



IBM Software Group

SW5706 Application deployment problems



© 2007 IBM Corporation

This presentation will focus on application deployment problem determination on WebSphere Application Server V6.

Unit objectives

After completing this unit, you should be able to:

- Detect an application deployment issue in WebSphere Application Server V6
- Determine the root cause of an application deployment problem by using the WebSphere Application Server V6 log files
- Resolve the application deployment issue by using the administrative console and the *Application Server Toolkit (AST)*

The goal of this presentation is to show methods of detecting, determining root cause, and resolving application deployment issues.

- An application can be installed into WebSphere Application Server V6 using two primary methods:
 - ▶ WebSphere Application Server V6 administrative console installation wizard
 - **Applications -> Install New Application** or **Applications -> Enterprise Applications -> Install**
 - ▶ wsadmin script interpreter (-> \$ wsadmin)
 - In a large enterprise, the wsadmin script interpreter might be the preferred method of application installation because it lends itself to automation
- During the application deployment process, problems could be encountered during any of the following activities:
 - ▶ Installing, deploying, or updating applications
 - ▶ Starting the application for the first time
 - ▶ Running the application for the first time

Applications may be installed into WebSphere Application Server V6 using two primary methods: using the administrative console installation wizard, or the wsadmin script interpreter. During the application deployment process, problems may be encountered during any of the following activities: installing, deploying, or updating the application; starting the application for the first time; or running the application for the first time. Such issues can manifest themselves as warnings or errors during the deployment process, or as failed requests when the application is tested.

Detecting an application deployment issue

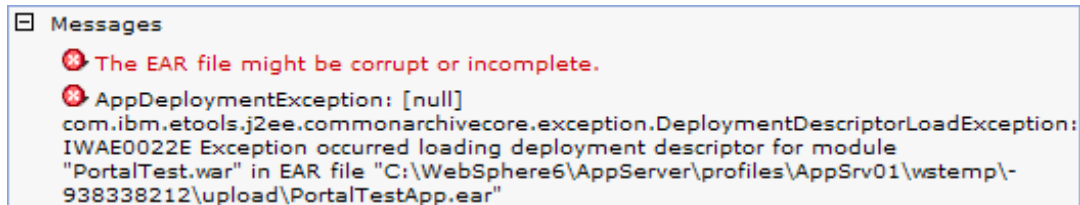
- Problem detection during installation is simple: The installation process fails.
 - ▶ An error message will be written to the administrative console if the failure occurs while using the administrative console installation wizard or to the command console if installation fails while using the wsadmin script interpreter
- Detecting an application start failure will occur in one of two ways:
 - ▶ In the administrative console, the application status could stay stopped, or change to unavailable
 - ▶ An error message will be written to one or more log files
- When running the application for the first time, any sort of transaction or request failure, such as missing content 404 Page Not Found, or other protocol errors will result.
 - ▶ If the application client does not provide visible feedback then detection of failure might not occur until the application client log files or status is investigated

Problem detection during application installation is simple - the installation process fails. An error message will be written to the administrative console if the failure occurs while using the administrative console installation wizard, or to the command console if installation fails while using the wsadmin script interpreter. Application start failures will manifest themselves as an application status of "stopped" or "unavailable" in the administrative console, or as an error message written to one or more log files.

When running the application for the first time, request errors, "page not found" errors, or other protocol errors may be presented, or, in lieu of visible feedback, application client log files or status may need to be investigated.

Examining console and error log file messages

- During the installation process, any exception that occurs will be logged to the WebSphere Application Server V6 SystemErr.log file
- If you are using the WebSphere Application Server V6 application installation wizard, the errors will be displayed to the administrative console
- For example, the following error message is a result of a corrupted or incorrect application WAR file:



```
Messages
* The EAR file might be corrupt or incomplete.
* AppDeploymentException: [null]
com.ibm.etools.j2ee.commonarchivecore.exception.DeploymentDescriptorLoadException:
IWAE0022E Exception occurred loading deployment descriptor for module
"PortalTest.war" in EAR file "C:\WebSphere6\AppServer\profiles\AppSrv01\wstemp\
938338212\upload\PortalTestApp.ear"
```

