



IBM Software Group

IBM® WebSphere® Application Server V7

Configuring business-level applications



@business on demand.

© 2008 IBM Corporation
Updated September 22, 2008

This presentation will cover configuration of business-level applications in WebSphere Application Server V7.

Agenda

- Business-level application (BLA) operations
- Composition unit relationships
- Session management



This presentation will first go over business-level application operations, followed by composition unit relationships, and session management.

Section

Business-level application operations



This section will cover configuration operations for business-level applications.

Business-level application operations

- Configure business-level applications using:
 - ▶ The administrative console
 - ▶ AdminTask commands for business-level applications
- Asset commands
 - ▶ importAsset, listAssets, deleteAsset, exportAsset, viewAsset, editAsset, updateAsset (full update or fine-grained update)
- Composition unit commands
 - ▶ addCompUnit, listCompUnits, deleteCompUnit, viewCompUnit, editCompUnit
- Business-level application commands
 - ▶ createEmptyBLA, listBLAs, viewBLA, deleteBLA
 - ▶ Start and stop are the only predefined control operations



Both the administrative console and wsadmin have been enhanced with support for working with business-level applications. This slide shows several new AdminTask commands in wsadmin that can be used for creating, displaying, and deleting assets, composition units, and business-level applications. Wsadmin also has commands for starting and stopping business-level applications.

Example: Creating an application

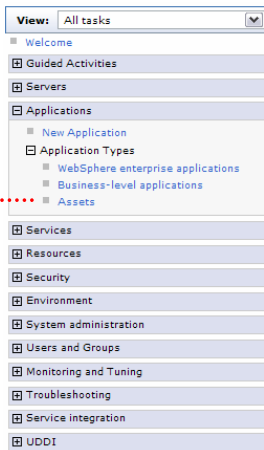
- Typical steps for creating a business-level application are:
 - ▶ Import an asset
 - ▶ Create an empty business-level application
 - ▶ Add the asset to the business-level application
 - ▶ Start the business-level application



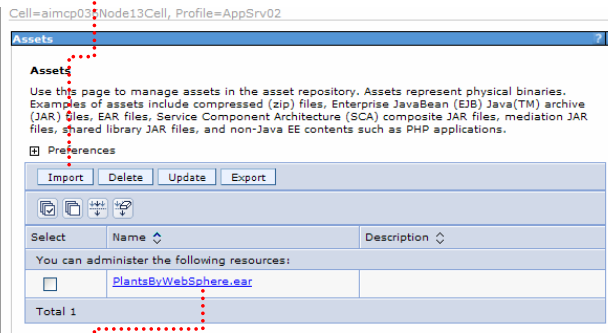
To create a business-level application, first import an asset that you want to be a part of the application. Then create a new business-level application, and add the asset to it. You then have a business-level application that has one asset and can be started and stopped using the controls for business-level applications.

Example: Import an asset

1 Expand **Applications > Application Types > Assets**



2 On the **Import** asset page, provide a path to the asset and continue through the panels, accepting the defaults; **Save** your changes



3 The new asset is displayed in the list of assets in the asset repository

This image shows the administrative console panel that lets you work with assets. After navigating to the “assets” panel, you can import, update, export, or delete assets. Always remember to save your changes, as well.

Example: Create an empty application

1 Expand **Applications > Application Types > Business-level applications**

2 On the **New** application page, provide a name for the business-level application, then **Apply** and **Save** your changes

Cell=aimcp036Node11Cell, Profile=stps1

Business-level applications

Use this page to manage business-level applications. A business-level application is a configuration that represents any artifacts that the application needs to run. Artifacts typically include Java(TM) Platform, Enterprise Edition (Java EE) applications or modules, shared libraries, data files, or other business-level applications.

Preferences

Start Stop New Delete

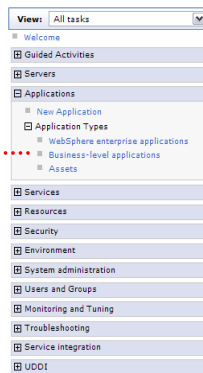
Select	Name	Description	Status
You can administer the following resources:			
<input type="checkbox"/>	DefaultApplication		➔
<input type="checkbox"/>	MySampleBLA		⊗
<input type="checkbox"/>	lvApp		➔
<input type="checkbox"/>	Quarv		➔
Total 4			

3 The new business-level application will appear as unavailable because it is empty

To create a new, empty business-level application, navigate to “business-level applications” in the “application types” sub-menu. In addition to being able to create a business-level application, this is also where you can start and stop them. In this example, “MySampleBLA” is not available to be started, since it is newly created and does not contain any assets.

