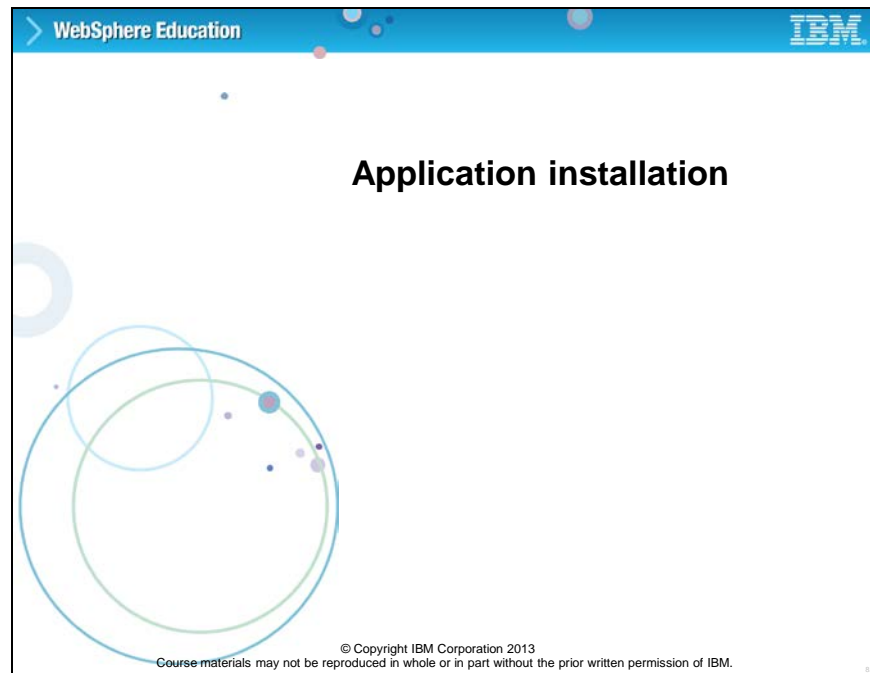
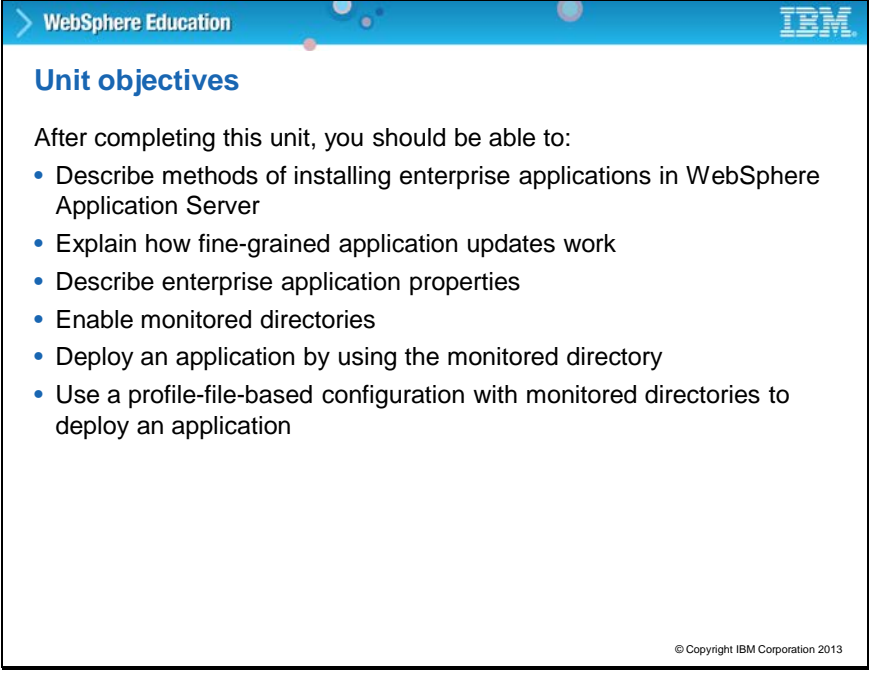


Slide 1



Application installation. This unit describes how to install enterprise applications into WebSphere Application Server.

Slide 2

The slide is titled 'WebSphere Education' in the top left corner and features the IBM logo in the top right corner. The main heading is 'Unit objectives' in blue. Below this, it states 'After completing this unit, you should be able to:' followed by a bulleted list of seven objectives. The slide also includes a copyright notice at the bottom right.

> WebSphere Education **IBM**

Unit objectives

After completing this unit, you should be able to:

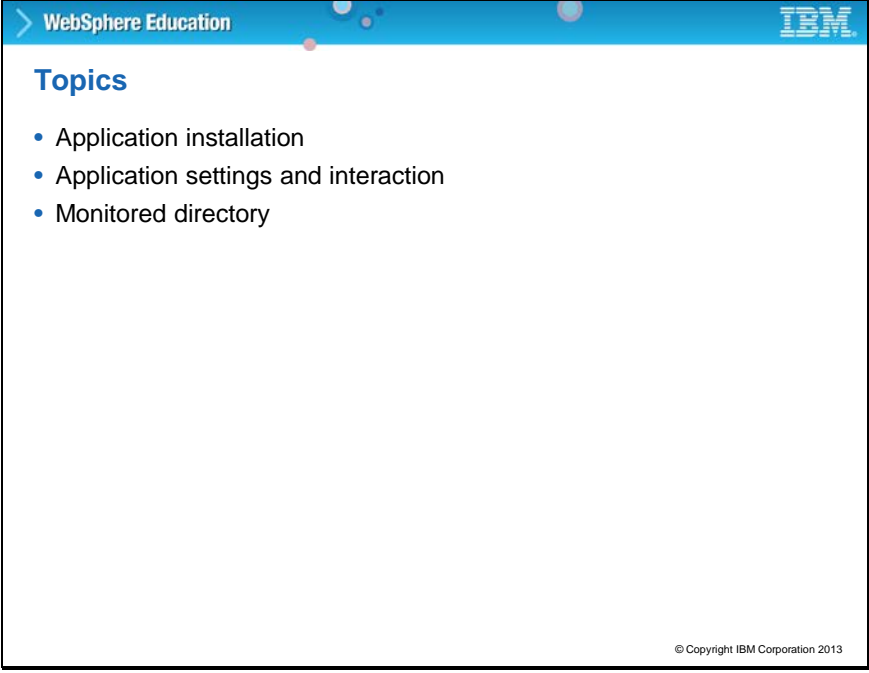
- Describe methods of installing enterprise applications in WebSphere Application Server
- Explain how fine-grained application updates work
- Describe enterprise application properties
- Enable monitored directories
- Deploy an application by using the monitored directory
- Use a profile-file-based configuration with monitored directories to deploy an application

© Copyright IBM Corporation 2013

After completing this unit, you should be able to:

- Describe methods of installing enterprise applications in WebSphere Application Server
- Explain how fine-grained application updates work
- Describe enterprise application properties
- Enable monitored directory
- Deploy an application by using the monitored directory
- Use a property-file-based configuration with monitored directories to deploy an application

Slide 3



The slide features a blue header bar with the text 'WebSphere Education' on the left and the 'IBM' logo on the right. Below the header, the word 'Topics' is written in a bold blue font. Underneath 'Topics', there is a bulleted list of three items: 'Application installation', 'Application settings and interaction', and 'Monitored directory'. In the bottom right corner of the slide, there is a small copyright notice: '© Copyright IBM Corporation 2013'.

WebSphere Education

IBM

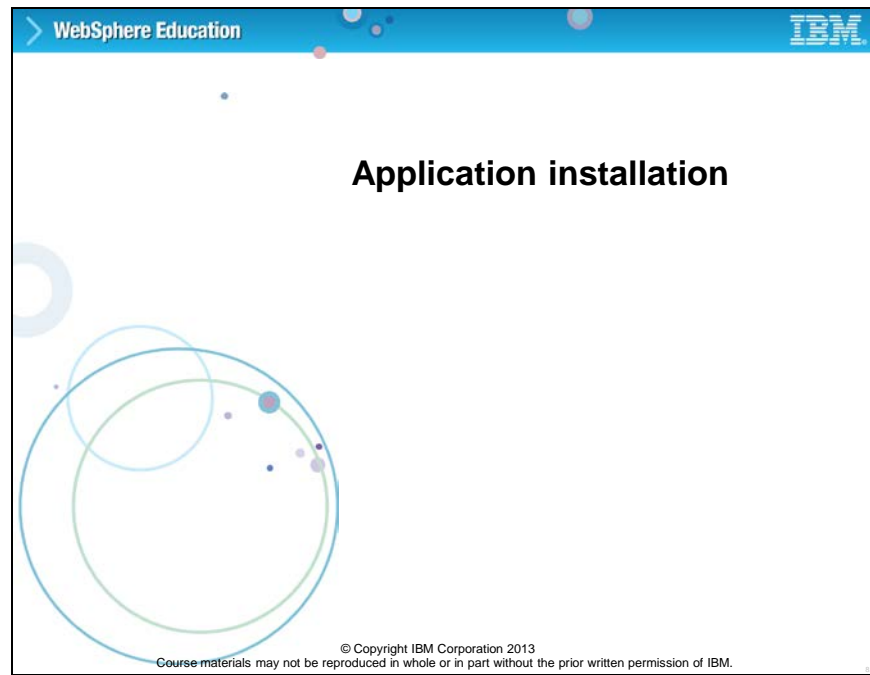
Topics

- Application installation
- Application settings and interaction
- Monitored directory

