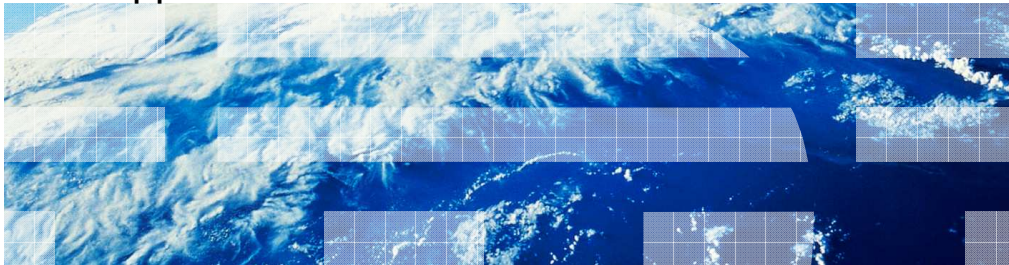


IBM WebSphere Application Server Feature Pack for OSGi Applications and Java Persistence API 2.0

JEE support



This module discusses JEE Support for IBM WebSphere® Application Server Feature Pack for OSGi Applications and Java™ Persistence API 2.

Table of contents



- What is meant by JEE support?
 - WABs
 - Resource references
 - JNDI support
 - JMS
 - Security
 - JEE support – Other topics
- Troubleshooting

The point of this presentation is to explore the Java Enterprise (JEE) support technologies that now exist for OSGi. At the end of this presentation, you'll have an idea of what technologies are available to users, and an idea of how they might be used in practice.

It will cover just what is meant by JEE support, and then go into more detail about some of the aspects of JEE Support that is provide; web application bundles (WABs), Resource references, JNDI Support, JMS, and Security,

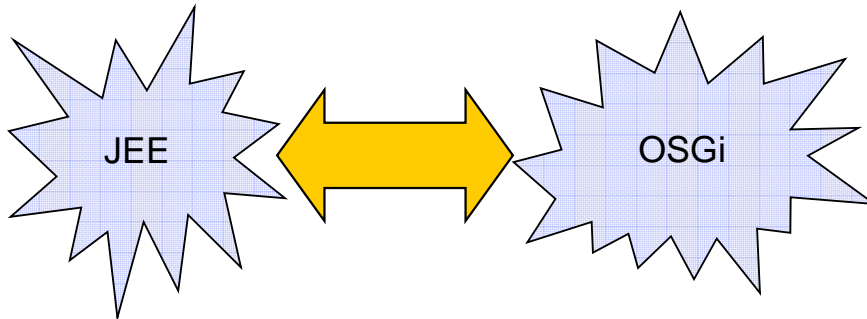
There is a brief treatment of other topics such as transactions and JMX.

Also, there is a brief look at how a JEE application might be converted into an OSGi one.

Finally there will be some troubleshooting comments.



JEE support – The big picture



So just what is JEE Support?

It's essentially the “glue” between using JEE technologies, and OSGi applications. Helping two different technologies fit together.

OSGi provides a lot of benefits such as life cycle and dependency management. Java Enterprise has a lot of mature and well-known APIs such as JNDI, JPA, and JTA which enterprise with which developers are very familiar, and which form the basis of many existing Java enterprise applications. Rather than reinventing the wheel, it makes sense for OSGi to reuse the APIs that already exist for JEE. The problem is that Java EE and Java SE are not OSGi aware. So to enable OSGi developers to make use of JEE technologies some sort of glue or bridge is needed between them – and this is what JEE Support provides.

Big picture



▪ JEE Support

- OSGi with Java EE
 - The point is to make OSGi applications more usable to JEE users
 - OSGi for Java enterprise applications is one focus of Version 4.2 of the OSGi service platform, which introduces the OSGi Service Platform Enterprise Specification.
 - Encompasses many things:
 - Web support (WABs)
 - resource references
 - JNDI support
 - Java 2 security and application security
 - JMS, JTA, JDBC, JMX....
 - (also JPA and blueprint covered in more detail elsewhere)
 - See OSGi Service Platform Release 4 Version 4.2 <http://www.osgi.org>

JEE Support within the OSGi applications feature pack refers to several different technologies. All these technologies aim to bring the benefits of OSGi to the enterprise application developer.