Example: Add the asset to the application

- 1** Expand **Applications > Application Types > Business-level applications** and select the name of your business-level application



Cell=aimcp036Node13Cell, Profile=AppSrv02

Business-level applications

Business-level applications > MySampleBLA

Use this page to manage the composition units in the business-level application.

General Properties

Name
MySampleBLA

Description
This is a sample business-level application

Deployed assets

Add + Delete -

Add Asset	Add Shared Library	Description	Type	Status	⌵
None					

Business-level applications

Add + Delete -

Select	Name	Description	Status	⌵
None				

OK Cancel

2

Use this menu to add asset or shared library composition units to the application

Use this menu to add business-level application composition units to the application

3

Step through the panels to complete the composition unit configuration and save your changes



If you click on the name of a business-level application, you will see a panel that allows you to add assets, shared-libraries, or other business-level applications to the application.

Example: Start the application

1 Expand **Applications > Application Types > Business-level applications**

2 Select the check box next to your business application and click the **Start** button

3 The status indicator will switch to green when all the composition units in the application have started

The screenshot shows a web interface titled "Business-level applications". At the top, there is a message box that says "MySampleBLA started successfully." Below this, there is a section for "Business-level applications" with a description: "Use this page to manage business-level applications. A business-level application is a configuration that represents any artifacts that the application needs to run. Artifacts typically include Java(TM) Platform, Enterprise Edition (Java EE) applications or modules, shared libraries, data files, or other business-level applications." There are "Start", "Stop", "New", and "Delete" buttons. Below these is a table of applications:

Select	Name	Description	Status
You can administer the following resources:			
<input type="checkbox"/>	DefaultApplication		✖
<input checked="" type="checkbox"/>	MySampleBLA		➔
<input type="checkbox"/>	lvApp		➔
<input type="checkbox"/>	query		➔
Total 4			

Once you have added assets to your business-level application and saved your changes, you will be able to start and stop your business-level application, with controls similar to the ones used for Java EE enterprise applications.

Example: Creation process with scripting

Description	Command
Create an asset	<code>asset1 = AdminTask.importAsset('-source \ears\asset1.zip')</code>
Verify that the asset was created with the required settings	<code>AdminTask.listAssets('-includeDescription true -includeDeplUnit true')</code>
Create an empty business-level application	<code>myBLA = AdminTask.createEmptyBLA('-name myBLA -description "BLA that contains asset1"')</code>
Verify that the empty application was created	<code>AdminTask.listBLAs()</code>



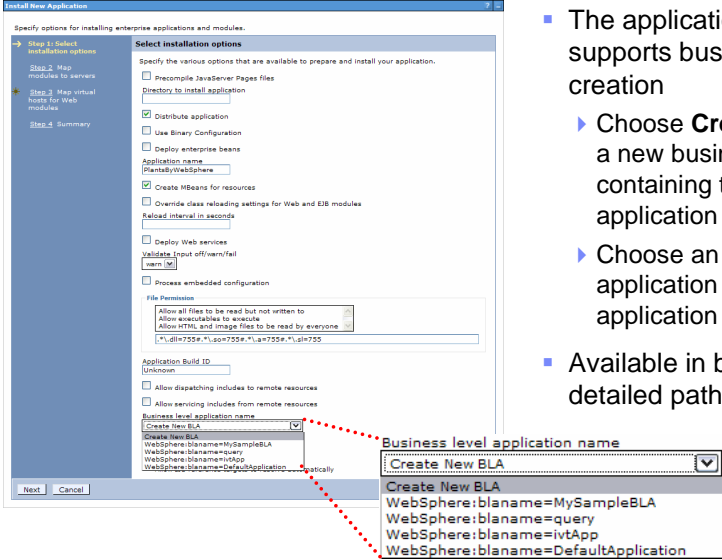
This table shows the equivalent wsadmin commands for creating an asset and a business-level application, and then using the “list” commands to verify that they were created.