© Copyright IBM Corporation 2013

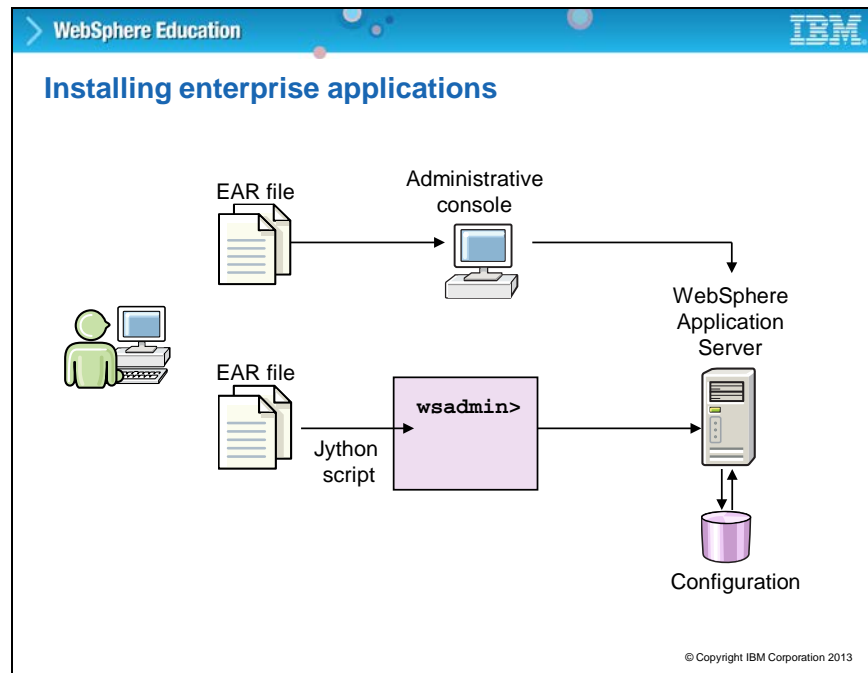
This unit is divided into three topics.

Slide 4



Topic: Application installation. In this topic, you learn the various methods of installing an application.

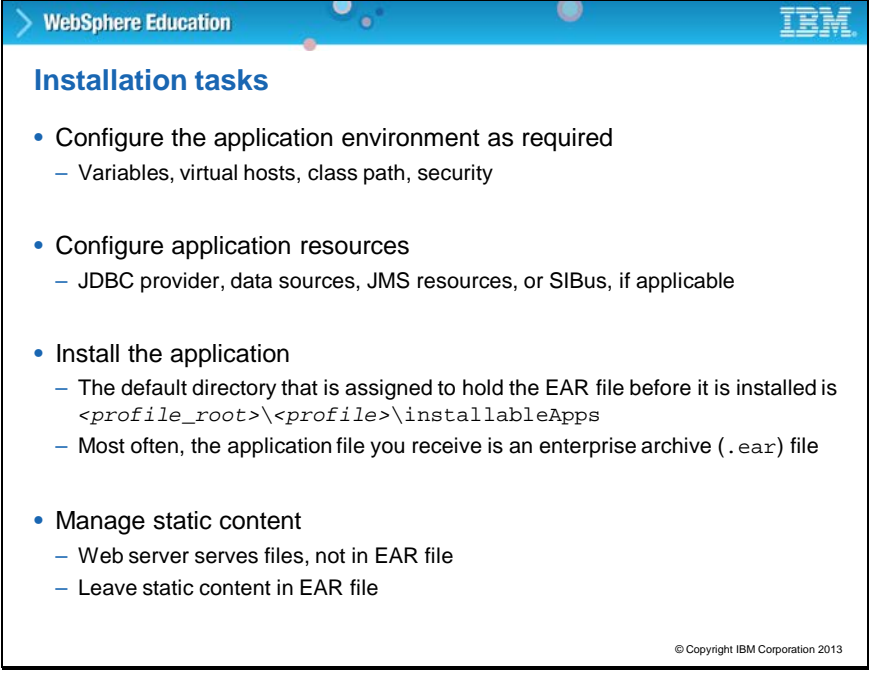
Slide 5



The two most common ways to install enterprise applications on the server are to use the administrative console or wsadmin scripts. The administrative console is the easier method of installing or updating an application. Wizards take you through the process of installing an application and provide helpful information at each step.

Using wsadmin scripts is more complicated than other methods. It requires skills in at least one of the supported scripting languages, and requires complete understanding of application configuration.

Slide 6



The slide is titled "WebSphere Education" and "Installation tasks". It contains a bulleted list of four main tasks, each with sub-points. The tasks are: 1. Configure the application environment as required (Variables, virtual hosts, class path, security). 2. Configure application resources (JDBC provider, data sources, JMS resources, or SIBus, if applicable). 3. Install the application (The default directory that is assigned to hold the EAR file before it is installed is `<profile_root>\<profile>\installableApps`; Most often, the application file you receive is an enterprise archive (.ear) file). 4. Manage static content (Web server serves files, not in EAR file; Leave static content in EAR file). The IBM logo is in the top right corner, and the copyright notice "© Copyright IBM Corporation 2013" is in the bottom right corner.

- Configure the application environment as required
 - Variables, virtual hosts, class path, security
- Configure application resources
 - JDBC provider, data sources, JMS resources, or SIBus, if applicable
- Install the application
 - The default directory that is assigned to hold the EAR file before it is installed is `<profile_root>\<profile>\installableApps`
 - Most often, the application file you receive is an enterprise archive (.ear) file
- Manage static content
 - Web server serves files, not in EAR file
 - Leave static content in EAR file

© Copyright IBM Corporation 2013

Installing an application involves configuring the runtime environment as required. You can define variables, virtual, hosts, and any other resources that the application needs before you actually install the application. In general, it is a good practice to leave the static content that the application uses in the EAR file and allow the infrastructure to take care of serving and caching the static content. In the end, leaving the static content in the EAR file can be as effective as moving it to the web server.

Slide 7

WebSphere Education
IBM

Creating a J2C authentication alias

- Wizard can be reached from many pages in the administrative console
 - Security > Global Security > Authentication > Java Authentication and Authorization Service > J2C authentication data**
- Provide
 - Alias name: console prefixes name with node name
 - User ID and corresponding password
 - Optional description
 - EJBs, data sources, JMS resources, and SIBus resources use them

Global security

Global security > JAAS - J2C authentication data > New...

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

Alias

PlantsApp

User ID

db2admin

Password

Description

For PlantsByWebSphere App

Apply

OK

Reset

Cancel

© Copyright IBM Corporation 2013

In this scenario, a J2C authentication alias is created for accessing resources that the application uses. You can access the page to create new J2C aliases from most pages in the console where an authentication alias is required. To create a J2C alias, provide a name, user ID, and password.

J2C aliases can also be created through scripting. J2C aliases are not scoped resources, so when installing an enhanced EAR file that has a J2C alias definition within it, the alias **is** created and available to any resource in the server. The administrative console adds the node name to the alias name you provide.

Slide 8

WebSphere Education

Creating a data source (1 of 3)

Create a data source

- Step 1: Enter basic data source information
- Step 2: Select JDBC provider
- Step 3: Enter database specific properties for the data source
- Step 4: Setup security aliases
- Step 5: Summary

Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your application is based on the Enterprise JavaBeans(TM) (EJB) 1.1 specification or the Java(TM) Servlet 2.2 specification

Scope:

* Data source name:

* JNDI name:

Select JDBC provider

Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

☒ Create new JDBC provider

☐ Select an existing JDBC provider

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope:

* Database type:

* Provider type:

* Implementation type:

* Name:

Description: Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.

- Create JDBC provider before or while defining data sources
- One JDBC provider is needed for each database driver type
- JDBC providers can be defined at cell, node, server, or application scope (in an enhanced EAR file)


© Copyright IBM Corporation 2013

These screen captures show the process for creating the required resources that an application uses to access a database. A data source is associated with a JDBC provider that supplies the specific JDBC driver implementation class. The data source represents the Java EE Connector Architecture (JCA) connection factory for the relational resource adapter.