The point is to make OSGi applications more usable to JEE users, enable OSGi applications to work better in an enterprise environment making use of JEE artifacts, and enabling the benefits of OSGi like modularity, dependency injection, and version control to be used in an enterprise environment which provides benefits such as scalability, reliability, and robustness.

OSGi for Java enterprise applications is one focus of Version 4.2 of the OSGi service platform, which introduces the OSGi Service Platform Enterprise Specification.

It encompasses many things: web Support (WABs), resource references, JNDI support, Java 2 security and application security, JMS, JTA, JDBC, JMX, also JPA and blueprint covered in more detail elsewhere.

To read the details, go to the web site on this slide at <http://www.osgi.org>

Component details



This section will go into more of the details of JEE Support.

JEE support - WABs



- WABs (web application bundle)
 - Summary
 - Web modules can consist of HTML + JSP pages to provide user interface and program logic, servlets for business logic
 - WABs are the OSGi version of WARs
 - Bundles – Web-ContextPath: /myContextPath in manifest
 - Usage scenarios
 - WARs within EBAs/EARs are converted directly into WABs
 - WABs can be placed into bundle repositories and used by reference
 - Access the bundle context from within servlet:
 - `BundleContext context = (BundleContext) servletContext.getAttribute("osgi-bundlecontext");`
 - Servlet context registered to OSGi service registry on startup

6

JEE support

© 2011 IBM Corporation

In the Enterprise world, WARs, web application archives, are a familiar concept. They are jar files containing various web components such as JavaServer Pages, servlets and HTML pages and the appropriate metadata.

The OSGi equivalent of a WAR is a WAB, a web application bundle. WABs can be deployed just like any other bundle. They have bundle symbolic names (BSN), package imports and exports. There is one required manifest header to define a WAS,

Web-ContextPath: /myContextPath.

WARs can also be converted directly into WABs, any ear that is migrated to OSGi containing WARs will result in those WARs being turned into WABs.

WARs are converted to WABs using a defined URL handler which adds required headers to the manifest.

Note on this slide how the access to bundle context from within a servlet is given through the statement:

```
BundleContext context = (BundleContext) servletContext.getAttribute("osgi-bundlecontext");
```

Servlet context is registered in the OSGi service registry for every successfully started web application.

This is likely to be a very popular way of getting started with OSGi due to widespread familiarity with WARs.

JEE support – Resource references (1 of 2)



- JEE - declared in the deployment descriptor by application provider
- OSGi, defined with blueprint and registered to OSGi service registry. Only this bundle is allowed to look it up – for security purposes.
- Bound to the actual runtime configuration name during application deployment
- Binding contained within ibm-eba-bnd.xml

7

JEE support

© 2011 IBM Corporation

Resource references support application provides access to a resource, for example, data source, URL, or mail provider, using a logical name instead of the actual name in the runtime environment.

Within both JEE and OSGi these resource references are bound during deployment.

In the JEE world, these are declared in the deployment descriptor by the application provider.

In the OSGi world, resource references are defined with blueprint and registered to OSGi service registry. For security reasons only this bundle is allowed to look it up.

The resource reference is bound to the actual runtime configuration name during application deployment completing the connection between the application code and the service provider.

The binding information is contained within the ibm-eba-bnd.xml file.