Example: Creation process with scripting

Description	Command
Add the asset, as a composition unit, to the business-level application	<code>AdminTask.addCompUnit('-blaID myBLA -cuSourceID asset1.zip -CUOptions [[.* .* compositionUnit1 "composition unit that is backed by asset1" 1 false DEFAULT]] -MapTargets [[.* server1]] -ActivationPlanOptions [[.* specname=actplan0+specname=actplan1]]')</code>
Save configuration changes	<code>AdminConfig.save()</code>
Synchronize nodes	<code>AdminNodeManagement.syncActiveNodes()</code>
Start the business-level application	<code>AdminTask.startBLA('-blaID myBLA')</code>

You can then add the asset to the business-level application, and after saving changes and synchronizing, start the application, with the wsadmin commands shown here.

Example: Creation during installation



The screenshot shows the 'Install New Application' wizard at Step 1: Select installation options. The 'Create New BLA' dialog box is open, showing a list of business-level application names to be added to the installation. The list includes:

- WebSphere:blaname=MySampleBLA
- WebSphere:blaname=query
- WebSphere:blaname=ivtApp
- WebSphere:blaname=DefaultApplication

The dialog box also has a 'Create New BLA' button and a 'Business level application name' label. The main wizard window shows various installation options, including 'Distribute application', 'Use Binary Configuration', 'Deploy enterprise beans', 'Create WBeans for resources', and 'File Permission'.

- The application installation wizard supports business-level application creation
 - ▶ Choose **Create New BLA** to create a new business-level application containing the enterprise application (default)
 - ▶ Choose an existing business-level application to add the enterprise application to it
- Available in both fast path and detailed path installation

You can also create a business-level application when installing an enterprise application. The highlighted menu lets you add an enterprise application to an existing business-level application or create a new one, directly from the installation wizard for enterprise applications. This option is available in both the “fast-path” and “detailed” installation wizards.

Section

Composition unit relationships



This section will discuss composition unit relationships.

Composition unit relationships


- Relationships are used to describe shared library dependencies in business-level applications
- Can be defined in two ways:
 - ▶ When an asset is being added to the business-level application
 - Choose the **Detailed - Show all installation options and parameters** path
 - ▶ After the asset has been added, from the composition unit details page
 - Choose **Applications > Application Types > Business-level applications > application_name > asset_name > Shared library relationships**
 - ▶ Select a check box next to one of the modules and click the **Reference shared libraries** button to configure the libraries

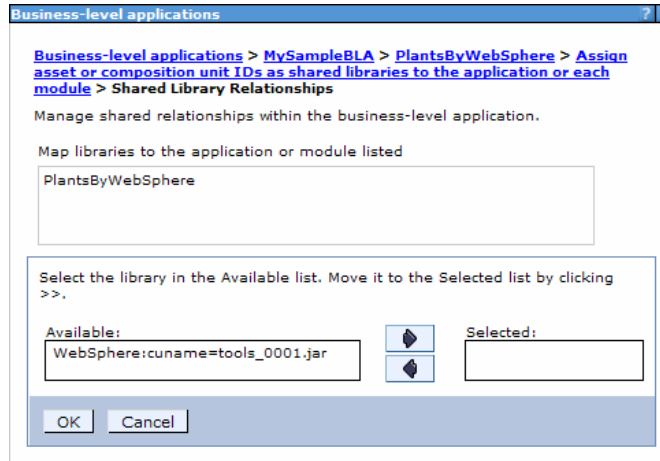
Reference shared libraries				
Select	Application	URI	Asset or composition unit IDs	Match target
<input type="checkbox"/>	PlantsByWebSphere	META-INF/application.xml		<input checked="" type="checkbox"/>
Select	Module	URI	Asset or composition unit IDs	Match target
<input type="checkbox"/>	PlantsByWebSphere Web Application	PlantsByWebSphere.war,WEB-INF/web.xml		<input checked="" type="checkbox"/>
<input type="checkbox"/>	PlantsByWebSphere Sample Gallery Web Application	PlantsGallery.war,WEB-INF/web.xml		<input checked="" type="checkbox"/>

14

Relationships are used to describe dependencies in business-level applications. You can define relationships by choosing the detailed path when adding an asset to your business-level application, or by navigating to the path shown here after installation. Select the modules that you want to associate with shared libraries, and click “reference shared libraries”.