Application components use the data source to access connection instances to a specific database. A connection pool is associated with each data source. You can create multiple data sources with different settings and associate them with the same JDBC provider. One reason to create multiple data sources with different settings is to provide access to different databases. This series of slides shows you how to create a data source. You can create the JDBC provider for the data source either before or while defining the data source. They can be defined in the data source wizard. One JDBC provider is needed for each database driver type used.

A JDBC provider is a scoped resource that can be defined at the cell, node, or server level, or at the application scope if included in an enhanced EAR file. The data source creation wizard is shown here. After entering a name for the data source, you are prompted to create or select an existing JDBC provider. If you are creating a JDBC provider, you specify that information here.

Slide 9

WebSphere Education 

Creating a data source (2 of 3)

- JDBC driver paths can be defined in the wizard

Class path:

```
#{DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar  
#{UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar  
#{DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
```

Apply

Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable `#{DB2UNIVERSAL_JDBC_DRIVER_PATH}`:

C:\Program Files\IBM\SQLLIB\java

Native library path


Directory location which is saved as WebSphere variable `#{DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}`:

C:\Program Files\IBM\SQLLIB\java

© Copyright IBM Corporation 2013

On this page, you specify the class path, the location of the JDBC driver, and the native library path. The paths are stored in WebSphere variables that the server uses to locate the JDBC driver code.

Slide 10

WebSphere Education 

Creating a data source (3 of 3)

- Provide the database-specific parameters:
 - Driver type
 - Database name
 - Database server name and communication port number
- Choose whether data source is going to be used with CMP beans

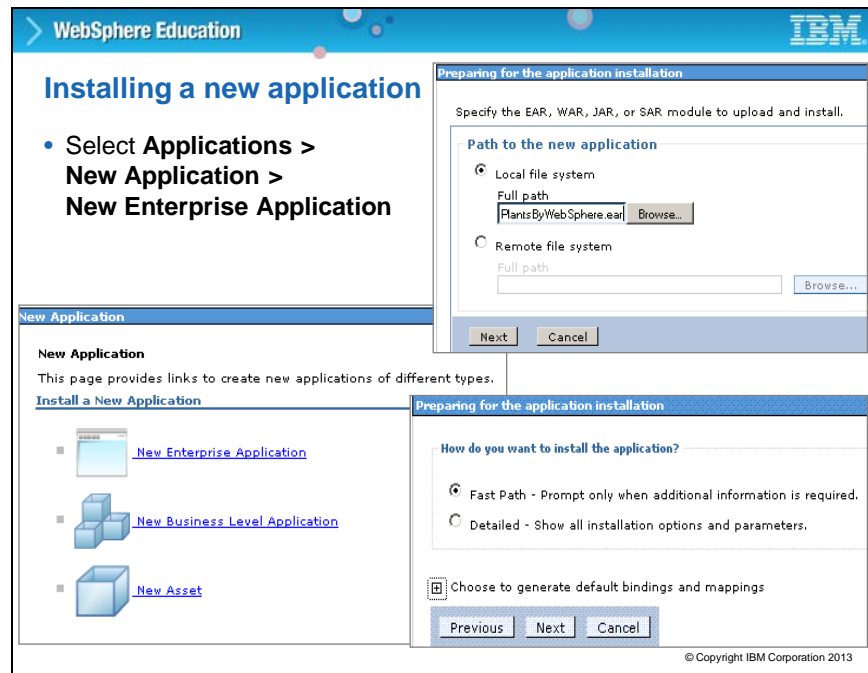
Name	Value
* Driver type	4
* Database name	PLANTS
* Server name	dbhost
* Port number	50000

☒ Use this data source in container managed persistence (CMP)

© Copyright IBM Corporation 2013

As the database protocol is vendor-specific, you must use separate JDBC drivers that are vendor-supplied to connect to the database. You specify the driver type, database name, server name, and port on this screen.

Type 4 drivers are suggested if available. Type 4 drivers are pure Java drivers that translate JDBC calls to a local database protocol. One advantage of type 4 drivers is that they do not require more database client code on the application server.



Here are the screen captures for the simple method of installing an application.

To start the application installation wizard, select **Applications > New application > New enterprise application**. Specify the path to the EAR file on the local or remote file system and click Next. You can choose the fast path, which prompts you only when more information is required, or you can choose the detailed path where all of the installation options are shown. You can also choose to generate default bindings and mappings.

Slide 12

WebSphere Education **IBM**

Example of fast path installation

Install New Application

Specify options for installing enterprise applications and modules.

Step 1: Select installation options

Step 2: Map modules to servers

Step 3: Metadata for modules

Step 4: Summary

Select installation options

Specify the various options that are available for your application.

☐ Precompile JavaServer Pages files

Directory to install application:

☒ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name:

☒ Create MBeans for resources

☐ Override class reloading settings for Web and EJB modules

Reload interval in seconds:

☐ Deploy Web services


Validate Input: off/warn/fail

warn

© Copyright IBM Corporation 2013

This slide displays an example of what the options look like if you choose the fast path for installing an application. Notice that few steps are listed in the left column. You can skip ahead through these steps by clicking the links. Many options are presented on this page. For example, you can select the box to precompile JSP pages. You can select the box to process embedded configuration settings, which creates the application-scoped resources that are defined in an enhanced EAR file. You can also choose to ignore these settings by leaving that box clear.

Each module of a deployed application must be mapped to one or more target servers. The target server can be an application server, a cluster of application servers, or a web server.

WebSphere Education 

Example of detailed installation

Summary step →

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Metadata for modules

Step 4: Summary

Summary	
Summary of installation options	
Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	PlantsByWebSphere
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\d =755#.*\s =755#.*\a=755#.*\s =755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No

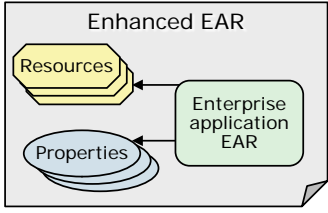
© Copyright IBM Corporation 2013

This screen capture shows the summary of information just before the application is installed. It is not important that you read all of the options on this slide, just that you understand the purpose of this step.

WebSphere Education

Enhanced EAR

- Enterprise archive that contains Java EE artifacts plus resource information necessary to install on WebSphere Application Server
 - JDBC resources (data sources)
 - Class loader
 - JAAS authentication aliases
 - Shared libraries
 - Virtual host information
- Benefits: improved productivity
 - Application resources and properties come with the application
 - Application installation process creates the necessary resources within the server or cluster
 - Moving application from one server to another also moves the resources
- Support integrated with the IBM Rational Development and Assembly and Deployment tools
 - Found on Deployment page of application deployment descriptor
- **Warning:** Can possibly cause problems if unintended application scoped resources are used in production
 - Enhancements can be removed or ignored during application installation



The diagram shows a box labeled 'Enhanced EAR'. Inside, there is a yellow hexagon labeled 'Resources' and a green rounded rectangle labeled 'Enterprise application EAR'. Below the 'Resources' hexagon is a blue oval labeled 'Properties'. Arrows point from the 'Enterprise application EAR' box to both the 'Resources' hexagon and the 'Properties' oval.

© Copyright IBM Corporation 2013

Application-scoped resources are resources that are scoped at the application level, rather than at the cell, node, or server level. An enhanced EAR file contains a simple embedded configuration archive. An enhanced EAR is where you can define resources for an application; for example, JDBC resources, class loaders, virtual hosts, and other items, so that all the information needed to deploy the application is included. In this way, you can deploy an application and its required resource definitions in one shot, and if you move the application, the resource definitions move with it. The benefit is that it can make deploying applications easier. However, it can also cause problems if the application-scoped resources do not coincide with the intended runtime environment. In this case, application scoped resources can be removed before deployment, or ignored during application installation, and the needed resources can be defined at the appropriate scope.

Enhanced EAR files are specific to IBM WebSphere and are not part of Java EE. However, artifacts that are defined within an enhanced EAR are ignored if this EAR file is installed on an application server other than WebSphere.

Developers or administrators can define resources and properties within an enterprise application in tools and import or export the enhanced EAR file. Settings are defined in IBM tools and are stored in the deployment.xml file, which is packaged with the EAR file. Some resources still must be defined in the application server, for example, JMS and JavaMail.

Slide 15

WebSphere Education

Removing enhancements

Step 1: Select installation options

Select installation options

Specify the various options that are available for your application.

- ☐ Precompile JavaServer Pages files
- Directory to install application
- ☒ Distribute application
- ☐ Use Binary Configuration
- ☐ Deploy enterprise beans
- Application name:
- ☒ Create MBeans for resources
- ☐ Override class reloading settings for Web and EJB modules
- Reload interval in seconds:
- ☐ Deploy Web services
- Validate Input off/warn/fail:
- ☒ Process embedded configuration

Resources can be ignored

- Remove enhancements from EAR before deploying (preferred)
- Clear **Process embedded configurations**
- Is prechecked only if there are enhancements

Resources can be viewed, but not through the normal screens

- Click **Application scoped resources** under the enterprise application

Enterprise Applications

Enterprise Applications > TradeApplication > Application scoped resources

Use this page to view the resources that are defined by the enhanced EAR within this application.