Application deployment problems, like all WebSphere Application Server V6 error, warning, and information messages, will be logged to the system logs based on the level of trace specified for the system. By default, any error that occurs will be captured in the SystemErr.log or the SystemOut.log. If you are using the administrative console, a message will be rendered to the panel that is currently active as soon as the message is generated. The *Troubleshooting* section of the administrative console navigation menu contains links that guide you to the messages that can occur as a result of configuration or runtime activities. If you are using the *wsadmin* script interpreter to install an application, then messages will be streamed to the command console as soon as they are generated. Receiving an error or warning message is an indication that the log files should be inspected for further information about the event that just occurred. Some application deployment problems are simple syntax problems. For example, invalid characters in the name of the application can lead to an error during installation. Check to verify that you are not using any special characters for your application name. Another common problem is user error. The application installation might complete correctly, using the *wsadmin* script interpreter, but when you look for your application in the administrative console, it does not show up. This is a result of not having run the *save* command after the installation process. As more experience is gained with the WebSphere Application Server V6 installation process, user error occurs less frequently.

Administrative console Troubleshooting menu

Runtime Events

Runtime events propagating from the server

Preferences

Timestamp	Message Originator	Message
Apr 4, 2006 8:42:35 AM CDT	com.ibm.ws.security.registry.nt.NTLocalDomainRegistryImpl	SECJ0340E: Could not get the uniqueid for the group
Apr 4, 2006 8:42:35 AM CDT	com.ibm.ws.security.registry.nt.NTLocalDomainRegistryImpl	SECJ0350E: Could not get the uniqueid of the user
Apr 4, 2006 1:59:18 PM CDT	com.ibm.ws.performance.tuning.serverAlert.calc.DataAccessWrapper	Advisor caught exception. Advisor was unable to q
Apr 4, 2006 1:58:38 PM CDT	com.ibm.ws.performance.tuning.serverAlert.calc.DataAccessWrapper	Advisor caught exception. Advisor was unable to q
Apr 4, 2006 1:50:23 PM CDT	com.ibm.ws.performance.tuning.serverAlert.calc.DataAccessWrapper	Advisor caught exception. Advisor was unable to q
Apr 4, 2006 1:46:31 PM CDT	com.ibm.ws.webcontainer.WebContainer	SRVE0017W: A WebGroup/Virtual Host to handle /Xsea
Apr 3, 2006	com.ibm.ws.security.role.RoleBasedAuthorizerImpl	SECJ0306E: No

Application deployment problems © 2007 IBM Corporation

Located under the "Troubleshooting" menu in the administrative console, the Configuration Validation and Runtime Messages functions are excellent places to detect if an error has occurred. The messages are time sorted and grouped by severity and provide immediate feedback regarding the state of the system after performing an application installation. Both the Configuration Validation messages and the Runtime Messages are grouped by Error, Warning, and Information types. Runtime messages are formatted across columns containing the timestamp, message originator and the actual message. The message is a hyperlink leading to additional detailed information.

Typical application installation errors

- While installing, deploying, or updating the application, the following errors might be encountered:
 - ▶ *"Timeout!!!"* error when attempting to install an enterprise application in the administrative console
 - ▶ *NameNotFoundException* message when deploying an application that contains an EJB module
 - ▶ *SRVE0026E: [Servlet Error]-[Unable to compile class for JSP file*
 - ▶ *WASX7015E* error when running wsadmin command "\$AdminApp installInteractive" or "\$AdminApp install"
 - ▶ *ADMA0004E: Validation error in task* when specifying the default data source for EJB modules during application installation using the administrative console or the wsadmin tool
 - ▶ *No valid target is specified in ObjectName anObject for module module_name* when installing an application

Typical application installation errors include timeouts, inability to precompile JSPs, and NameNotFoundExceptions. These errors are runtime exceptions based on the failure of the installation tool as it attempts to install the application, not on the runtime of the application itself. So while the symptom being reported is an application failure, the cause is based on something either missing from the application or incorrect with the deployment environment. Some examples of environmental problems are incorrect permissions, lack of connectivity, or an incorrectly specified path or other variable. Any exception that occurs will be logged to the SystemErr.log file.