JEE support – Resource references (2 of 2)

Snippets from the Blueprint xml

```
xmlns:rr="http://www.ibm.com/appserver/schemas/8.0/blueprint/resourcereference"

<bean id="testResourceReferenceNonSharedScopeBean"
class="some.package.ResourceReferenceClass" init-method="launch">
  <argument value="false"/>
  <property name="dataSource" ref="nonSharedResRef"/>
</bean>

<rr:resource-reference id="nonSharedResRef"
  interface="javax.sql.DataSource"
  filter="(osgi.jndi.service.name=jdbc/AccountDS1)">
  <rr:res-auth>Container</rr:res-auth>
  <rr:res-sharing-scope>NonShareable</rr:res-sharing-scope>
</rr:resource-reference>
```

ibm-eba-bnd.xml

```
<eba-bnd>
  <resource-ref>
    <jndi-name>jdbc/AccountDS1</jndi-name>
    <authentication-alias>Alias1</authentication-alias>
  </resource-ref>
</eba-bnd>
```

8

JEE support

© 2011 IBM Corporation

This slide shows an example of how a resource reference is defined within a blueprint xml file.

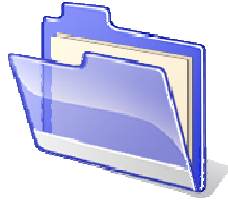
In this case the resource reference is for a JDBC data source jdbc/AccountDS1.

The ibm-eba-bnd.xml file defines the binding for the resource references.

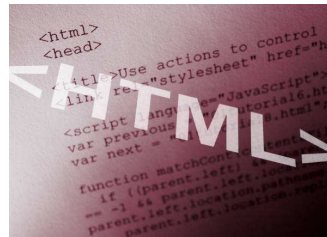
The binding to the actual runtime configuration name is done when the EBA asset is imported into the BLA.

JEE support details - Security

Java 2 security



Application security



Security support can be split into two sections:

Java 2 security which is all about controlling access to system resources and web application security which controls user authentication to application resources including web resources such as JavaServer Pages files and servlets.

There are no administrative security settings for OSGi, although to enable application security, WebSphere requires that administrative security is also enabled.

JEE support – Security



- Java 2 security
 - Guards access to system resources for example sockets and files
 - In OSGi uses (Conditional) Permission Administration Service to apply permissions to bundles – dynamic, activated immediately once set.
 - Some default permissions denied/granted. Additional ones provided to match WebSphere Application Server application policy if no permissions already provided.
 - Hierarchical permissions.perm – at application level, and within bundles if required

Warning: Must be set in server.policy file

```
grant codeBase "file:${was.install.root}/feature_packs/aries/plugins/-" {  
    permission java.security.AllPermission;  
};  
  
grant codeBase "file:${was.install.root}/feature_packs/aries/osgiappbundles/-" {  
    permission java.security.AllPermission;  
};
```

10

JEE support

© 2011 IBM Corporation

Java 2 security guards access to system resources, for example sockets and files.

In OSGi the Conditional Permission Administrative Service is used to apply permissions to bundles.

This is done dynamically and permissions are activated immediately once set.

Some permissions are applied to all applications. These include denying permission to key services, such as the conditional permission administrative service itself!

Applications without their own permissions set are given default permissions designed to match the default permissions set by WebSphere Application Server for JEE applications.

Note that if an applications then adds its own permissions, the default permissions provided by the system will no longer be provided, so if any of them are needed by the application, they will also need to be added.

Note the warning here about checking that these extra entries exist in the server.policy file. This should not be necessary if the feature pack has been installed correctly, but is worthwhile checking if you are having a lot of trouble with Java 2 security exceptions.

JEE support details – Java 2 security policy files

Sample WebSphere Application Server policy file content

```
// The following permissions apply to all the
components under the application.
grant codeBase "file:${application}" {
    // The following are required by JavaMail
    permission java.io.FilePermission
        "{was.install.root}${/}lib${/}activation-impl.jar",
        "read";
    .

// The following permissions apply to all utility .jar
files
grant codeBase "file:${jars}" {
    permission java.net.SocketPermission "*",
        "connect";
    permission java.util.PropertyPermission "*",
        "read";
    .
```

Sample OSGi permissions file content

```
(java.io.FilePermission "${was.install.root}/-" "read,write")
(org.osgi.framework.PackagePermission "*" "exportonly")
(org.osgi.framework.ServicePermission "datasource.TestEBA" "register")
```



Java 2 security enables users to specify particular permissions for their applications.