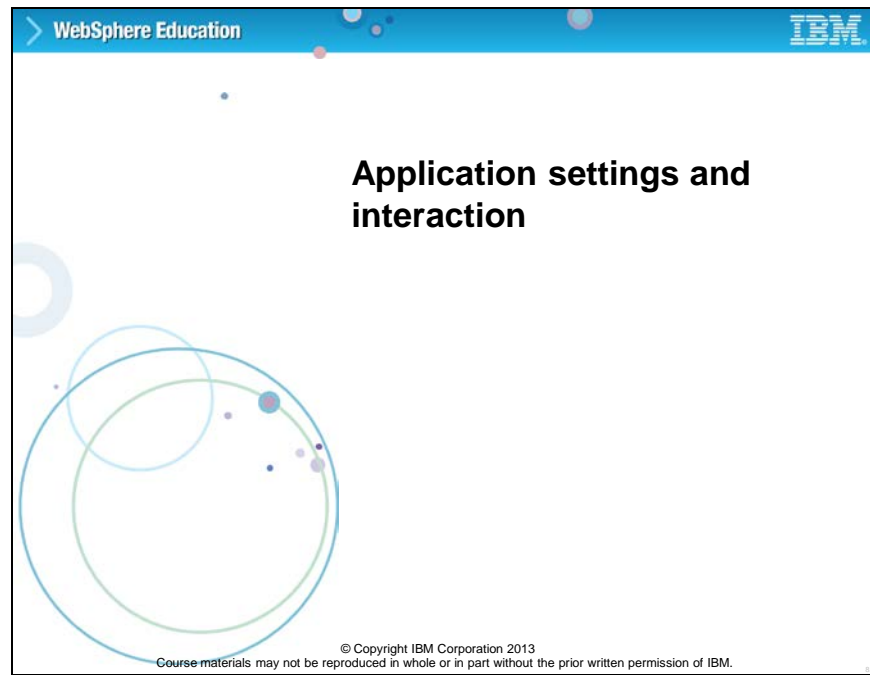
Preferences

Name	JNDI name	Resource type	Provider	Description
Trade	jdbc/tradedts	DataSource	Trade	Trade Datasource

© Copyright IBM Corporation 2013

There are several methods to ignore resources in an enhanced EAR file. The preferred method is to remove enhancements from the EAR before deploying. Here is a screen capture of the administrative console for installing an application where you can choose to ignore application-scoped resources. If the box Process embedded configuration is clear, the application-scoped resources are ignored. By default, the box is selected if the application EAR file is an enhanced EAR file.

You can also view the application-scoped resources for installed applications in the console under **Enterprise applications > application name > application scoped resources**.



Topic: Application settings and interaction. In this topic, you learn about more application settings and how you can interact with an installed application.

Slide 17

WebSphere Education **IBM**

Starting an application

- After the application is installed, you can select from a number of options to manage the application
 - Click **Applications > Application types > WebSphere enterprise applications**

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Preferences

Start Stop Install Uninstall Update Rollout Update Remove File Export

Select Name Application Status

You can administer the following resources:

Select	Name	Application Status
<input type="checkbox"/>	DefaultApplication	➔
<input type="checkbox"/>	IBMUTC	➔
<input checked="" type="checkbox"/>	PlantsByWebSphere	✖

© Copyright IBM Corporation 2013

After the application is installed, you can see it listed on the Enterprise Applications page by going to **Applications > Applications types > WebSphere enterprise applications**.

To start the application you installed, select the box beside it, and click **Start**. The application status symbol changes from a red X to a green arrow.

The other actions available are:

- Stop - Stops the selected application.
- Install - Displays the application installation page.
- Uninstall - Deletes a deployed application from the WebSphere Application Server configuration repository. It also deletes application binary files from the file system.
- Update - Replaces an application that is deployed on a server with an updated application. As part of the updating, you can complete steps on the Preparing for application installation and Update application pages.
- Rollout Update - Sequentially updates an application that is installed on multiple cluster members across a cluster.
- Remove File - Removes the specified file from the selected application or module.

Slide 18

WebSphere Education

Application update

Specify the EAR, WAR, JAR, RAR, or SAR module to upload and update.

Application to be updated:
PlantsByWebSphere

Application update options

☒ Replace the entire application
Upload an enterprise archive (*.ear) to replace the entire installed application.

Specify the path to the replacement ear file.

☒ Local file system
Full path
ps\PlantsByWebSphere.ear

☐ Remote file system
Full path

☐ Replace or add a single module
If the path to the new module matches an existing path to a module in the installed application, the new module replaces the existing module. If the path to the module does not exist in the installed application, the new module is added to the application.

☐ Replace or add a single file
If the path to the new file matches an existing path to a file in the installed application, the new file replaces the existing file. If the path to the file does not exist in the installed application, the new file is added to the application.

☐ Replace, add, or delete multiple files
Use a compressed file format such as .zip or .gzip. The compressed file is unzipped into the

You can update the full application, a single module, a single file, or part of the application

© Copyright IBM Corporation 2013

The Update option displays a page that helps you update application files that are deployed in the cell. You can update the full application, a single module, a single file, or part of the application. If a new file or module has the same relative path as a file or module that exists on the server, the new file or module replaces the existing file or module. If the new file or module does not exist on the server, it is added to the deployed application. After replacement of a full application, the product uninstalls the old application. After replacement of a module, file, or partial application, the product removes the old installed module, file, or partial application from the installed application. If a changed application or module is deployed on a cluster, changes are made to all cluster members of the cluster on which the application or module is deployed.

Click **Rollout Update** on the Enterprise applications page to propagate the changed configuration on all members of the cluster on which the application or module is deployed. Rollout update sequentially updates the configuration on the nodes that contain cluster members.

Slide 19

WebSphere Education

Other application configuration settings

Enterprise Applications > PlantsByWebSphere

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration

- Click **Applications > Application Types > WebSphere enterprise applications > application_name**

General Properties

- Name: PlantsByWebSphere
- Application reference validation: Issue warnings

Detail Properties

- Target specific application status
- Startup behavior
- Application binaries
- Class loading and update detection
- Request dispatcher properties
- Security role to user/group mapping
- JASPI provider
- Custom properties
- View Deployment Descriptor
- Last participant support extension

References

- Resource references
- EJB references

Modules

- Manage Modules
- Metadata for modules
- Display module build Ids

Web Module Properties

- Session management
- Context Root For Web Modules
- JSP and JSF options
- Virtual hosts

Enterprise Java Bean Properties

- Default messaging provider references
- Bind EJB Business
- EJB JNDI names

Client Module Properties

- Client module deployment mode

Database Profiles

- SQL profiles and pureQuery bind files

Annotations:

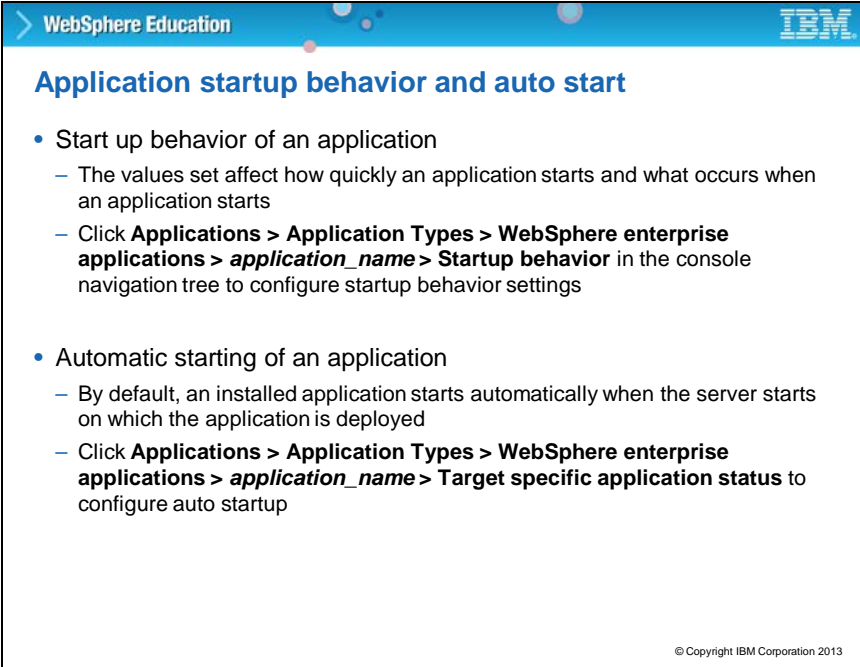
- Enable or disable automatic start
- Configure startup behavior
- View deployment descriptor
- New options:
 - Metadata for modules
 - Manage modules

© Copyright IBM Corporation 2013

This screen capture shows a broad overview of the metadata for an enterprise application that can be modified or configured through the administrative console.

