IBM WebSphere® Process Server 6.1 – Lab exercise

WebSphere Process Server V6.1: Administration

Installing service applications

What this exercise is about

This lab is intended for users who do not have experience with application management in WebSphere Process Server V6.1 environment. WebSphere Process Server utilizes the administration functionality provided by WebSphere for application deployment and management. There are several options to deploy and manage applications in WebSphere Process Server runtime:

- Deploy source code (generate and compile) into an application ready to be installed to WebSphere Process Server
  - Use WebSphere Integration Developer
  - Use command line based serviceDeploy
- Deploy application (install application) to WebSphere Process Server
  - Use the administrative console
  - Use command line wsadmin administrative tool

In this exercise you will use the '**serviceDeploy**' tool to generate the EAR file, and will then use both the administrative console and wsadmin scripting tool to install and uninstall the application.

Lab requirements

- WebSphere Process Server V6.1

What you should be able to do

At the end of this lab you should be able to:

- Use serviceDeploy to deploy (generate and compile) a simple application from the command line.
- Install and uninstall a simple application using the administrative console.
- Install and uninstall a simple application using wsadmin.

Introduction

One of the most common tasks for an administrator is to install and update applications. There are several ways to install and update applications in WebSphere Process Server. The Web browser based administrative console provides an easy to follow wizard. The command line tool wsadmin allows scripting and automation.

Developers will be most likely to use the '**serviceDeploy**' command line tool which provides an environment for automating the deployment of source code into applications.

You will go through application deployment and installation using each of these methods.

Exercise instructions

Some instructions in this lab are specific for Windows® platforms. If you run the lab on a platform other than Windows, you will need to run the appropriate commands, and use appropriate files (for example .sh in place of .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references as follows:

Reference variable	Windows location	Linux® location
<WID_HOME>	C:\IBM\WID61	
<WPS_HOME>	<WID_HOME>\runtimes\bi_v61	
<WPS_POFILE_HOME>	<WID_HOME>\pf\wps	
<LAB_FILES>	C:\Labfiles61\Admin	

Windows users: When directory locations are passed as parameters to a Java™ program such as EJBdeploy or wsadmin, you must replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\LabFiles61\ with C:/LabFiles60/.

Part 1: Create an installable EAR file using serviceDeploy

The serviceDeploy tool allows for the ability to deploy applications (generate and compile) from the command line resulting in a Service Component Architecture (SCA) application .ear file that is good for installing to the WebSphere Process Server runtime. Using the **serviceDeploy** tool, developers can automate their production builds by invoking scripts for the build process.

The archive, upon which the **serviceDeploy** tool operates, is the only mandatory parameter, and it must be the first parameter to the command. The archive can be a .jar, .zip, or .ear file. For SCA applications, which include SCA component files and possibly J2EE\index.jsp, archive these files into a .jar. For SCA applications that also include separate J2EE modules, for example Web, EJB, or connector modules, archive these files into a .zip. If a .zip if passed to serviceDeploy, it expects to find J2EE modules along with the SCA files. If none exist, you will receive an error.

Troubleshooting tips: If errors occur while using serviceDeploy, invoke serviceDeploy with optional -keep parameter to keep the temporary eclipse workspace. Note the name of the workspace, during processing of the serviceDeploy command, change to the <workspace>\.metadata subdirectory and view the .log file.

When passing a .jar to serviceDeploy that includes a top-level J2EE folder with possibly a jsp for testing purposes, remember to use the optional –freeform parameter, which copies the J2EE\xxxx files to the most reasonable J2EE staging module.

Note: The serviceDeploy archive file must be exported with '**Command line Deployment**' option using the WebSphere Integration Developer. Visit : '**Task 1: Exporting modules as serviceDeploy files**'

The following file types can be used as input to the **serviceDeploy** command:

jar: - The most useful file type for the simplest applications. The resulting .ear file contains a single .jar file and any needed generated staging modules. The .jar file must contain the service module file.

zip (Project Interchange):- You can export from WebSphere Integration Developer an archive file in project interchange format. This format is unique to the Eclipse development. The exported zip archive file must contain exactly one project with the service module file. The resulting .ear file contains any number of modules, depending upon exactly what is in the project interchange.

zip:-You can create a zip archive file containing .jar files, .war files, and .rar files. Exactly one .jar file must contain the service module file. All contained archives become members of the final exported .ear file.

ear:- You can always run the **serviceDeploy** command against an .ear file as long as exactly one .jar file in the .ear file contains a the service module file.