WebSphere Application Server traditionally does this using policy files.

OSGi does this with permission files (permissions.perm)

WebSphere Application Server policy files, as can be seen above, are based around setting permissions for different code bases. This is a major difference compared with how permissions work with OSGi. To set permissions at an application level, like WebSphere Application Server .policy files, the permissions.perm file is located in the META-INF directory of the application, that is the EBA.

Finer granularity of permissions can be provided by having additional permissions.perm files within the OSGI-INF directories of any bundles within the EBA.

Another key thing to notice from this example is that the format of these files is different. The format of these files is crucial! Be careful.

Note especially that these lines are not meant to span lines. They span lines here so they can fit on the page but in reality they cannot. Each permission needs to be on one line, in their specific, and slightly differing, formats.

For OSGi permissions, the file is an optional one, called permissions.perm, and located in META-INF directory. It is UTF-8, all single-line, with comments allowed. The format of a permission is "Open Bracket"- "Permission"- "Space"- "name in quotation marks"- "Space"- "comma separated actions in quotation marks"- "Close bracket"

If you have syntax that is not valid, the application will not start.

JEE support – Security

- Application security
 - Guards access to web resources for example JavaServer Pages files and servlets
 - Very similar to JEE, configuration when adding asset to BLA
 - Access control by web container
 - Access manager decides authorization based on security policy from deployment descriptor
 - Authenticated user principal allowed access if it has one of required security roles



Web application security works in much the same way as it does with JEE.

It guards access to web resources, for example JavaServer Pages files and servlets and access control is handled by the web container

Access manager decides authorization based on security policy from the deployment descriptor.

Configuration of application security is done when adding asset to BLA

JEE support details - Security

Sample web.xml

```
<servlet id="Security Config 1">
  <servlet-name>Security Servlet 1</servlet-name>
  <display-name>Security Servlet 1</display-name>
  <description>Servlet to verify security</description>
  <servlet-class>some.package.SecurityServlet</servlet-class>
  <load-on-startup></load-on-startup>
  <security-role-ref>
    <description>1st role ref</description>
    <role-name>LOCALROLE</role-name>
    <role-link>ROLE1</role-link>
  </security-role-ref>
</servlet>
...
<servlet-mapping id="SecurityServletMappingConfig1">
  <servlet-name>Security Servlet 1</servlet-name>
  <url-pattern>/security1/*</url-pattern>
</servlet-mapping>
...
```

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>config 1</web-resource-name>
    <url-pattern>/security1/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
...
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>Security Auth Domain</realm-name>
  <form-login-config>
    <form-login-page>/login.html</form-login-page>
    <form-error-page>/error.html</form-error-page>
  </form-login-config>
</login-config>
```



This slide shows an example of a web.xml deployment descriptor for a secured WAB.

None of this is specific to OSGi so if you need to find out more about this, the standard security sections of the information center is the place to look.

In brief, this shows how the security role can be defined, an example of a security constraint, the security runtime matches against the URL-pattern, and the transport guarantee of confidential ensures it uses HTTPS. This particular example is using form-based authentication as can be seen from the login configuration.

JEE support – JNDI support



- JNDI Support
 - OSGi services registered in registry can be looked up with JNDI by JNDI clients
 - `osgi:service/<interface_name>/<filter>`
 - `osgi:serviceList/<interface_name>/<filter>`
 - OSGi bundles can look up JNDI in the OSGi service registry
 - JNDIContextManager service gives JNDI initial context
 - `new InitialContext()` possible, but use service registry instead