Click **Applications > Application Types > WebSphere enterprise applications > application_name** to change other application settings to get to this page.

On this page, you can enable or disable automatic start. You can also configure startup behavior, view the deployment descriptor, and initiate other tasks. One of the new features is the capability to view metadata for modules and manage modules for an application.

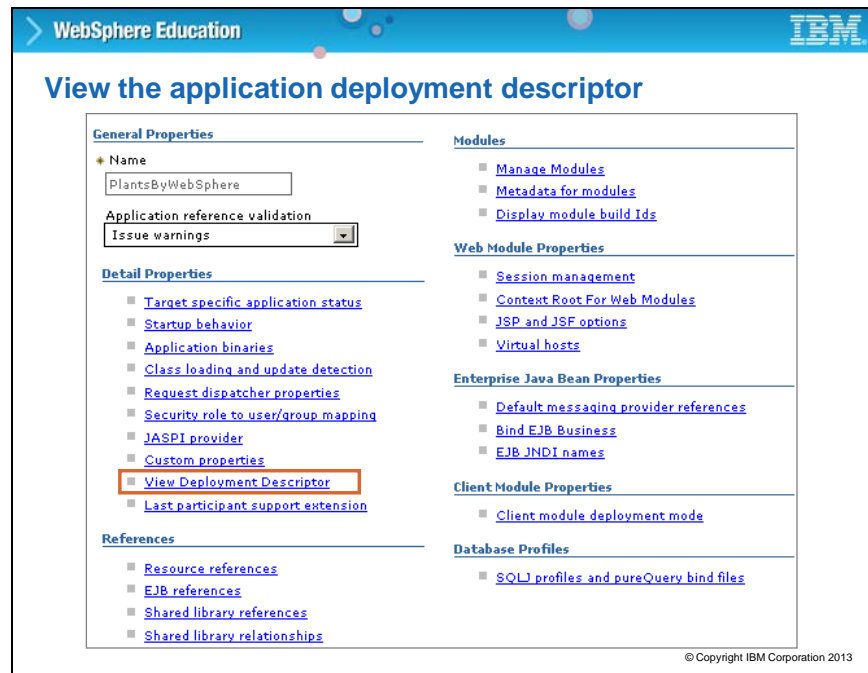


The slide is titled "Application startup behavior and auto start" and is part of a "WebSphere Education" presentation. It contains two main bullet points. The first bullet point, "Start up behavior of an application", has two sub-points: one explaining that values affect startup speed and occurrence, and another providing a navigation path in the console: **Applications > Application Types > WebSphere enterprise applications > application_name > Startup behavior**. The second bullet point, "Automatic starting of an application", also has two sub-points: one stating the default behavior (automatic startup with the server) and another providing a navigation path: **Applications > Application Types > WebSphere enterprise applications > application_name > Target specific application status**. A copyright notice for IBM Corporation 2013 is at the bottom right.

- Start up behavior of an application
 - The values set affect how quickly an application starts and what occurs when an application starts
 - Click **Applications > Application Types > WebSphere enterprise applications > application_name > Startup behavior** in the console navigation tree to configure startup behavior settings
- Automatic starting of an application
 - By default, an installed application starts automatically when the server starts on which the application is deployed
 - Click **Applications > Application Types > WebSphere enterprise applications > application_name > Target specific application status** to configure auto startup

© Copyright IBM Corporation 2013

For the startup behavior of an application, the values set affect how quickly an application starts and what occurs when an application starts. By default, an application starts when its parent server starts.



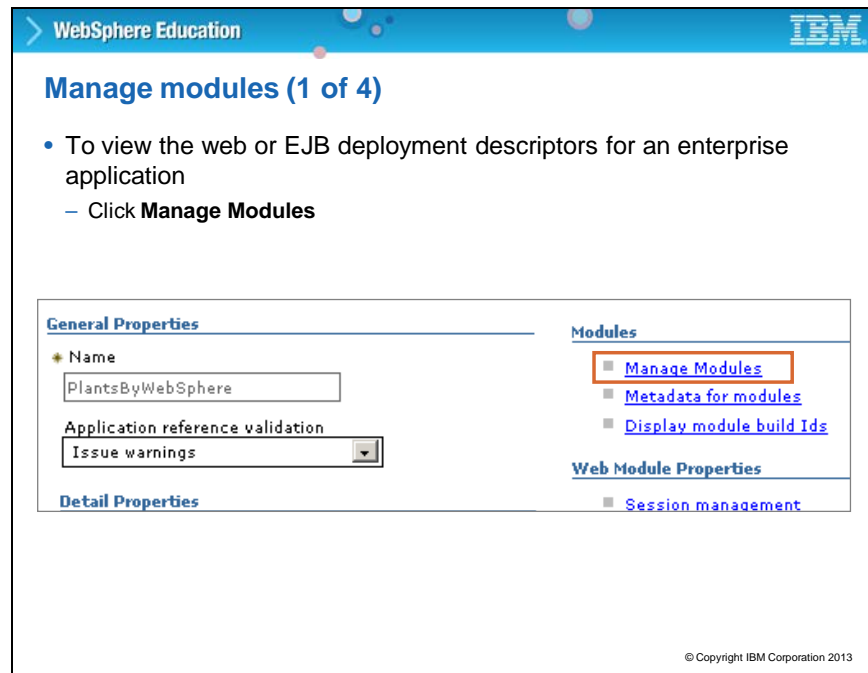
You can view the application.xml deployment descriptor for the installed application from the administrative console. Select **Enterprise applications > application_name** and then click **View Deployment Descriptor** under **Detail Properties**.

The screenshot shows the 'Enterprise Applications' window in the WebSphere Education interface. The breadcrumb navigation is 'Enterprise Applications > PlantsByWebSphere > Deployment Descriptor'. Below the navigation, there is a text instruction: 'Expand and collapse the application deployment descriptor data to view.' and two buttons: 'Expand All' and 'Collapse All'. The XML content is displayed with expandable/collapsible icons. The XML structure is as follows:

```
<application version="5"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/application_5.xsd" >
  <module>
    <web>
      <web-uri> PlantsByWebSphereWeb.war </web-uri>
      <context-root> PlantsByWebSphere </context-root>
    </web>
  </module>
  <security-role>
    <description> Samples Administrator </description>
    <role-name> SampAdmin </role-name>
  </security-role>
</application>
```

© Copyright IBM Corporation 2013

This slide displays the result of selecting View Deployment Descriptor (as seen on the previous slide) for the PlantsByWebSphere application, which is the application deployment descriptor called application.xml. In the example, you can see the context root for the PlantsByWebSphere application. In addition, a security role is seen for this Java Enterprise application.



These next four slides show the steps for viewing the deployment descriptor of an EJB JAR file (EJB module). You can view the web or EJB deployment descriptors for the installed application from the administrative console. Select **Enterprise applications > *application_name***; then click **Manage Modules** under **Modules**. In this example, you see the deployment descriptors for the PlantsByWebSphere enterprise application.

Slide 24

WebSphere Education

Manage modules (2 of 4)

[Enterprise Applications](#) > [PlantsByWebSphere](#) > Manage Modules

Manage Modules

Specify targets such as application servers or clusters of application servers where you want to install the module. Modules can be installed on the same application server or dispersed among several application servers. Also, specify routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated through.

Clusters and servers:

WebSphere:cell=was8host01Node01Cell,node=was8host01Node01,server=server1

Select	Module	URI	Module Type	Server
<input type="checkbox"/>	PlantsByWebSphere	PlantsByWebSphereWeb.war,WEB-INF/web.xml	Web Module	WebSphere:cell=was8host01Node01

© Copyright IBM Corporation 2013

This screen capture shows that the PlantsByWebSphere application has one web module and no EJB module. Click the **module_name** link under the Module column to display the properties of the module.

WebSphere Education IBM

Manage modules (3 of 4)

[Enterprise Applications](#) > [PlantsByWebSphere](#) > [Manage Modules](#) > [PlantsByWebSphereWeb.war](#)

Use this page to configure an instance of a deployed web module in the application. This page contains deployment-specific information for a web module and session management settings.