Complete the instructions below, to invoke the **serviceDeploy** command line tool and generate an installable SCA application file:

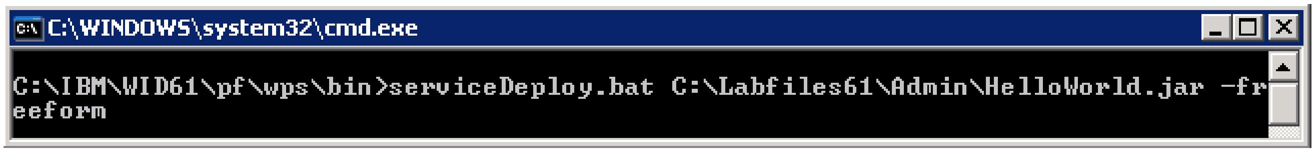
____ 1. Open a command line window and change to the bin directory of the WebSphere Process Server installation profile subdirectory

- cd <WPS_PROFILE_HOME>\bin

  Ex: **`cd  C:\IBM\WID61\pf\wps\bin`**

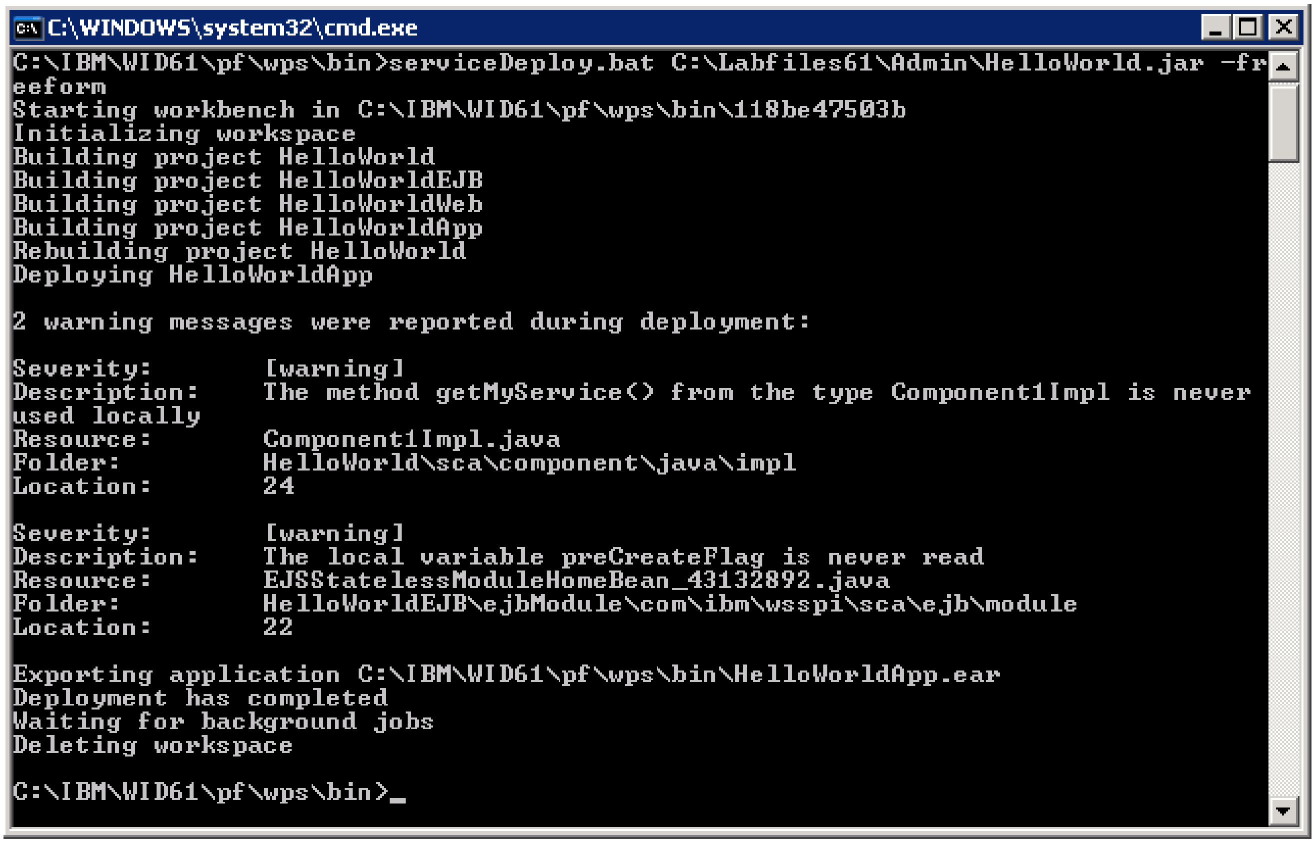
____ 2. Invoke the **serviceDeploy** command-line tool and pass the **<LAB_FILES>\HelloWorld.jar** archive and the **–freeform** as parameters.