Typical application start-up errors

- The installation process could be successful but the first time attempt to start the application could result in failure.
- Some of the errors that might occur when attempting to start the application for the first time could be:
 - ▶ *java.lang.ClassNotFoundException*: classname Bean_AdderServiceHome_04f0e027Bean
 - ▶ *ConnectionFactory E J2CA0102E*: Invalid EJB component: Cannot use an EJB module with version 1.1 using The Relational Resource Adapter
 - ▶ *NMSV0605E*: A Reference object looked up from the context... was sent to the JNDI Naming Manager and an exception resulted.
 - ▶ Deployed application that uses Java Native Interface (JNI) code does not start
 - ▶ Other name server ("NMSV...") errors

After an application is installed for the first time its initial state is *stopped*. An explicit start is required in order to bring the application on line.

An error during the application start operation is of the *configuration* type. What this means is that the actual installation and deployment was satisfied but now that the runtime bindings are being exercised for the first time, some reference is not satisfied. You can see this by the types of errors that are detected such as Class loader exceptions, version problems, naming reference issues, and so on.

Typical application first run errors

- Running the application for the first time means that you have installed, deployed, and started it successfully – the application dependencies are satisfied!
- However, running the first transaction through the application could result in further issues, such as:
 - ▶ *javax.naming.NameNotFoundException*: Name name not found in context "local"
 - ▶ *SRVE0026E*: [Servlet Error]-[Unable to compile class for JSP error on JSP]
 - ▶ Message similar to "Message: /jspname.jsp(9,0) Include: Mandatory attribute page missing" displays when trying to access JSP file
 - ▶ *org.omg.CORBA.BAD_PARAM*: Servant is not of the expected type. minor code: 4942F21E completed: No
 - ▶ A "Page cannot be displayed... server not found or DNS error" error displays when attempting to browse a JavaServer Pages (JSP) file using Internet Explorer

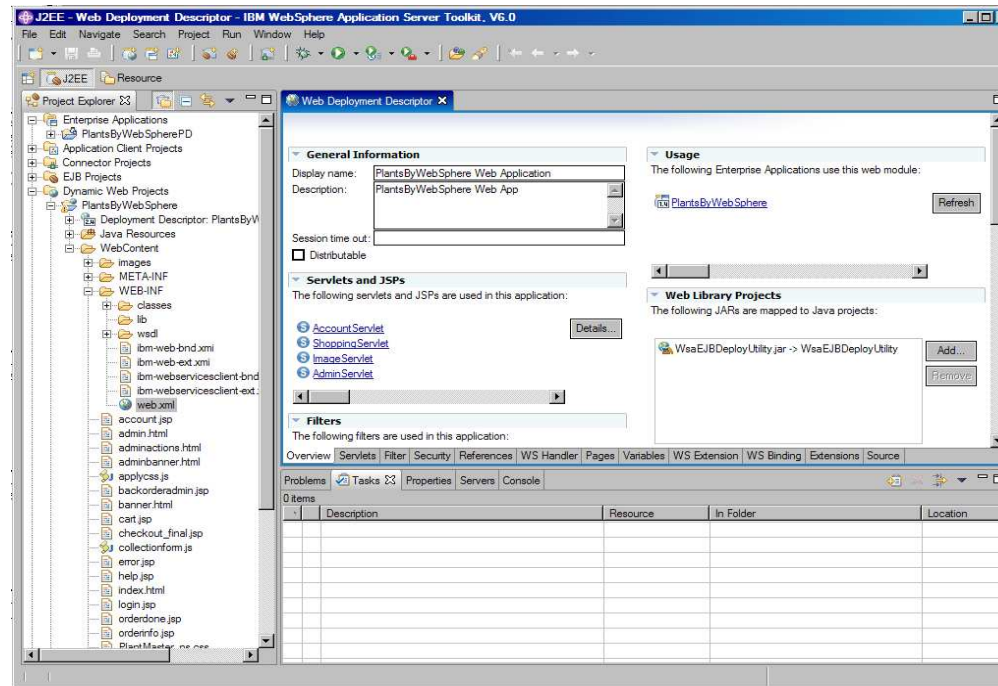
Errors which arise during the first requests to an application are of the environmental type. The application might be integral, but the environment in which it is operating has unsatisfied references which cause the transaction to fail. The actual symptoms of an environmental problem might be obvious, such as missing content or an error page returned to the browser. However, transaction failure can be more subtle. Then there are the various browser differences that have to be considered. Depending on the client accessing the content, the type of client being used could have a significant impact on what is considered correct operation.