Configuration

General Properties	Additional Properties
<p>✦ URI <input type="text" value="PlantsByWebSphereWeb.war"/></p> <p>Alternate deployment descriptor <input type="text"/></p> <p>✦ Starting weight <input type="text" value="10000"/></p> <p>✦ Class loader order <input type="text" value="Classes loaded with parent class loader first"/></p>	<ul style="list-style-type: none">■ View Module Class Loader■ Custom properties■ Target specific application status■ View EJB Deployment Descriptor■ View Web Deployment Descriptor■ Session Management

© Copyright IBM Corporation 2013

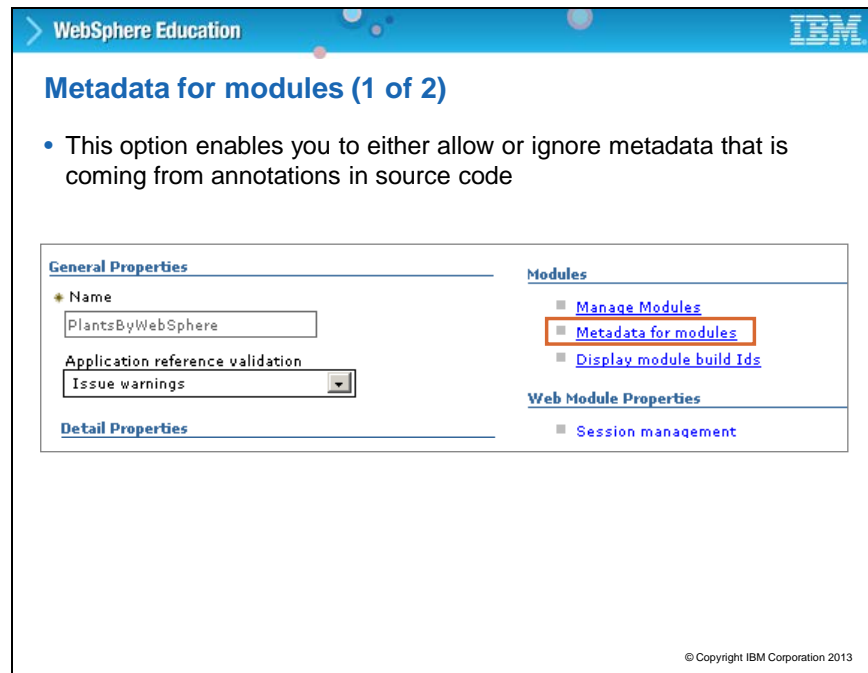
You can now view the ejb-jar.xml deployment descriptor of the PlantsByWebSphere.war module, of the PlantsByWebSphere application, by clicking **View EJB Deployment Descriptor** under **Additional Properties**. In Java EE 6, EJBs can be stored within a WAR file.

The screenshot shows the 'Manage modules (4 of 4)' page in the WebSphere Education interface. The breadcrumb trail is: Enterprise Applications > PlantsByWebSphere > Manage Modules > PlantsByWebSphereWeb.war > Deployment Descriptor. Below the breadcrumb, there is a text instruction: 'Expand and collapse the application deployment descriptor data to view.' and two buttons: 'Expand All' and 'Collapse All'. The main content area displays an XML snippet of the deployment descriptor for the 'CatalogMgr' module. The XML is as follows:

```
<ejb-jar id="ejb-jar_ID" version="3.1" metadata-complete="false" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/ ejb-jar_3_1.xsd" >
  <enterprise-beans>
    <session>
      <ejb-name> CatalogMgr</ejb-name>
      <mapped-name/>
      <ejb-class> com.ibm.websphere.samples.pbw.ejb.CatalogMgr</ejb-class>
      <session-type> Stateless</session-type>
      <init-on-startup> False</init-on-startup>
      <concurrency-management-type> Container</concurrency-management-type>
      <local-bean/>
      <persistence-context-ref>
        <persistence-context-ref-name> com.ibm.websphere.samples.pbw.ejb.CatalogMgr/em</persistence-context-ref-name>
        <persistence-unit-name> PSBW</persistence-unit-name>
        <persistence-context-type> Transaction</persistence-context-type>
        <injection-target>
          <injection-target-class> com.ibm.websphere.samples.pbw.ejb.CatalogMgr</injection-target-class>
          <injection-target-name> em</injection-target-name>
        </injection-target>
      </persistence-context-ref>
    </session>
  </enterprise-beans>
</ejb-jar>
```

At the bottom right of the page, there is a copyright notice: © Copyright IBM Corporation 2013.

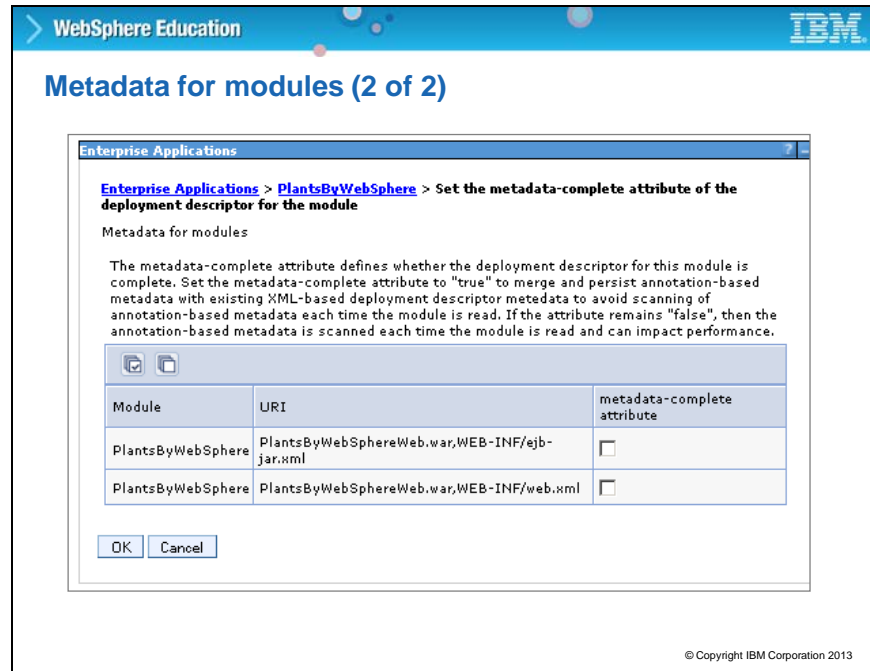
This screen capture shows the ejb-jar.xml deployment descriptor for the PlantsByWebSphere application. The part of the deployment descriptor inside the frame is information that WebSphere Application Server generates from annotations in the PlantsByWebSphere.war file and inserted in the ejb-jar.xml file when the PlantsByWebSphere application was deployed.



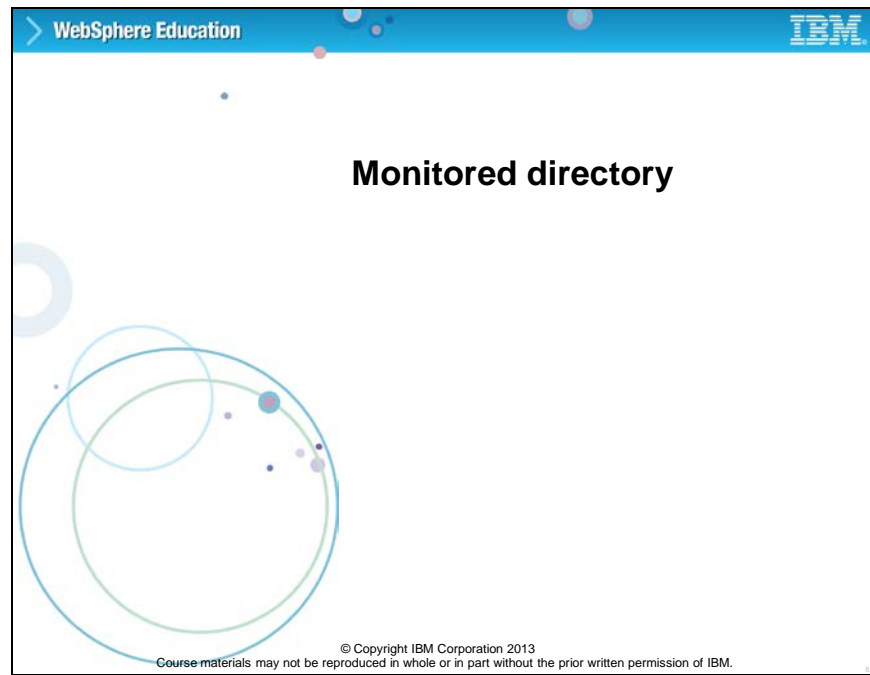
This screen capture shows where you can control the metadata for the application modules. Select **Enterprise applications > application_name**, and click **Metadata for modules** under **Modules**.

Using this option, you can specify whether the application metadata can be specified in annotations in code together with metadata in deployment descriptors, or whether the metadata can be specified through deployment descriptors. This option is useful because the deployer, or administrator, of the application can change some of the application metadata. For example, an administrator can change the data sources that the enterprise application uses. The administrator can change deployment descriptors only, since in many cases, the application source code is not packaged with the EAR file.