- **`serviceDeploy <LAB_FILES>\Admin\HelloWorld.jar -freeform`**



```
C:\IBM\WID61\pf\wps\bin>serviceDeploy.bat C:\Labfiles61\Admin\HelloWorld.jar -fr
eeform
```

____ 3. Successful completion of the serviceDeploy will result in a **HelloWorldApp.ear** file in the same subdirectory from which the command was invoked as shown in the picture below:



```
C:\IBM\WID61\pf\wps\bin>serviceDeploy.bat C:\Labfiles61\Admin\HelloWorld.jar -fr
eeform
Starting workbench in C:\IBM\WID61\pf\wps\bin\118be47503b
Initializing workspace
Building project HelloWorld
Building project HelloWorldEJB
Building project HelloWorldWeb
Building project HelloWorldApp
Rebuilding project HelloWorld
Deploying HelloWorldApp

2 warning messages were reported during deployment:

Severity:       [warning]
Description:    The method getMyService() from the type Component1Impl is never
used locally
Resource:       Component1Impl.java
Folder:         HelloWorld\sca\component\java\impl
Location:       24

Severity:       [warning]
Description:    The local variable preCreateFlag is never read
Resource:       EJSStatelessModuleHomeBean_43132892.java
Folder:         HelloWorldEJB\ejbModule\com\ibm\wsspi\sca\ejb\module
Location:       22

Exporting application C:\IBM\WID61\pf\wps\bin\HelloWorldApp.ear
Deployment has completed
Waiting for background jobs
Deleting workspace

C:\IBM\WID61\pf\wps\bin>_
```

____ 4. Ensure the **HelloWorldApp.ear** file is generated (**<WPS_PROFILE_HOME>\bin**)

Part 2: Deploy an application using the administrative console

In this part of the lab, you will install the Service Component Architecture (SCA) application to WebSphere Process Server runtime using the administrative console.

____ 1. Start the WebSphere Process Server if the server is stopped

__ a. Open a Command window and change to the bin directory of the WebSphere Process Server installation directory by entering the following command from the command prompt

- `cd <WPS_PROFILE_HOME>\bin`

__ b. Start the server with the '**startServer**' command by entering the following command from the command prompt.

- `startServer.bat server1`

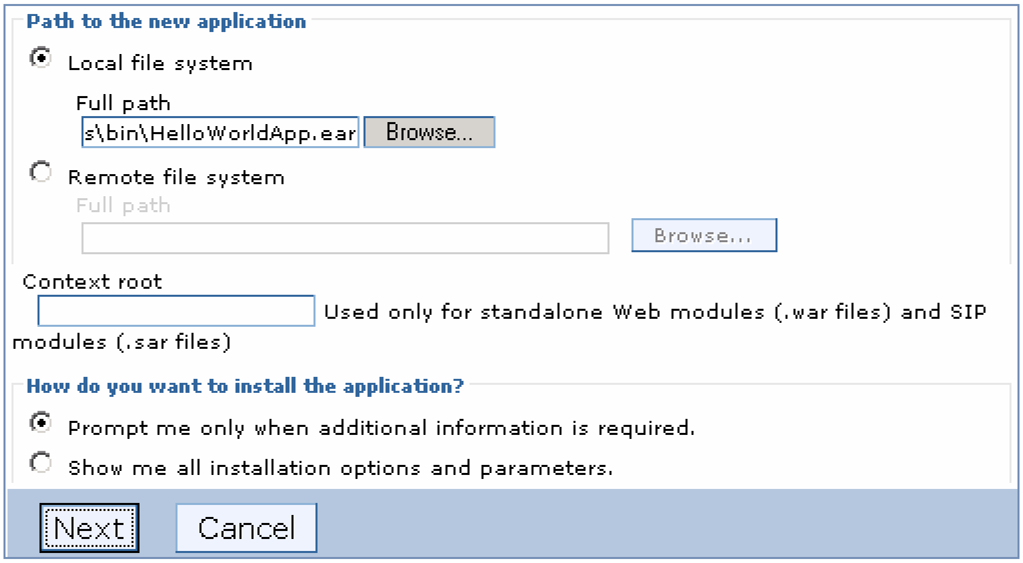
____ 2. Open the administrative console

__ a. Open a Web browser and enter http://<HOSTNAME>:9060/admin for the location

__ b. When the console appears, enter the administrative credentials and click **Log in** button

____ 3. In the left navigation pane of the administrative console, expand '**Applications**' and click the '**Install New Application**' link

____ 4. In the following panel, select the radio button for '**Local file system**' and click the **Browse** button to select the path **<WPS_PROFILE_HOME>\bin\HelloWorldApp.ear**



Path to the new application

● Local file system

Full path

s\bin\HelloWorldApp.ear

○ Remote file system

Full path

Context root

Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

● Prompt me only when additional information is required.

○ Show me all installation options and parameters.