Shared library relationship configuration

- Select a library in the **Available** list and click the arrow button () to add it to the **Selected** list



You will then be shown this panel, which displays the available shared libraries, and lets you choose them by using the arrows to add them to the “selected” list.

Describing shared library relationships

- When configuring relationships, shared libraries are described using their asset or composition unit IDs
 - ▶ If it is a composition unit, that unit must be part of the business-level application
 - ▶ If it is an asset, a composition unit is created and added to the application
- When editing an application, only composition unit IDs can be specified as shared libraries



To define a relationship with a shared library, that relationship is described using a composition unit. If a composition unit is already part of the business-level application, you can refer to that composition unit. If you define an asset as a shared library, a composition unit will be created and added to the application.

Section

Session management



This section will cover session management.

Sharing session attributes

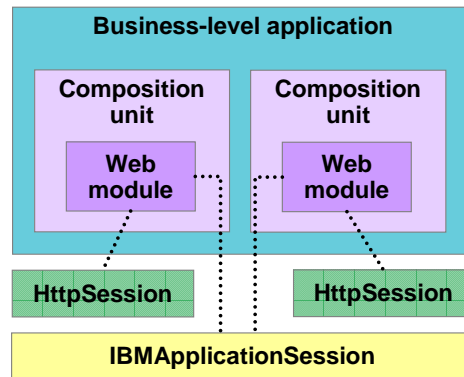
- WebSphere Application Server session management provides methods to share the HTTP session across enterprise applications
 - ▶ Shared session context
 - Assembly setting that makes attributes in the HttpSession object available to all Web modules in an enterprise application
 - Deprecated (but still available) in V7
 - ▶ IBMApplicationSession
 - New object that allows all Web modules in a business-level application to share attributes, while maintaining separate HttpSession objects
 - Recommended method in V7



There are two ways to share HttpSession data between Web applications in WebSphere Application Server V7. The shared session context is a setting that can be configured during application assembly that makes attributes in the HttpSession object available to all Web modules in an enterprise application. This method is deprecated in V7, meaning the capability may be removed in a future release. The IBMApplicationSession is a new object that can be shared by all Web modules in a business-level application. This approach is preferred, because it gives the Web modules access to a shared session object, while they also each maintain separate, non-shared HttpSession objects. Note that both of these methods are WebSphere-specific.

Business-level application session

- Example of the IBMApplicationSession configuration for a business-level application



In this example, a business-level application contains two Web modules. When this is the case, each of them has its own, unshared HTTPSession, and the IBMApplicationSession is shared by the Web modules.

Using an IBMApplicationSession

- Retrieve the HttpSession object

```
HttpSession session = request.getSession();
```

- Cast to an IBMSession object and call the getIBMApplicationSession() method

```
IBMApplicationSession appSession =  
((IBMSession) session ).getIBMApplicationSession();
```

- Use this object like an HttpSession object

To access an IBMApplicationSession object from your Web application, first retrieve the HttpSession object. Then cast that object to an IBMSession object, and call the getIBMApplicationSession() method, which will return an IBMApplicationSessionObject. You can use this object just as you normally use an HttpSession object.

Characteristics of IBMApplicationSessions

- IBMApplicationSession objects exhibit the same traits as HttpSession objects:
 - ▶ They can failover between application servers
 - ▶ Listeners can be configured which are notified of session events
 - ▶ Objects can be stored as attributes on the session and shared between Web modules



IBMApplicationSessions share the same characteristics as HTTPSessions. They support failover to other application servers, session event listeners, and object storage for sharing between Web modules.

Section

Summary



This section will summarize the presentation.

Summary

- Business-level application can be administered through the administrative console and with scripting
- Composition unit relationships define dependencies among business-level application components
- New IBMApplicationSession object provides session sharing among Web modules in a business-level application



Business-level applications are a new feature in WebSphere Application Server that enable logical grouping of multiple assets, such as enterprise applications and shared libraries. These relationships can be defined using the administrative console and wsadmin. You have seen how to create and manage assets and business-level applications, and how the new IBMApplicationSession object can be used to share data across Web modules.

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

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. 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.

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

© Copyright International Business Machines Corporation 2008. 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.