Troubleshooting tools and techniques

- The WebSphere Application Server Toolkit (AST) provides basic assembly and deployment tools for publishing to WebSphere Application Server V6.0
- WebSphere Application Server V6 contains embedded tools that can be useful for debugging application deployment problems
 - ▶ The Class Loader Viewer that began shipping with the product in version 6.0.2 is excellent for resolving class loading issues
- The WebSphere Application Server V6 deployment environment might require changes in order to have applications successfully deploy and run
 - ▶ Such changes could include defining or re-defining system variables such as PATH or CLASSPATH

Several tools are available for resolving application installation issues. The *Application Server Toolkit* (AST) is an Eclipse based tool that provides a rich integrated development interface (IDE) for developing and deploying J2EE applications. Making changes to the application can be accomplished using the AST or one of the other Eclipse based IDEs from IBM. If there is a JSP compilation problem or a configuration change, then the AST provides facilities for making changes to the code and the application deployment information. Class loading is a critical part of the application server environment. As of version 6.0.2 of the application server, the *Class Loader Viewer* provides visibility into the relationship between the class loader, the class path, and the class. The order or precedence of the class loaders are responsible for co-existence between the services provided by WebSphere Application Server V6 and the application functionality. The *drill down* capabilities that the *Class Loader Viewer* provides greatly facilitates resolution of class loader problems. The WebSphere Application Server V6 deployment environment and the application deployment descriptors are tightly linked and require correct information in order for the application to run successfully. The WebSphere Application Server V6 environment provides the services that the application leverages through name space references, such as JNDI names for data sources and resource connections in general. The administrative console installation wizard and the *wsadmin* script interpreter provide APIs for specifying the bindings between the application and the WebSphere Application Server V6 containers.

WebSphere Application Server Toolkit



Using the AST requires some application development knowledge and skills. The AST displays projects based on *perspectives*, and a brief description of the J2EE perspective follows. The J2EE perspective breaks the enterprise application down into categories such as *Enterprise Application*, *EJB Projects*, and *Dynamic Web Projects*. The *Enterprise Application* category lists all of the enterprise applications currently loaded into the AST. The EJB and Web modules that are part of the enterprise application are listed under their *EJB Projects* and the *Dynamic Web Projects* categories, respectively. The enterprise application and its modules contain deployment descriptors that are specific to that component. The deployment descriptor for the *Enterprise Application* lists each of the modules that make up the application and any application level security descriptors. The deployment descriptor for the *EJB Project* contains enterprise bean, session, database and security information, and anything else that the EJBs need for integration with the EJB container. The *Dynamic Web Project* deployment descriptor contains information pertaining to servlets, JSPs, HTML files, and security information.

Class Loader Viewer (1 of 2)

Enterprise Application (rallanCell01 : rallanNode01 : server1 : DefaultApplication.ear : DefaultWebApplication.war)

[Enterprise Application \(rallanCell01 : rallanNode01 : server1 : DefaultApplication.ear : DefaultWebApplication.war\)](#) > **Class Loader Viewer**

Displays information about a classloaders visible to the selected module.

Export Table View Search

Class Loader

- [-] [+] [?] [↑] **JDK Extension Loader**
 - Classes
 - Classpath
- [-] [+] [?] [↑] **JDK Application Class Loader**
 - Classes
 - Classpath
- [-] [+] [?] [↑] **WAS Bootstrap Extension Class Loader**
 - Classes
 - Classpath
- [-] [+] [?] [↑] **WAS Protection Class Loader**
 - Classes
 - Classpath
- [-] [+] [?] [↑] **WAS Module - Jar Class Loader**
 - Classes
 - Classpath
- [-] [+] [?] [↑] **WAS Module - Compound Class Loader**
 - Classes
 - Classpath

The Class Loader Viewer displays an expansion of the class loaders for a Web application module. The Class Loader Viewer is located in the navigation menu on the left of the administrative console under *Troubleshooting > Class Loader Viewer*.

The hierarchy of class loaders used by WebSphere Application Server V6, from highest to lowest, are: Java class loaders, WebSphere extension class loaders, Application module class loaders, and Web module class loader.