____ 5. Click **Next**

____ 6. Click **Next** to accept the default values for all the screens until you get to the **Summary** panel. This is a simple application that does not require any specific configuration

____ 7. Click the **Finish** button on the **Install New Application** summary panel

____ 8. On the next screen, wait for the message "Application HelloWorldApp installed successfully." to display

____ 9. Save the configuration changes to the master configuration.

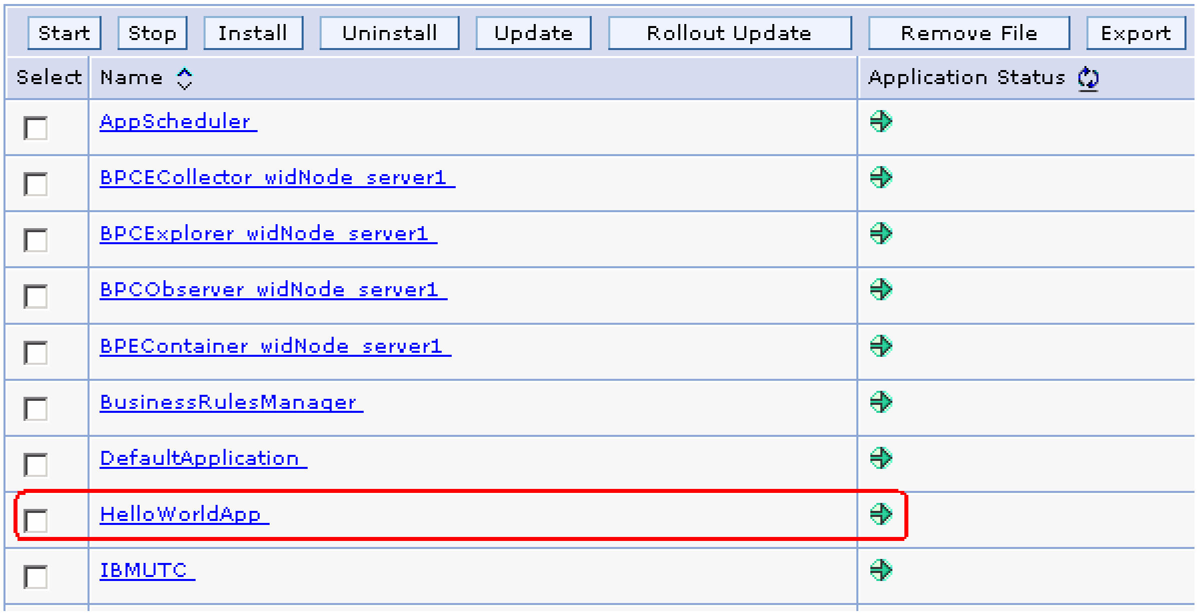
Note: Now that the application HelloWorldApp has been installed, you must start the application before you can access it.

____ 10. Start the **HelloWorldApp** application

__ a. In the left navigation pane, expand **Applications** and then click the **Enterprise Applications** link. You should see the HelloWorldApp in the list of applications and the status is in Stopped mode (✖)

__ b. Select the check box for **HelloWorldApp**

__ c. Click the **Start** button



Select	Name	Application Status
□	AppScheduler	➜
□	BPCECollector_widNode_server1	➜
□	BPCExplorer_widNode_server1	➜
□	BPCObserver_widNode_server1	➜
□	BPEContainer_widNode_server1	➜
□	BusinessRulesManager	➜
□	DefaultApplication	➜
□	HelloWorldApp	➜
□	IBMUTC	➜

____ 11. To run the application: open another browser and type in the URL: http://<HOSTNAME>:9080/HelloWorldWeb/index.jsp

NOTE: You have installed and started the HelloWorldApp and verified that you can access the application from a browser. Now, you will uninstall the application and later use wsadmin to install the application.

____ 12. Stop the **HelloWorldApp** application

__ a. Expand **Applications** on the Navigator panel and click **Enterprise Applications**. You should see the HelloWorldApp in the list of applications and the status is in Started mode (➜)

__ b. Select the check box for **HelloWorldApp**

__ c. Click the **Stop** button

____ 13. Uninstall the HelloWorldApp

__ a. Select the check box for **HelloWorldApp** on the **Enterprise Applications** panel

__ b. Click **Uninstall** button

__ c. Click **OK** to confirm in the following panel

__ d. Save to the master configuration

Part 3: Deploy an application using wsadmin client (-lang jacl)

In this part of the lab, you will use the **wsadmin** scripting client to install the Service Component Architecture (SCA) application using the **jacl** scripting language.

Note: wsadmin is a command line scripting client that provides most functionality available from the administrative console. The primary benefit of using wsadmin is that it allows for automation.

Note: The **wsadmin** client uses **jacl** as the default scripting language. You need not specify the '**-lang**' parameter.