```
InitialContext ctx = new InitialContext();  
DataSource ds = (DataSource) ctx.lookup(  
    "osgi:service/javax.sql.DataSource/(osgi.jndi.service.name=jdbc/mydatasource)");
```

14

JEE support

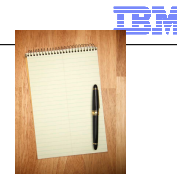
© 2011 IBM Corporation

The **JNDI** specification provides a bridge between JNDI and the OSGi service registry. It enables the lookup of an OSGi Service or BundleContext through JNDI. It also describes how you can obtain a JNDI Initial Context from the OSGi Service registry in case your bundle needs to do a JNDI lookup or bind.

Using “`new InitialContext()`” is probably more familiar to JEE developers but it is better to use the `JNDIContextManager` service when possible.

The example shows how you can get the JNDI Initial context from within a bundle, and use it to look up a data source registered as an OSGi service.

JEE support – JMS support



- JMS support
 - **JMS and OSGi applications**
 - An OSGi application can send and receive Java Message Service (JMS) messages.
 - Each reference to a JMS resource is declared in a Blueprint XML file
 - Your OSGi application can bind to any of these JMS resource types:
 - Default messaging JMS queues destinations
 - Default messaging JMS topic destinations
 - Generic JMS connection factory
 - JMS queue connection factory for the JMS provider of WebSphere MQ
 - JMS queue destination for WebSphere MQ
 - JMS topic connection factory for WebSphere MQ
 - JMS topic destination for WebSphere MQ
 - Unified JMS connection factory for WebSphere MQ
 - You bind JMS resource references to your OSGi application when you add the EBA asset to a business-level application.
 - Usage scenarios
 - Normal JMS messaging
 - [Enabling OSGi bundle or a web application bundle \(WAB\) to interact with Enterprise Java Beans - configure your OSGi application to send JMS messages to destinations](#)

15

JEE support

© 2011 IBM Corporation

An OSGi application can send and receive Java Message Service (JMS) messages. Your OSGi application can use JMS to interact with Enterprise JavaBeans.

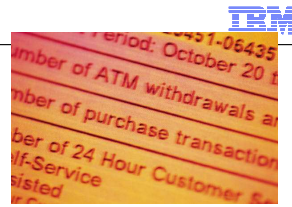
OSGi applications can use JMS resources that are configured within WebSphere Application Server, in a similar way to using JMS resources with Java EE applications. For OSGi applications, each reference to a JMS resource is declared in a Blueprint XML file. Each bundle in an OSGi application can contain any number of resource reference declarations in its various Blueprint XML files.

Your OSGi application can bind to any of several JMS resource types. Those types include; Default messaging JMS queues destinations, Default messaging JMS topic destinations, Generic JMS connection factory, JMS queue connection factory for the JMS provider of WebSphere MQ, JMS queue destination for WebSphere MQ, JMS topic connection factory for WebSphere MQ, JMS topic destination for WebSphere MQ and Unified JMS connection factory for WebSphere MQ

You bind JMS resource references to your OSGi application when you add the EBA asset to a business-level application.

Using JMS is one method to allow bundles and WABs to interact with EJBs. The OSGi application can send JMS messages to a destination, and the EJBs or message driven beans (MDBs) can be used to retrieve the messages from those destinations and respond to them.

JEE support – Other topics



- JTA support
 - OSGi services available in the registry
 - Javax.transaction.UserTransaction
 - Javax.transaction.TransactionManager
 - Javax.transaction.TransactionSynchronizationRegistry
 - Supports XA resources
- JMX support
 - MBean services allowing the querying of aspects of a running OSGi framework
- JDBC support
 - Service registered exporting DataSourceFactory API. Provider can look up based on driver type

16

JEE support

© 2011 IBM Corporation

Here are briefly covered other aspects of JEE Support that is provided. These are covered in other modules.

There is transaction support in the form of JTA (Java Transaction API). XA resources are supported and these OSGi services are made available for use in the OSGi service registry. These include; `javax.transaction.UserTransaction`, `javax.transaction.TransactionManager` and `javax.transaction.TransactionSynchronizationRegistry`

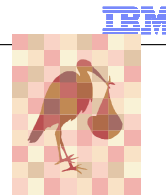
JTA allows applications to perform distributed transactions, for example transactions which access or update data across multiple networked resources

There is also JMX support (Java Management Extensions) which enables you to find out the status of a running OSGi framework and a selection of other services.

JDBC support is also provided where database drivers are registered to the OSGi service registry exported as `DataSourceFactory` objects.