Class Loader Viewer (2 of 2)

- A class loader problem will not be detected until the first time the class is required and the JVM attempts to load it
- WebSphere Application Server V6.0.2+ contains an embedded Class Loader Viewer utility in the administrative console that can assist in resolving the problem
- The Class Loader Viewer is structured as a hierarchy based on the cell, nodes of the cell, servers in each node, application on each server, and each module of the application
- Each module is represented as a hyperlink that leads to a page containing the class loader hierarchy
 - ▶ Each class loader in this view contains a tabular view of the class path that it follows and the classes that it loads to satisfy the dependencies of the module

Class loading of WebSphere Application Server V6 enterprise applications are controlled by three parameters: The *class loader mode*, the *war class loader policy* and the *Isolation Policy*. The *class loader mode* parameter can be specified as *parent first* or *parent last*. The default in WebSphere Application Server V6, *parent first*, causes the class loader to delegate the loading of classes to its parent class loader before attempting to load the class from its local class path while *parent last* causes classes to be loaded from the current class path, first. *Parent last* is useful to override functionality of a higher scope and leverage functionality that is local to the application, instead. The *war class loader policy* can be specified as *module* or *application*. The default setting is *module*, meaning that the Web module class loader will be used to load classes for the Web module. Changing the setting to *application* has the effect of using the application module class loader to load the Web module, resulting in the sharing of classes between modules of the same application. If the isolation policy is set to *Single*, then all applications share the same class loader resulting in resources being shared globally among the applications. The *Isolation Policy* has 2 values: *Single* and *Multiple*. The application class-loader policy controls the isolation of applications that are running in the system. When set to *Single*, applications are not isolated. When set to *Multiple*, applications are isolated from each other.

Resolving an application installation error

- Here is a sample of the problems that can be resolved using the AST:
 - ▶ *"Timeout!!!"* error displays when attempting to install an enterprise application in the administrative console
 - If this problem occurs, open the enterprise application in the AST, right-click the enterprise application, and select *Deploy* from the pop-up menu. This action creates a file with a name like **Deployed_filename.ear**. Then, in the administrative console, install the deployed .ear file.
 - ▶ *NameNotFoundException* message displays when deploying an application that contains an EJB module
 - To resolve this problem, open the enterprise application in the AST and expand the *ejbModule* directory of the Enterprise Java Bean (EJB) project. Either remove the source files or include all dependent classes and resource files on the class path
 - ▶ *SRVE0026E: [Servlet Error]-[Unable to compile class for JSP file]*
 - This type of problem is a code defect. The AST can be used to edit the incorrect JSP file and verify the changes. Highlight the JSP in the Project Explorer view, right-click it and select *Run Validation* from the pop-up menu

Resolving application deployment issues using the AST will often involve changing the information in one or more of these deployment descriptors, saving the changes, and exporting the enterprise application for deployment to the application server.

For example, a "Timeout" error may display when attempting to install an enterprise application. By loading the application into the AST and exporting it with a new name such as `Deployed_filename.ear`, the timeout will cease to be a factor. The timeout occurs because the application was not been deployed.

Resolving an application start-up error (1 of 2)

- Some typical problems that can arise while starting an application
 - ▶ *java.lang.ClassNotFoundException: classname Bean_AdderServiceHome_04f0e027Bean*
 - Totally uninstall the application from WebSphere Application Server V6
 - Open the EAR file in the AST, right-click the application and select *Deploy* from the pop-up menu to produce new deployment code
 - Save the application and export the EAR file
 - Re-install the application into WebSphere Application Server V6 (if you are using the wsadmin command line tool, remember to save the configuration using \$AdminApp save)
 - ▶ Deployed application that uses Java Native Interface (JNI) code does not start
 - To fix such an issue, recompile the native code for the OS

Some `ClassNotFoundException`s occur when one tries to start an undeployed application containing EJBs, or containing undeployed EJB modules. To correct this error, uninstall the application, open the EAR file in the Application Server Toolkit, deploy, save, and export the EAR file, and re-install.

An application may include Java Native Interface code. If an application is migrated from one operating system to another, the code may be incompatible, such as 32 bit versus 64 bit addressing. To fix such an issue, recompile the native code for the target operating system.

Resolving an application start-up error (2 of 2)

- More typical problems that can arise while starting an application
 - ▶ **A ConnectionFac E J2CA0102E: Invalid EJB component: Cannot use an EJB module with version 1.1 using The Relational Resource Adapter error occurs**
 - In order to fix this problem the association between EJB 1.1 and the default data source needs to be deleted and the version 4 data source needs to be associated with EJB version 1.1
 - ▶ **NMSV0605E: "A Reference object looked up from the context..." error when starting an application**
 - Verify that the JNDI Name given in administrative console under **Resources-> JDBC Providers > (JDBC provider name) > Data sources > (Data source name) > JNDI name** matches the JNDI Name given for CMP or BMP Resource Bindings at the time of assembling the application in an assembly tool.