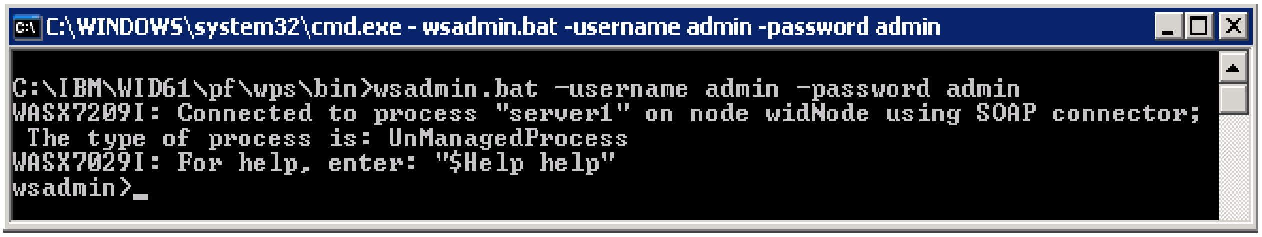
____ 1. Start **wsadmin** scripting client using the **jacl** language

__ a. Open a Command window and change to the bin directory of the WebSphere Process Server profile directory by entering the following command from the command prompt

```
cd  <WPS_PROFILE_HOME>\bin
```

__ b. Start the **wsadmin** client by entering the appropriate command from the command prompt. Ensure you specify username and password parameters if the administrative security is enabled

```
wsadmin.bat –lang jacl –username admin –password admin
```

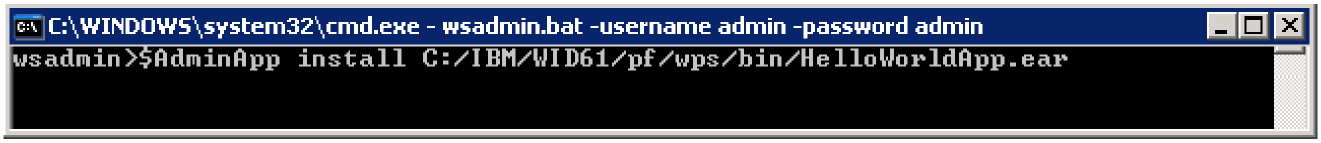


C:\WINDOWS\system32\cmd.exe - wsadmin.bat -username admin -password admin

```
C:\IBM\WID61\pf\wps\bin>wsadmin.bat -username admin -password admin
WASX7209I: Connected to process "server1" on node widNode using SOAP connector;
 The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>_
```

____ 2. Use the built-in '**AdminApp**' object to install the application by entering the following command in the wsadmin prompt. (Remember to use forward slashes when specifying the location of the ear file).

```
$AdminApp install <WPS_PROFILE_HOME>/bin/HelloWorldApp.ear
```



C:\WINDOWS\system32\cmd.exe - wsadmin.bat -username admin -password admin

```
wsadmin>$AdminApp install C:/IBM/WID61/pf/wps/bin/HelloWorldApp.ear
```

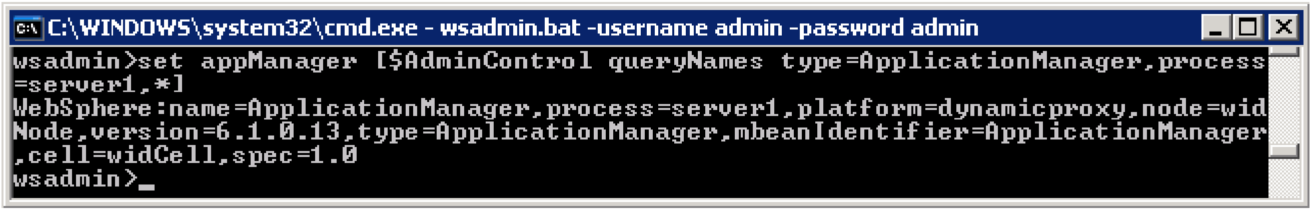
____ 3. Save the configuration by entering the following command

```
$AdminConfig save
```

____ 4. Start the installed application using wsadmin commands

__ a. Get the application manager object by entering the following command

```
set appManager [$AdminControl queryNames
type=ApplicationManager,process=<ServerName>,*]
```



C:\WINDOWS\system32\cmd.exe - wsadmin.bat -username admin -password admin

```
wsadmin>set appManager [$AdminControl queryNames type=ApplicationManager,process
=server1,*]
WebSphere:name=ApplicationManager,process=server1,platform=dynamicproxy,node=wid
Node,version=6.1.0.13,type=ApplicationManager,mbeanIdentifier=ApplicationManager
,cell=widCell,spec=1.0
wsadmin>_
```

Note: This assumes there is only one node.

__ b. Start the **HelloWorldApp** using the application manager object by entering the following command

```
$AdminControl invoke $appManager startApplication HelloWorldApp
```

____ 5. Verify the **HelloWorldApp** is accessible by entering the following URL in your Web browser: http://localhost:9080/HelloWorldWeb/index.jsp.