It is not necessarily a recommended practice to set metadata-complete to *true* if your programmers intend to use annotation-based programming techniques. The technique that you adopt can be based on personal preference.



This screen capture displays the metadata-complete attributes for PlantsByWebSphere that are not checked. When the attributes are checked, all annotation metadata is written to the appropriate deployment descriptor and afterward all annotations in the enterprise application are ignored.

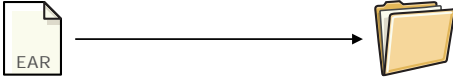


Topic: Monitored directory. In this topic, you learn about the feature called “monitored directory”, which is also known as “drag-and-drop.”

WebSphere Education 

Overview of monitored directory deployment

- A simple and fast way to install, update, and uninstall applications without:
 - The administrative console
 - Rational Application Developer
 - wsadmin
 - A specially configured environment
- Tasks can be accomplished by copying archive files in or out of a monitored directory
 - The application must be an EAR, JAR, WAR, or SAR



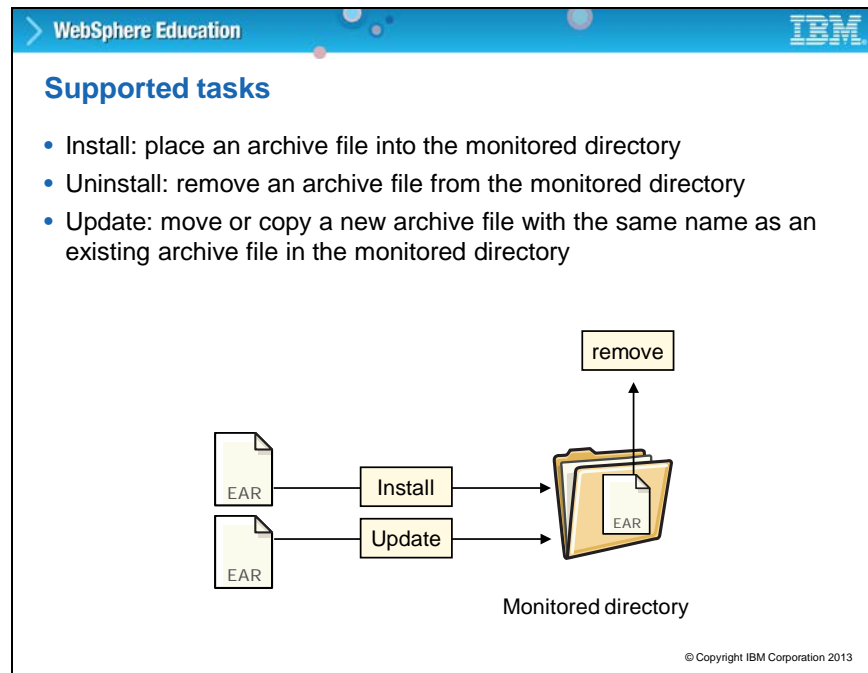
© Copyright IBM Corporation 2013

The basic definition for the monitored directory feature is “a place where the WebSphere Application Server looks for updates.” The use of this feature is as simple as copying and deleting files. There are a few more things to know before using the monitored directory feature.

A monitored directory application deployment is a simple and fast way for administrators and developers to install, update, and uninstall applications by moving archive files in or out of a monitored directory. A user who prepackages an application file with all bindings specified, can quickly deploy that application without any tools other than a running application server, or, in a network deployment environment, a deployment manager.

Monitored directory application deployment can be performed with applications packaged as enterprise archive (EAR) files, web archive (WAR) files, Java archive (JAR) files, or SIP application resources (SARs). In addition to the mentioned archive files, an application can be deployed by using a property file.

Property files are different from the other file types. They are not archives. Instead, they contain properties that describe an application, including the source archive location and its installation parameters. Monitored directory deployment with property files relies on the properties file based configuration (PFBC) feature introduced in WebSphere Application Server Version 7. Using a property file can allow a higher degree of control over the deployment than with a plain archive.



To install an application by using the monitored directory feature, use a graphical file manager or the command line to copy or move its archive file into a monitored directory. Deleting a file from a monitored directory causes uninstallation of the corresponding application.

To perform a full replacement update of a deployed application, move or copy an updated archive file with the same application name into a monitored directory. The archive file name determines the application name. Or, in case of an EAR archive, the display name determines the application name if a display name is specified in the archive.

Slide 32

WebSphere Education IBM

Enabling the monitored directory

- Disabled by default in both stand-alone federated environments
- Click **Applications > Global deployment settings**

WebSphere, software Welcome wasadmin

Cell=was8host01Cell01, Profile=Dmgr

Global deployment settings

Use this page to manage settings that apply to all applications. NOTE: application types.

Configuration

General Properties

Monitored Directory Deployment

☒ Monitor directory to automatically deploy applications

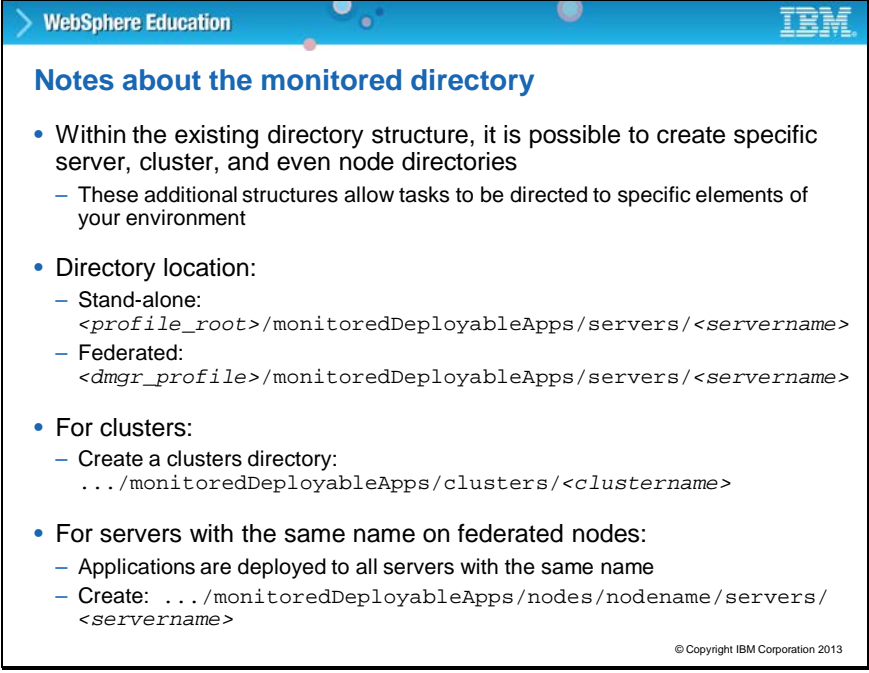
Monitored directory


Polling interval
 seconds

© Copyright IBM Corporation 2013

The monitored directory feature is not enabled in the application server after installation by default. Before using the monitored directory, enable it from the **Applications > Global deployment settings** page of the administrative console. This feature requires a path (the directory) and a polling interval (in seconds) that tells the server where to look and how often to check. Enabling the service must be done only once.

The deployment manager, or base server where the service is running, must be restarted to register any changes on the **Global deployment settings** page.



WebSphere Education 

Notes about the monitored directory

- Within the existing directory structure, it is possible to create specific server, cluster, and even node directories
 - These additional structures allow tasks to be directed to specific elements of your environment
- Directory location:
 - Stand-alone:
`<profile_root>/monitoredDeployableApps/servers/<servername>`
 - Federated:
`<dmgr_profile>/monitoredDeployableApps/servers/<servername>`
- For clusters:
 - Create a clusters directory:
`.../monitoredDeployableApps/clusters/<clustername>`
- For servers with the same name on federated nodes:
 - Applications are deployed to all servers with the same name
 - Create: `.../monitoredDeployableApps/nodes/nodename/servers/<servername>`

© Copyright IBM Corporation 2013

This slide indicates that while the monitored directory feature is defined and described as a single directory, it is composed of several subdirectories. The path to a monitored directory depends on the type of application server, and what the target of the deployment is going to be. To control where the monitored directory applications are deployed, you create more directories to represent their clusters or servers. These names must exactly match what is actually in the cell.

If you are using a stand-alone application server, then the only possible target is the server itself, and the monitored directory is automatically created if the service is enabled. For example, if the profile is called AppSrv01, and the server is named server1, the path is:
`../profiles/AppSrv01/monitoredDeployableApps/server1.`

If you are using a network deployment system, it is necessary to create the monitored directories manually. For application servers on a node that is federated with a deployment manager, you must create the monitored directories for servers under the deployment manager profile:
`../profiles/dmgrprofilename/monitoredDeployableApps/servers/ server_name.`

If multiple servers on different nodes have the same name and you want to target only one of the servers, you can specify the node and server in the path to the monitored directory. Create a directory for the node by using the node name, then servers, and finally the server_name

directory:

../profiles/dmgrprofilename/monitoredDeployableApps/nodes/node_name/servers/server_name.