JPA and Blueprint are also key parts of the Enterprise Specification. To find out more about those, see the presentations which specifically cover these.

Converting a JEE application into an OSGi one



- Automatic? Partial automatic? Manual?
- Starting point: a single EAR file.
- Only web modules? Automatic – all WARs to WABs
- Utility JAR files (local using Class-Path or external using Extension-List)? Manual steps
 - Move jar to WEB-INF/lib or convert JAR to OSGi bundle
- Enterprise JavaBeans? Manual steps. Ears containing EJBs can be converted and imported, but EJBs will not get converted.
 - Rewrite application logic or use JMS messages to interact or use SCA and specify bindings
- Large or complex? Partial conversion focus on areas such as:
 - Areas that use a utility JAR file, so that the application footprint is reduced.
 - Areas that benefit from the version control capabilities of OSGi.
 - Areas that benefit from the dependency injection approach of OSGi.
- **Migration of Java 2 security settings**
 - was.policy file converted into a permissions.perm
 - all permissions are promoted to the application level.
 - Finer granularity? Manual steps needed

17

JEE support

© 2011 IBM Corporation

This slide provides information on how to convert a JEE application into an OSGi one. These steps are the module about OSGi applications, and the information is also in the information center, so there are not many details here.

Essentially, when you want to convert an EAR file, you need to look at the contents of it, and evaluate how much of it you want to convert, what manual steps need to be taken, and how much of it can be automatically converted.

Some enterprise applications can be converted automatically to an OSGi application, some require manual conversion steps, and some require only partial conversion.

It is assumed that the enterprise application that you convert is a single EAR file.

You can automatically convert an enterprise application that contains only web modules. An EAR file that contains only web application archive (WAR) files can be converted to an EBA file. When you import the EBA file as an asset, the WAR files are automatically converted to web application bundles, WABs.

You can't automatically convert an enterprise application that contains utility JAR files. These utility JAR files might be in the EAR file, that is, referenced by using a Class-Path attribute, or in an external location, that is, referenced by using an Extension-List attribute. You need manual conversion steps.

You can't automatically convert an enterprise application that contains Enterprise JavaBeans (EJB). You can convert the EAR file to an EBA file and import that file as an asset. However, the EJB JAR files are not converted, and the EJB function is not directly accessible in the EJB container. You need manual conversion steps.

With a large or complex application, it might be appropriate to convert only aspects of the application. For example, these areas of the application might be suitable for conversion:

Areas that use a utility JAR file, so that the application footprint is reduced.

Areas that benefit from the version control capabilities of OSGi.

Areas that benefit from the dependency injection approach of OSGi.

When you migrate an application from Java EE to OSGi, any existing was.policy file is converted into a permissions.perm file to be used with the OSGi permissions framework, and permissions are all promoted to application level. If finer granularity of permissions is needed, these extra permissions files will need to be added in separately.

Summary

- This session has covered various aspects of JEE support within OSGi and how they can be used in practice
- Information center
- OSGi Service Platform Release 4 Version 4.2
 - Enterprise Specification
- <http://osgithoughts.blogspot.com/>

This session covered Java Enterprise support within OSGi.

If you need to find out more about anything covered here, the information center for the feature pack is invaluable.

Also check out the OSGi Service Platform Release 4, Version 4.2 specification, in particular the enterprise specification.

Here also notice the a blog listed on the slide which has a nice presentation about JEE.



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 email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_wasosgijpafep_OAGi_JEE_support.ppt

This module is also available in PDF format at: [../wasosgijpafep_OAGi_JEE_support.pdf](..../wasosgijpafep_OAGi_JEE_support.pdf)

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



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. in the United States, other countries, or both.

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

© Copyright International Business Machines Corporation 2010. All rights reserved.