____ 6. Stop the application using wsadmin command by entering the following command under the wsadmin prompt

```
$AdminControl invoke $appManager stopApplication HelloWorldApp
```

____ 7. Uninstall the application by entering the following command under the wsadmin prompt

```
$AdminApp uninstall HelloWorldApp
```

____ 8. Save the configuration by entering the following command under the wsadmin prompt

```
$AdminConfig save
```

____ 9. Exit wsadmin by entering the following command at the wsadmin prompt

```
exit
```

____ 10. Use wsadmin script to deploy and start the application by issuing the appropriate command: (enter the command on a single line)

```
cd <WPS_PROFILE_HOME>\bin

wsadmin.bat –lang jacl –username admin –password admin –f
<LAB_FILES>\Admin\installSimpleApp.jacl HelloWorldApp.ear
```

____ 11. Verify the HelloWorldApp is accessible by entering the following URL in your Web browser

____ 12. Use wsadmin script to stop and uninstall the application by issuing the appropriate command:

```
cd <WPS_PROFILE_HOME>\bin

wsadmin.bat –lang jacl –username admin –password admin –f
<LAB_FILES>\Admin\uninstallSimpleApp.jacl
```

Part 4: Deploy an application using wsadmin client (-lang jython)

In this part of the lab, you will use the **wsadmin** scripting client to install the Service Component Architecture (SCA) application using the **jython** scripting language.

Note: The **wsadmin** client uses **jacl** as the default scripting language. You need to specify the '**-lang jython**' parameter to pass the jython scripting commands.

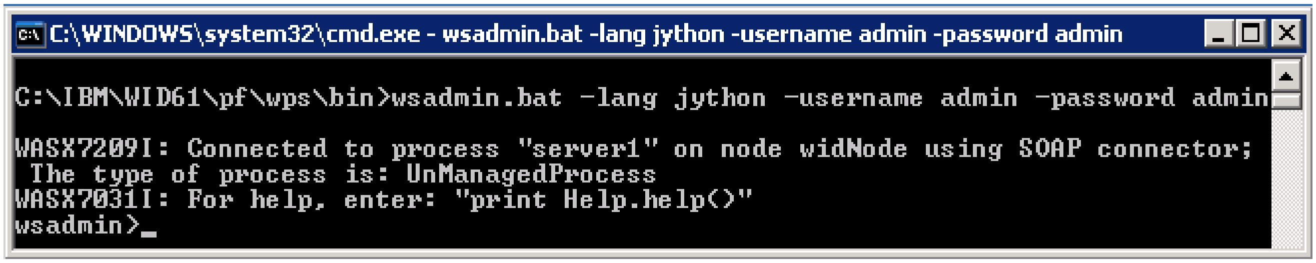
____ 1. Start **wsadmin** scripting client using **jython** scripting language

__ a. Open a Command window and change to the bin directory of the WebSphere Process Server profile directory by entering the following command from the command prompt

```
cd  <WPS_PROFILE_HOME>\bin
```

__ b. Start the **wsadmin** client by entering the appropriate command from the command prompt. Ensure you specify username and password parameters if the administrative security is enabled

```
wsadmin.bat –lang jython –username admin –password admin
```


C:\WINDOWS\system32\cmd.exe - wsadmin.bat -lang jython -username admin -password admin

```
C:\IBM\WID61\pf\wps\bin>wsadmin.bat -lang jython -username admin -password admin

WASX7209I: Connected to process "server1" on node widNode using SOAP connector;
 The type of process is: UnManagedProcess
WASX7031I: For help, enter: "print Help.help()"
wsadmin>_
```

____ 2. Use the built-in '**AdminApp**' object to install the application by entering the following command in the wsadmin prompt. (Remember to use forward slashes when specifying the location of the ear file).

```
print AdminApp.install('<WPS_PROFILE_HOME>/bin/HelloWorldApp.ear')
```

____ 3. Save the configuration by entering the following command

```
AdminConfig.save()
```

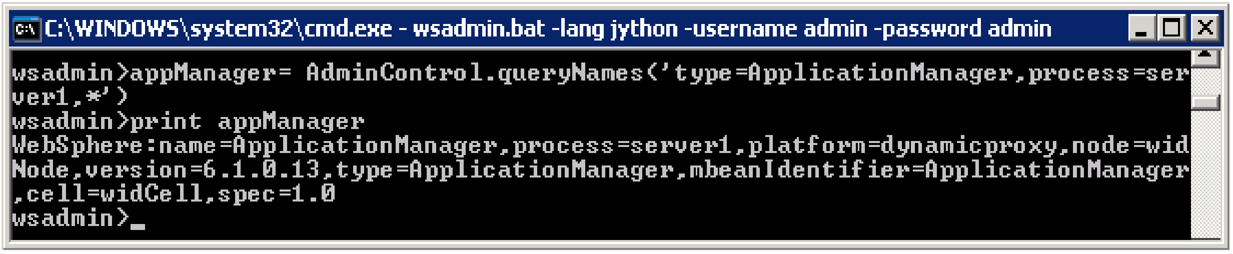
____ 4. Start the installed application using wsadmin commands

__ a. Get the application manager object by entering the following command

```
appManager=
AdminControl.queryNames('type=ApplicationManager,process=server1,*')

print appManager
```

Note: This assumes there is only one node.