For clusters, create a monitored directory under the deployment manager profile with the name of the targeted cluster: ../profiles/dmgrprofilename/monitoredDeployableApps/clusters/ .

Slide 34

WebSphere Education

IBM

Drag-and-drop properties files

- The standard drag-and-drop approach lacks the ability to do anything but the default
 - There is no ability to customize a deployment in any way
- Drag-and-drop technique also supports properties file based configuration
 - A property file can define which EAR file to install, and also configure any of the necessary attributes

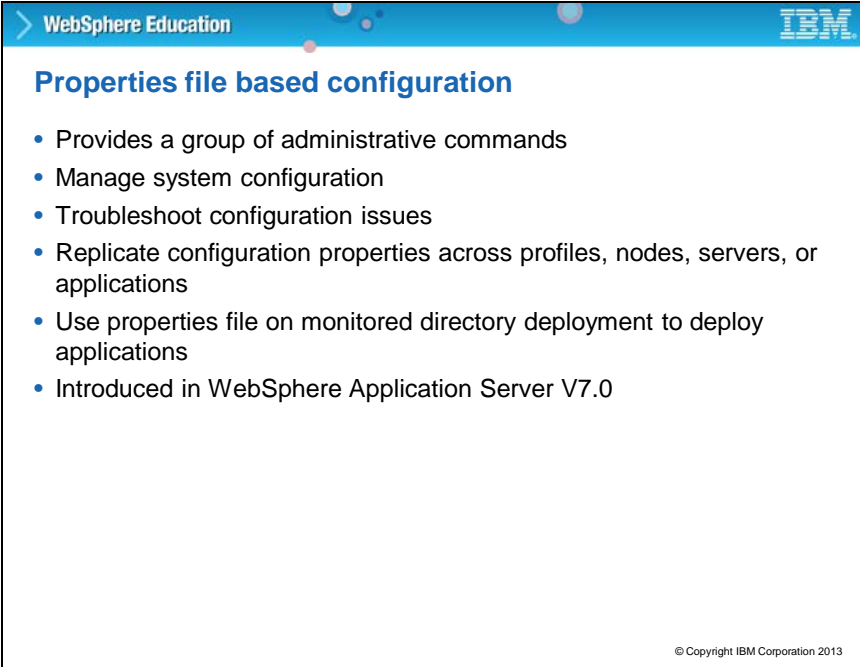
The diagram illustrates the process of deploying an application using a properties file. It shows an EAR file being dragged to a Properties file, which is then dropped into a Monitored directory. The EAR file is represented by a yellow icon with the text 'EAR'. The Properties file is represented by a white icon with the text 'Properties file'. The Monitored directory is represented by a yellow folder icon with the text 'Monitored directory'. A dashed arrow points from the EAR file to the Properties file, and a solid arrow points from the Properties file to the Monitored directory.

© Copyright IBM Corporation 2013

The standard monitored directory feature supports the ability to drag an application from one directory and drop it into another directory. But it does not allow you to customize the deployment.

To customize the deployment, you use a properties file. Monitored directory supports the ability to drag and drop a properties file it into another directory, which allows you to customize any installation settings that might be required for monitored directory installations.

Slide 35





The slide is titled "WebSphere Education" in the top left corner and features the IBM logo in the top right corner. The main title is "Properties file based configuration". Below the title is a bulleted list of six items. At the bottom right of the slide, there is a small copyright notice: "© Copyright IBM Corporation 2013".

- Provides a group of administrative commands
- Manage system configuration
- Troubleshoot configuration issues
- Replicate configuration properties across profiles, nodes, servers, or applications
- Use properties file on monitored directory deployment to deploy applications
- Introduced in WebSphere Application Server V7.0

© Copyright IBM Corporation 2013

The ability of the monitored directory to process property files was introduced with WebSphere Application Server version 7. Property files allow for a select set of administrative commands, which allow for activities such as the ability to manage the system configuration, troubleshoot configuration issues, and replicate configuration properties across profiles, nodes, servers, or applications. In addition, a property file can be used to deploy an application.



Steps to use properties file to deploy applications (1 of 2)

- Step 1: create a properties file that defines deployment options
 - Use properties file based configuration command to create an application properties file
 - Extract application properties to a file with version 7 output format

```
AdminTask.extractConfigProperties('[-propertiesFileName myApp.props -configData Deployment=MyApplication]')
```
 - Extract application properties to a file with simple output format

```
AdminTask.extractConfigProperties('[-propertiesFileName myApp.props -configData Deployment=MyApplication -options [[SimpleOutputFormat true]]]')
```
 - Note: the monitored directory installation process uses only the properties that relate to an application
 - Create the file manually

© Copyright IBM Corporation 2013

This slide displays an example of using a properties file to deploy applications. The wsadmin AdminTask.extractConfigProperties command is useful for capturing properties for use with the monitored directory feature.

Step 1 shows how to use the properties file based configuration **extractConfigProperties** command to extract application properties to a file. By default, the **extractConfigProperties** command produces output that displays all columns, including all hidden and non-hidden columns of installation tasks and task data values, in separate rows. You can also extract application properties in simple output format to display non-hidden columns of installation task data in columnName=value pairs.

You can also use the properties file examples that are documented in the information center to create an application properties file to deploy, uninstall, or update an application.

WebSphere Education IBM

Steps to use properties file to deploy applications (2 of 2)

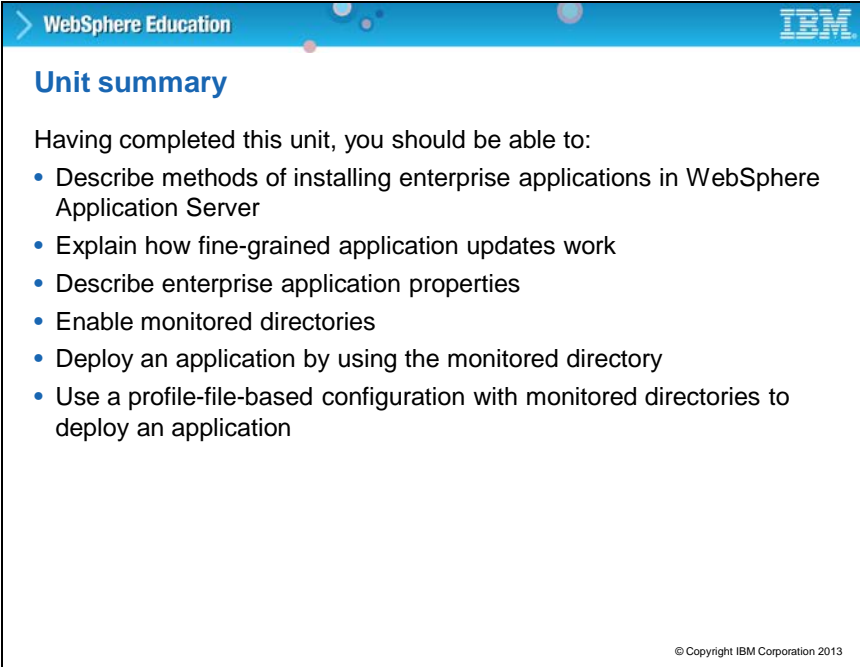
- Step 2: verify that the targeted server or cluster member is running
- Step 3: verify that monitored directory deployment is enabled
- Step 4: copy the properties file to the deploymentProperties directory

The diagram illustrates the process of copying a properties file to the deploymentProperties directory. It shows an EAR file pointing to a Properties file, which is then copied to the deploymentProperties folder. Below this, a screenshot of a file browser shows the monitoredDeployableApps directory containing clusters, deploymentProperties, and servers folders.

© Copyright IBM Corporation 2013

Before a property file can be processed, the monitored directory service must be active and the application must be running. The property file must be copied to the deploymentProperties directory for the application. For example, you must copy a property file to the cluster where the application is running for the event to succeed. You can use a file browser to drag the properties file to the monitored directory or use the operating system command to copy the properties file to the directory.

Slide 38



The slide is titled 'Unit summary' and is part of a 'WebSphere Education' presentation. It lists the following learning objectives:

- Describe methods of installing enterprise applications in WebSphere Application Server
- Explain how fine-grained application updates work
- Describe enterprise application properties
- Enable monitored directories
- Deploy an application by using the monitored directory
- Use a profile-file-based configuration with monitored directories to deploy an application

© Copyright IBM Corporation 2013

You completed this unit.

Having completed this unit, you should be able to:

- Describe methods of installing enterprise applications in WebSphere Application Server
- Explain how fine-grained application updates work
- Describe enterprise application properties
- Enable monitored directory
- Deploy an application by using the monitored directory
- Use a property-file-based configuration with monitored directories to deploy an application