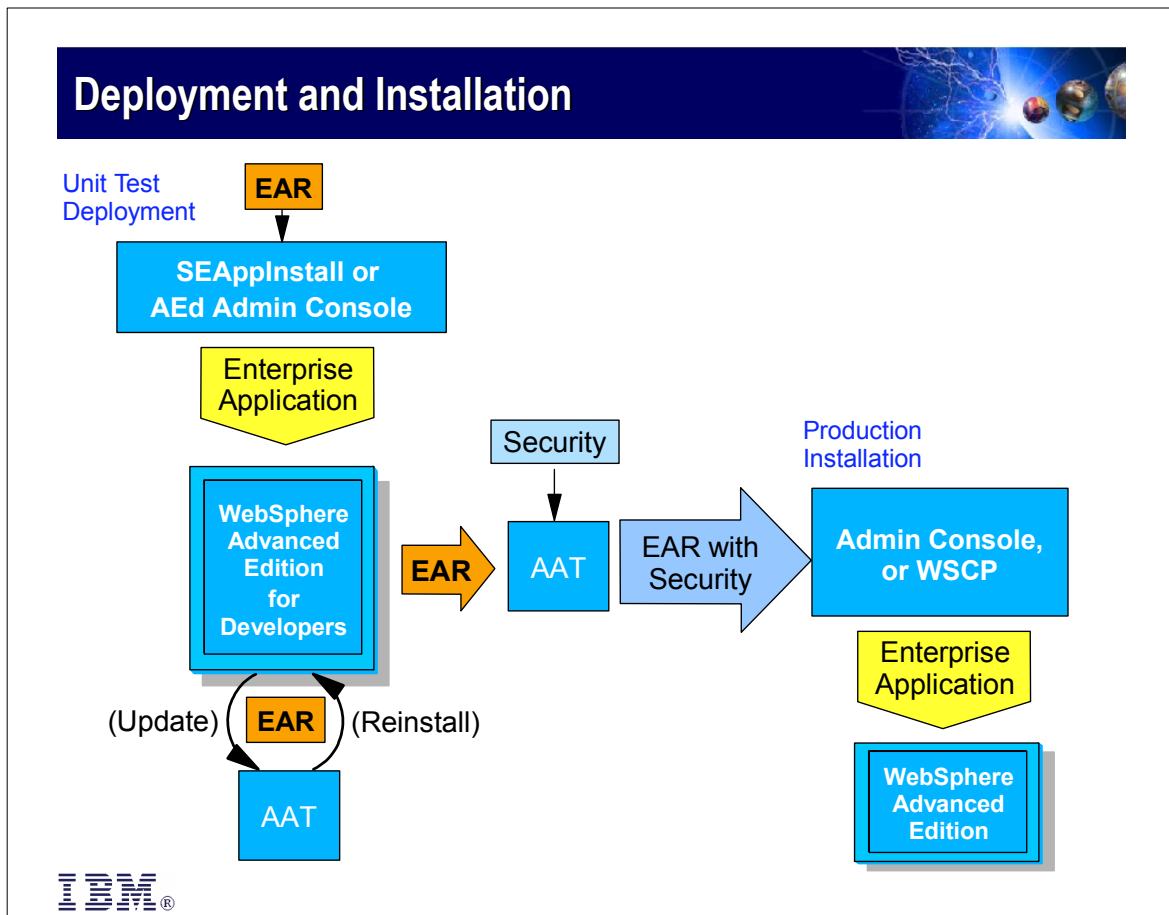
- ▶ This slide is different from the previous slide in that, here the Application Component Provider is using Visual Age for Java 4.0 instead of Visual Age for Java 3.5.3. Visual Age for Java 4.0 exploits Servlet2.2 and JSP1.1 and allows for deployment to EJB1.1.
- ▶ With Visual Age for Java 4.0, the Application Component Provider will code to EJB1.0, However the Application Component Provider can deploy to EJB1.0 or EJB 1.1. This is possible because Visual Age for Java 4.0 supports the XML based deployment descriptors of EJB1.1. The EJB1.1 jars and the other modules can then be assembled by the Application Assembler, using the Application Assembly Tool, to create an Enterprise Application EAR file.

Development Process - Upcoming Tooling



- ▶ 2001 is the year IBM initiates the journey to take Web Application Development to the next level. IBM is moving from the existing development tools to a new development platform providing significantly increased team development productivity and integration so that developers can work together more easily.
- ▶ The diagram shows the future directions for WebSphere Foundation Tooling or WebSphere Studio Application Developer.
- ▶ The Application Component Provider can develop J2EE compliant applications using Servlet 2.2, JSP 1.1, and EJB 1.1.
- ▶ The Application Component Provider will be able to create JAR and WAR files, and combine individual modules into EAR files. The Application Component Provider can also create EJB1.1 deployed JAR files.
- ▶ The EAR and deployed JARs created using WebSphere Studio Application Developer can also be imported into Application Assembly Tool by the Application Assembler.