C:\WINDOWS\system32\cmd.exe - wsadmin.bat -lang jython -username admin -password admin

```
wsadmin>appManager= AdminControl.queryNames('type=ApplicationManager,process=ser
ver1,*')
wsadmin>print appManager
WebSphere:name=ApplicationManager,process=server1,platform=dynamicproxy,node=wid
Node,version=6.1.0.13,type=ApplicationManager,mbeanIdentifier=ApplicationManager
,cell=widCell,spec=1.0
wsadmin>_
```

__ b. Start the **HelloWorldApp** using the application manager object by entering the following command

```
AdminControl.invoke(appManager, 'startApplication', 'HelloWorldApp')
```

____ 5. Verify the **HelloWorldApp** is accessible by entering the following URL in your Web browser: http://localhost:9080/HelloWorldWeb/index.jsp.

____ 6. Stop the application using wsadmin command by entering the following command under the wsadmin prompt

```
AdminControl.invoke(appManager, 'stopApplication', 'HelloWorldApp')
```

____ 7. Uninstall the application by entering the following command under the wsadmin prompt

```
AdminApp.uninstall('HelloWorldApp')
```

____ 8. Save the configuration by entering the following command under the wsadmin prompt

```
AdminConfig.save()
```

____ 9. Exit wsadmin by entering the following command at the wsadmin prompt

```
exit
```

____ 10. Use wsadmin script to deploy and start the application by issuing the appropriate command: (enter the command on a single line)

```
cd <WPS_PROFILE_HOME>\bin

wsadmin.bat –f <LAB_FILES>\Admin\installSimpleApp.py HelloWorldApp.ear
```

____ 11. Verify the HelloWorldApp is accessible by entering the following URL in your Web browser: http://<HOSTNAME>:9080/HelloWorldWeb/index.jsp.

____ 12. Use wsadmin script to stop and uninstall the application by issuing the appropriate command:

```
cd <WPS_PROFILE_HOME>\bin

wsadmin.bat –f c:\<LAB_FILES>\Admin\uninstallSimpleApp.py HelloWorldApp
```

What you did in this exercise

In this exercise you used the following tools to deploy, install, and uninstall a simple application:

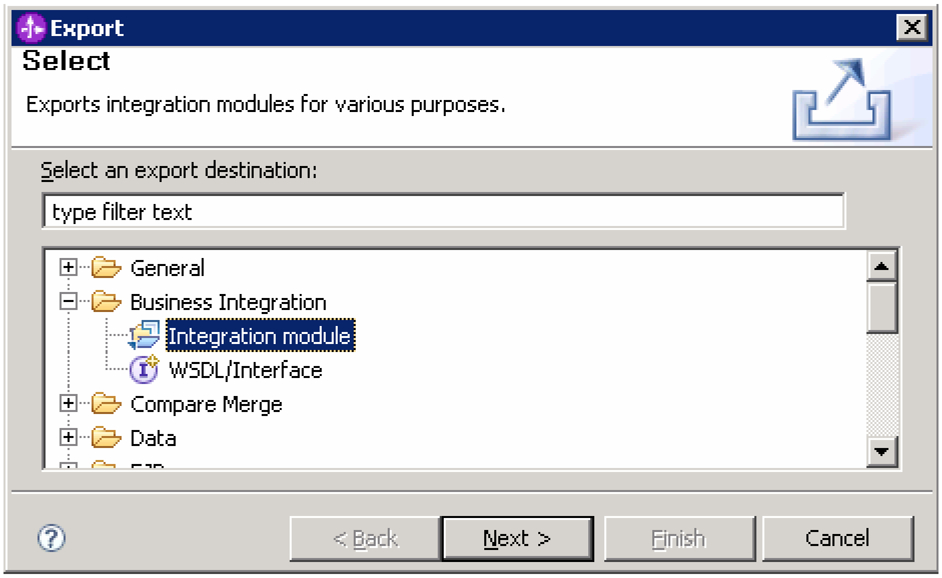
- Command line deployment using serviceDeploy
- Web browser based administrative console
- Command line based wsadmin

Task 1: Exporting modules as serviceDeploy files

To use the '**serviceDeploy**' command of WebSphere Process Server or WebSphere Enterprise Service Bus (mediation modules only) to build and deploy service modules as EAR files, you should export an archive that is good for '**Command line service deployment**'.