Errors relating to incorrectly mapped resources may not arise until application start-up. Problems of compatibility may arise. WebSphere Application Server V6 is backwards compatible to older EJB specification levels, such as EJB 1.1. If the EJB module being deployed is an older version the default data source it specifies and the default data source for WebSphere Application Server V6 will be incompatible. In order to fix this problem the association between EJB 1.1 and the default data source needs to be deleted and the version 4 data source needs to be associated with EJB version 1.1. Naming issues are another common start-up error. Sometimes a data source intended to support a CMP enterprise bean is not correctly associated with the enterprise bean. To resolve this problem: Select the **Use this Data Source in container managed persistence (CMP)** check box in the data source "General Properties" panel of the administrative console. Verify that the JNDI Name given in administrative console under **Resources-> JDBC Providers > (JDBC provider name) > Data sources > (Data source name) > JNDI name** matches the JNDI Name given for CMP or BMP Resource Bindings at the time of assembling the application in an assembly tool. Alternatively, check the JNDI Name for CMP or BMP resource bindings specified in the code by J2EE Application Developer. Open the deployed .ear folder in an assembly tool, and look for the JNDI Name for your entity beans under CMP or BMP resource bindings. Verify that the names match.

Resolving an application first-run issue (1 of 2)

- Some examples of runtime problems and their resolutions are:
 - ▶ **Message similar to “Message: /jspname.jsp(9,0) Include: Mandatory attribute page missing” displays when trying to access JSP file**
 - To resolve this problem, check the logs for the actual line number that the error occurred on in the JSP, open the JSP in the AST, and correct the reference error.
 - ▶ **A “Page cannot be displayed... server not found or DNS error” error displays when attempting to browse a JSP file using Internet Explorer**
 - To correct this problem, increase the ConnectionIOTimeOut value:
 - i.) From the administrative console, select **System administration > Deployment manager > Administration Services > Custom Properties**
 - ii.) Select ConnectionIOTimeOut.
 - iii.) Increase the ConnectionIOTimeOut value.
 - iv.) Click OK.

Runtime problems could be caused by an error in a JSP file. There could be an invalid reference in an include statement or some other problem when the compiler attempts to resolve a resource in the page. To resolve this problem, check the logs for the actual line number that the error occurred on in the JSP, open the JSP in the AST, and correct the reference error. Sometimes an error can occur when an HTTP timeout causes the servant to be brought down and restarted. To correct this problem, increase the ConnectionIOTimeOut value:

- i.) From the administrative console, select **System administration > Deployment manager > Administration Services > Custom Properties**
- ii.) Select ConnectionIOTimeOut.
- iii.) Increase the ConnectionIOTimeOut value.
- iv.) Click OK.

Resolving an application first-run issue (2 of 2)

- Some examples of runtime problems and their resolutions are:
 - ▶ *javax.naming.NameNotFoundException*: Name name not found in context "local"
 - To solve this problem, change the calling method to be remote or promote the local method into the remote interface
 - ▶ *org.omg.CORBA.BAD_PARAM*: Server is not of the expected type. minor code: 4942F21E completed: No
 - To resolve this issue, change the application class loading scheme to multiple, then stop and start the application

Some runtime problems are caused by an EJB client attempting to access an EJB through a local interface. This might not be caught during development if the client and the EJB were being developed inside the same JVM. To solve this problem, change the calling method to be remote or promote the local method into the remote interface.

Errors can also occur when an application server is configured to use a single class loading scheme. When redeploying an application, the classes will remain loaded even though the application was uninstalled. To resolve this issue, change the application class loading scheme to multiple, then stop and start the application

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject= Feedback about SW5706G11_AppDeployProblems.ppt



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

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

IBM WebSphere

Internet Explorer, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

EJB, Java, JavaServer, JavaServer Pages, JDBC, JNI, JSP, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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 or changes in the products or programs described herein at any time without notice.

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 (for example, 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.

© Copyright International Business Machines Corporation 2007. All rights reserved.

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.