Deployment and Installation



- ▶ With the Enterprise Archive or EAR file deployed, the Deployer can install and test the application in WebSphere Advanced Developer Edition (AEd).
- ▶ WebSphere Advanced Developer Edition is the recommended development and testing platform. AEd provides all of the features needed for J2EE applications, allowing developers, assemblers, and deployers to concentrate more on the application components and less on the implementation.
- ▶ The Deployer will use SEAppInstall or the AEd Admin Console to install the Enterprise Application into AEd.
- ▶ During testing, the Deployer may need to update external references or other dependencies. The Deployer must use the AAT tool to make these updates and then reinstall the application using SEAppInstall or the AEd console.
- ▶ After testing is complete, the Deployer uses the Application Assembly Tool to add security to the Enterprise Archive or EAR file.
- ▶ With security fully supplied, the Enterprise Application is ready to be installed into production in WebSphere Application Server, Advanced Edition (AE) using WSCP or the Admin Console.
- ▶ Note: The command line tool SEMigrator can be used to automate movement from AEd to AE. More information available at a later date.

Deployment and Installation Considerations

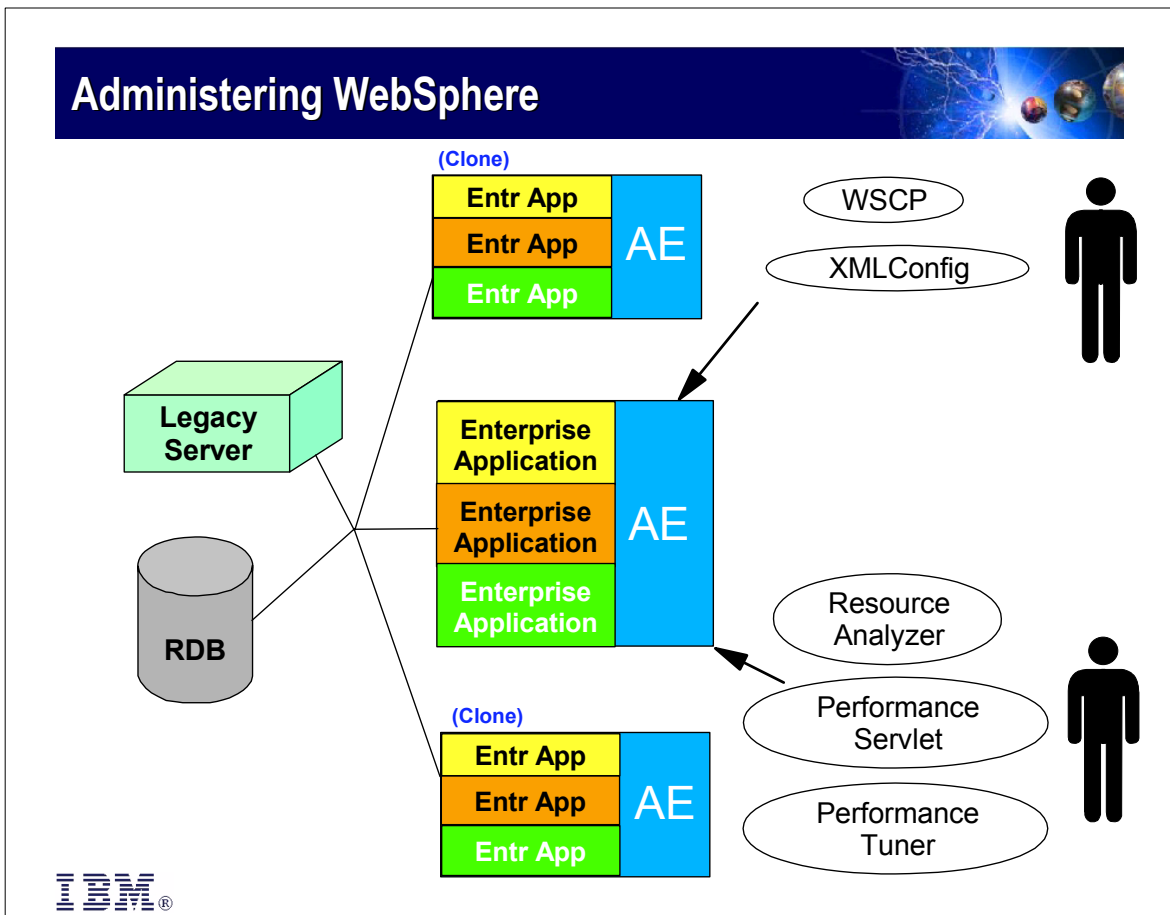


- Application Assembly Tool (AAT) is the primary tool for making changes to applications
- Admin Console is for management
- With J2EE, settings are stored in XML files, the Admin Repository, or in both
- Manual changes to XML files and console settings of installed applications quickly make EAR file "out of date"
- AAT can read the installed apps directory, but changes must be saved to another file
- EARExpander utility can collapse an installed app directory into an EAR file
- More features to follow in Best Practice documentation