____ 1. In the WebSphere Integration Developer, from the '**File**' menu, select '**Export**'. The export wizard opens

____ 2. From the '**Export**' wizard, expand '**Business Integration**' and then select '**Integration Module**'



Export

Select

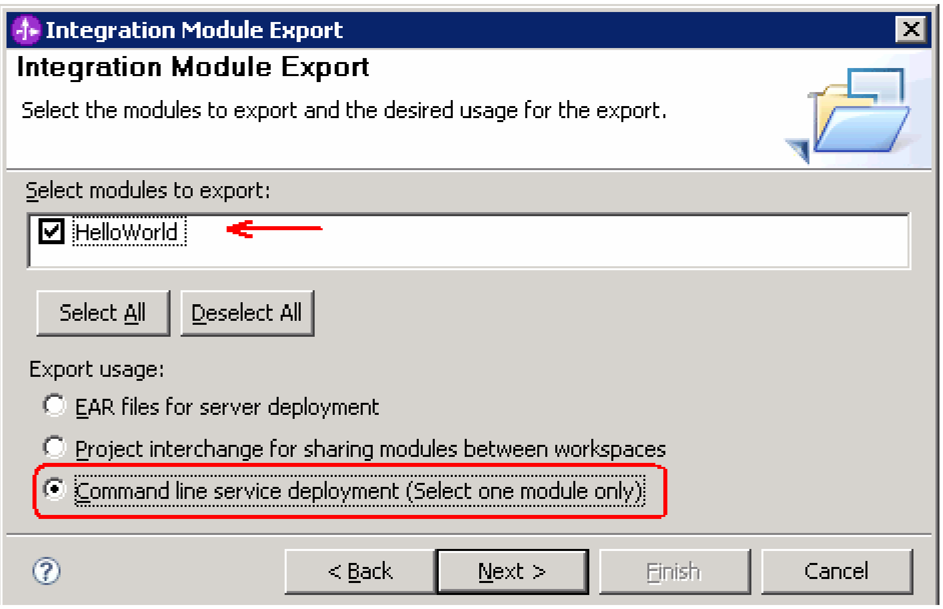
Exports integration modules for various purposes.

Select an export destination:

type filter text

____ 3. Click '**Next**'

____ 4. In the following '**Integration Module Export**' panel, select the check box the module you want to export and then select the radio button for '**Command line service deployment (Select one module only)**'



Integration Module Export

Integration Module Export

Select the modules to export and the desired usage for the export.

Select modules to export:

☑ HelloWorld

Export usage:

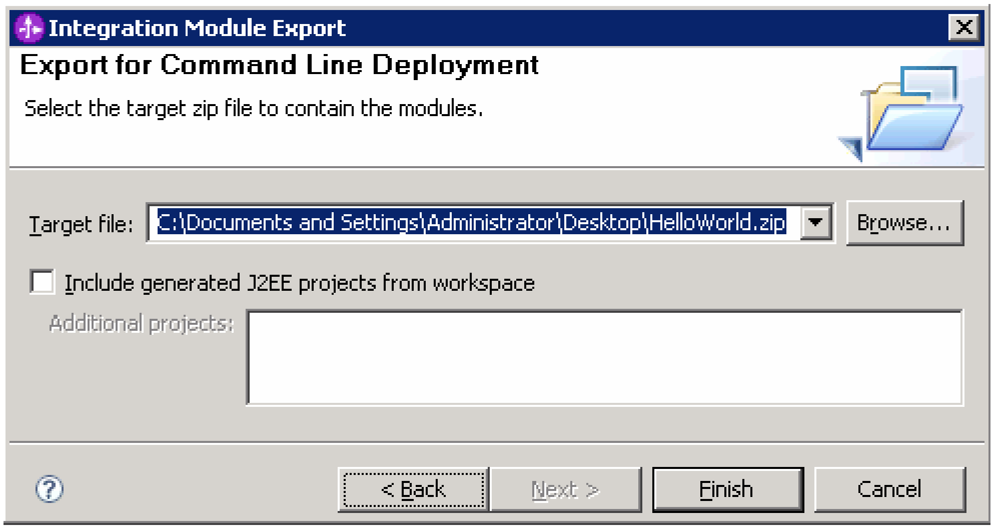
○ EAR files for server deployment

○ Project interchange for sharing modules between workspaces

● Command line service deployment (Select one module only)

____ 5. Click **Next**

____ 6. In the following '**Export for Command Line Deployment**' panel, enter the target file name (**.zip**)



Integration Module Export

Export for Command Line Deployment

Select the target zip file to contain the modules.

Target file: C:\Documents and Settings\Administrator\Desktop\HelloWorld.zip

☐ Include generated J2EE projects from workspace

Additional projects:

Note: If you have changed any of your generated J2EE projects, select the **Include generated J2EE projects from workspace** check box. The names of the J2EE projects will be displayed in the **Additional projects** list box. (By default, all dependencies are automatically included in the serviceDeploy zip archive file.

____ 7. Click **Finish** to export the selected module in a zip archive file.