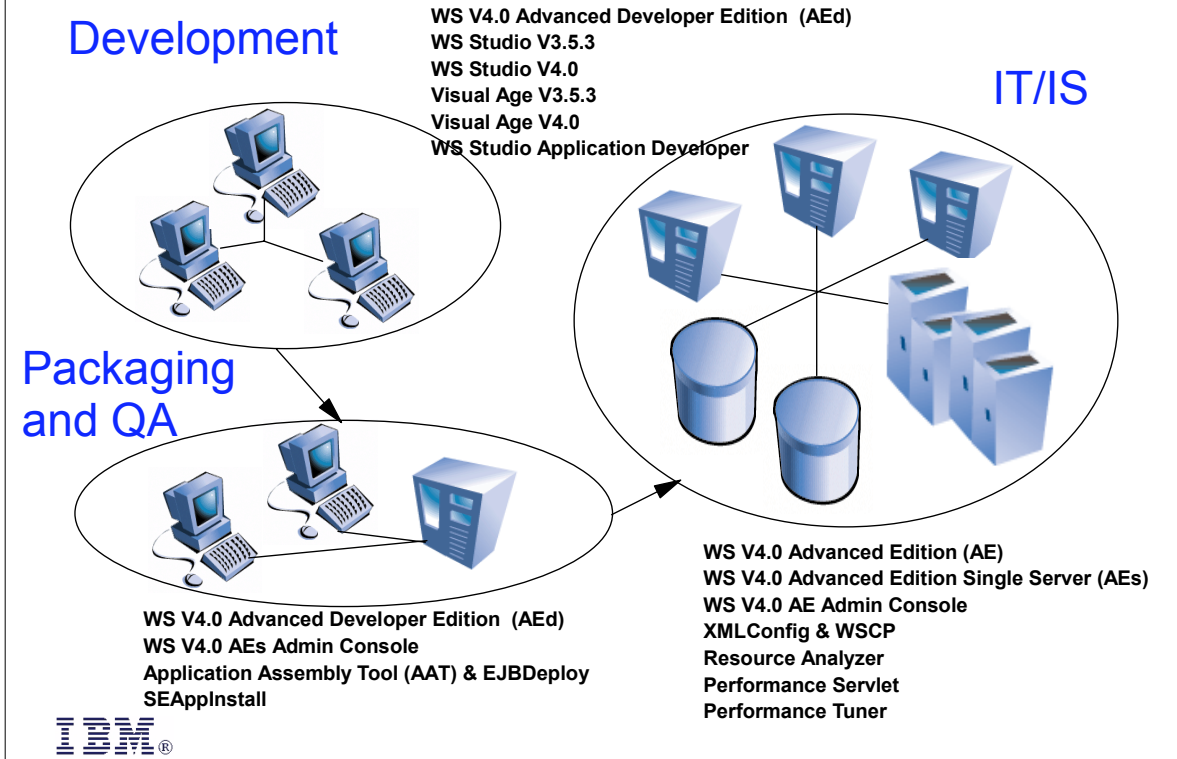
- ▶ There are a number of items for the Deployer to remember when moving an application from WS Advanced Developer Edition to WS Advanced Edition. Most of these items have to do with the tooling behind the separation of roles listed in the J2EE specification.
- ▶ AAT is the primary tool used for making changes to an application. The Admin console should be considered a management tool. This implementation emphasizes the separate J2EE roles of Deployer and System Administrator. The Deployer should perform testing before installing into a production environment.
- ▶ Once an application is installed into production, managing and monitoring are the main actions performed on the application under the J2EE specification.
- ▶ Changes to the servlet class files and helper classes can be done by replacing individual JAR files. This will however cause the EAR file and the installed application to be out of synch.
- ▶ Changes on the security of methods, environment variables for EJBs, and init parameters of EJBs or servlets in an installed applications in production will require reinstallation of the Enterprise Application.
- ▶ Changes to specific URI names for servlets in applications installed into production will require a reinstall or a manual update of the XML deployment descriptors and extension files, which will also cause the EAR file and the installed application to be out of synch.
- ▶ Best Practice documentation on updating multiple nodes in production is being developed.

Administering WebSphere



- ▶ With the application installed by the Deployer, the System Administrator can focus on the daily operations of the applications and the servers. In WebSphere V4.0, the Tool Provider has created a number of tools for managing and monitoring the installed J2EE applications.
- ▶ For managing the different nodes and Enterprise Applications on a node, WSCP and XMLConfig can be used to automate tasks such as starting and stopping the application servers.
- ▶ For monitoring of different parts of a domain and the different modules, the Resource Analyzer, which uses the PMI interface, can be used to watch for high utilization or bottlenecks. The Performance Servlet is another tool shipped with WebSphere that provides an HTTP/XML interface allowing for easy access to performance data for third party tools. Finally, there is the Performance Tuner, a tool available in the Admin Console that provides a wizard interface for changing specific performance setting found over a node.

WebSphere V4.0 in the Enterprise



- ▶ In summary, with the separation of roles and the tools corresponding to the roles, we can see where the different WebSphere tools and applications fit into the Ideal J2EE Enterprise.
- ▶ The development tools provided by WebSphere allow developers to focus on creation of JSPs, servlets, and EJBs.
- ▶ AAT facilitates application packaging into easy-to-install EAR files. Testing of the installed package can also be performed using WebSphere Advanced Developer Edition outside of the production environment.
- ▶ With the J2EE-defined roles, IT/IS can focus on the daily operation of installed J2EE applications. Tools are available for managing and monitoring the different modules and nodes.

AEd/AEs vs. AE



- **Advanced Developer Edition (AEd) and Advanced Single Server Edition (AEs)**
 - ▶ Differences
 - AEd contains the same features (code) as AEs - License restricted
 - AEd designed for development, testing, and troubleshooting
 - AEs single process production server
 - ▶ Major distinguishing features from AE
 - XML Admin Repository
 - Web-based Console
 - Local OS Security
- **Advanced Edition (AE)**
 - ▶ WLM Support
 - ▶ Clustering/Cloning
 - ▶ Administration of multiple nodes and processes
 - ▶ Administrative Tools (XMLConfig, WSCP)
 - ▶ Database Admin Repository
 - ▶ Distributed Security (LTPA, LDAP)



- ▶ As mentioned earlier, WebSphere Advanced Developer Edition (AEd) is the recommended development and unit test platform. WS AEd shares the same code base as WS Advanced Single Server Edition (AEs). The only difference is the license and price.
- ▶ This small difference provides confidence to Assemblers and Deployers that their applications written in the development environment will function as expected in a production environment.
- ▶ One of the main differentiating features of AEd/AEs is that it offers administration of a single node and application server.
- ▶ AEs/AEd has the ability to use local OS Security within an installed application.

- ▶ Advanced Edition (AE) is a robust production solution offering Work Load Management (WLM) features such as clustering and cloning on across multiple processes and nodes.
- ▶ AE also provides Administrative tools such as XMLConfig and WSCP for managing nodes across the enterprise.
- ▶ AE stores the repository in a database, allowing for easy backup and centralized management.
- ▶ With AE, there is also the added feature of using distributed Security such as Lightweight Third-Party Authentication (LTPA) or Lightweight Directory Authentication Protocol (LDAP).

Advanced Edition Feature Comparison



Feature	AE for Development (AEd)	AE Single Server (AEs)	AE Production
Production certified	No	Yes	Yes
Browser-based administration	Yes	Yes	No
Multi-node administration	No	No	Yes
Database backed administration	No	No	Yes
Distributed security	No	No	Yes
Workload management	No	No	Yes
Clustering / cloning	No	No	Yes
Directory services	No	No	Yes
Enterprise Edition Integration	No	No	Yes



- ▶ Here are some of the detailed differences between the three editions of WebSphere Version 4.0.
 - ▶ Only the Single Server AEs and Advanced Edition AE are production certified.
 - ▶ WebSphere Application Server, Advanced Developer Edition (AEd) is limited by license for development use.
 - ▶ Both AEs and AEd use a lightweight Administrative Console that runs in a Web browser.
 - ▶ The Advanced Edition (AE) Console runs in a standalone Java client application.
- ▶ The rest of the features are available only in Advanced Edition.
 - ▶ These features make AE a more robust, scalable, and highly available production environment that can span multiple machines.

Summary



- J2EE suggests development paradigm
- WebSphere tooling maps to J2EE Platform Roles
- Administrative tools enhance Administrative experience
- Edition Differences map (roughly) to
 - ▶ Development
 - ▶ Test
 - ▶ Production

