AVP-2929

Jython DeepDive for WebSphere Application Server

IBM® WebSphere® Application Server V7 – LAB EXERCISE

Scripting tools

What this exercise is about

The objective of this lab is to provide you with an understanding of the improvements made to the scripting tools supplied with WebSphere Application Server V7. These changes include new scripting libraries, improved command assistance and better integration with script development tools.

This lab is provided AS-IS, with no formal IBM support.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Application Server V7 with an application server profile created

- Rational Application Developer V7.5

- You can also run this exercise with Rational Application Developer V5 and the WebSphere Application Server V7 test environment. In this case a stand-alone WebSphere Application Server installation is not required.

http://www-01.ibm.com/software/support/acceleratedvalue/

What you should be able to do

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

- Understand the basics of writing administrative scripts and using the administrative objects to perform common tasks.

- Use command assistance to view the commands the console uses to perform configuration changes and learn how to import these commands into the Jython script being currently edited in Rational Application Developer.

- Understand the basics of using Rational Application Developer to write, test and debug administrative scripts.

Introduction

This lab introduces you to the concept of making configuration changes using Jython scripts.

Using scripts is generally viewed as a better way to make configuration changes than using the administrative console. This is especially true when applying changes to large and complex installations. Scripts can be checked and tested before applying them to the production systems. They can also be source and change managed, providing better control; and they are reusable.

There are several ways to run Jython scripts in WebSphere Application Server:

- Using the wsadmin tool:

    o Command line mode

    o Batch mode

    o Interactive mode

- Using Rational Application Developer's facilities for writing, testing and debugging Jython scripts

The scripting language can be Jython or JACL. JACL has been deprecated and is not covered in this exercise.

There are several scripting objects that can be used to write scripts. These objects are fairly low level and a deeper understanding of the task to be accomplished is needed to use them. Starting in this version of the product, a new scripting library has been introduced. This library provides a much higher level of interface than the scripting objects.

This exercise is not designed to make you an expert at scripting, but to expose you to the basic concepts of scripting objects, the scripting library and the tools provided to aid in scripting.

http://www-01.ibm.com/software/support/acceleratedvalue/

Exercise instructions

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

Reference Variable	Command Line Path	Cygwin Path
<WAS_HOME>	C:\IBM\SDP\runtimes\base_v7\	/IBM/SDP/runtimes/base_v7/ or /IBM/WASV7/ (symlink)
<PROFILE_HOME>	C:\IBM\WASProfiles\AppServ1\	/IBM/WASProfiles/AppServ1/
<RAD_HOME>	C:\IBM\SDP\	/IBM/SDP/
<TEMP>	C:\temp\	/temp/

Note for Windows users: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\Labfiles70\ with C:/Labfiles70/

Part 1: Setting up the environment has been completed in advance by the AVP team.

Please start the lab at Part 2 on page 8

http://www-01.ibm.com/software/support/acceleratedvalue/

IBM Software Accelerated Value Program

Part 1: Setting up the environment

Without the environment specified in the Lab requirements section you cannot perform this lab. Make sure that you meet these requirements. Below are some acceptable environments for this exercise.
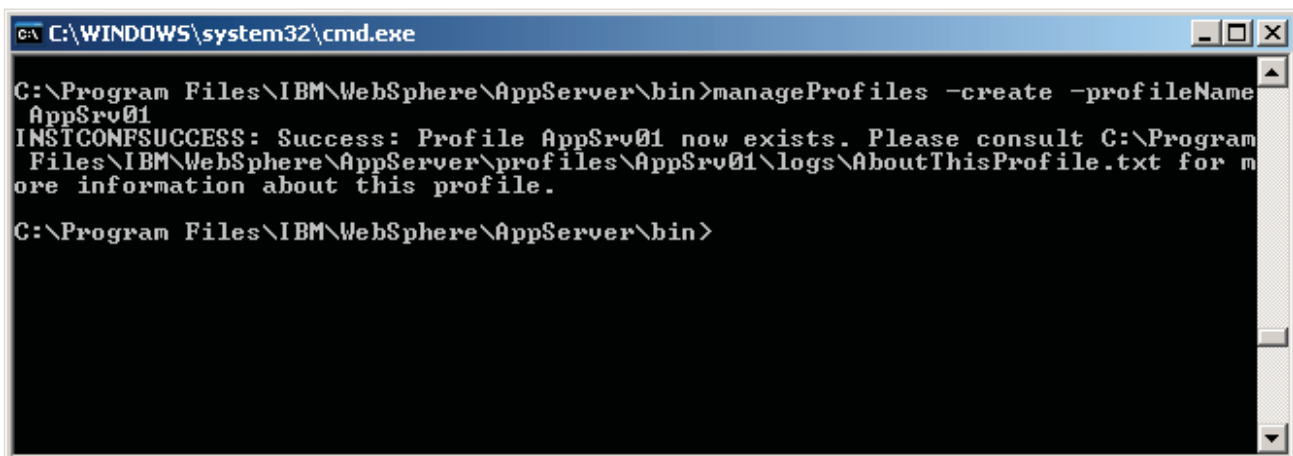
If you have Rational Application Developer installed with the WebSphere Application Server V7 test environment installed on your machine you may continue to the next part of this exercise, as you need no further setup instructions.

If you have Rational Application Developer installed without the WebSphere Application Server V7 test environment installed on your machine, and you are using a stand-alone WebSphere Application Server V7 installation and you already have a standalone profile you want to use, you may skip to step 2.

However, if you have WebSphere Application Server V7 test environment installed on your machine but you have not yet create a stand-alone server profile you want to use, perform steps 1 and 2 below.

____ 1.   Create a stand-alone application server profile using the manageProfiles utility.

   __ a. Open a command window and change directory to the <WAS_HOME>\bin folder.

   __ b. Run the following command to create a stand-alone profile:
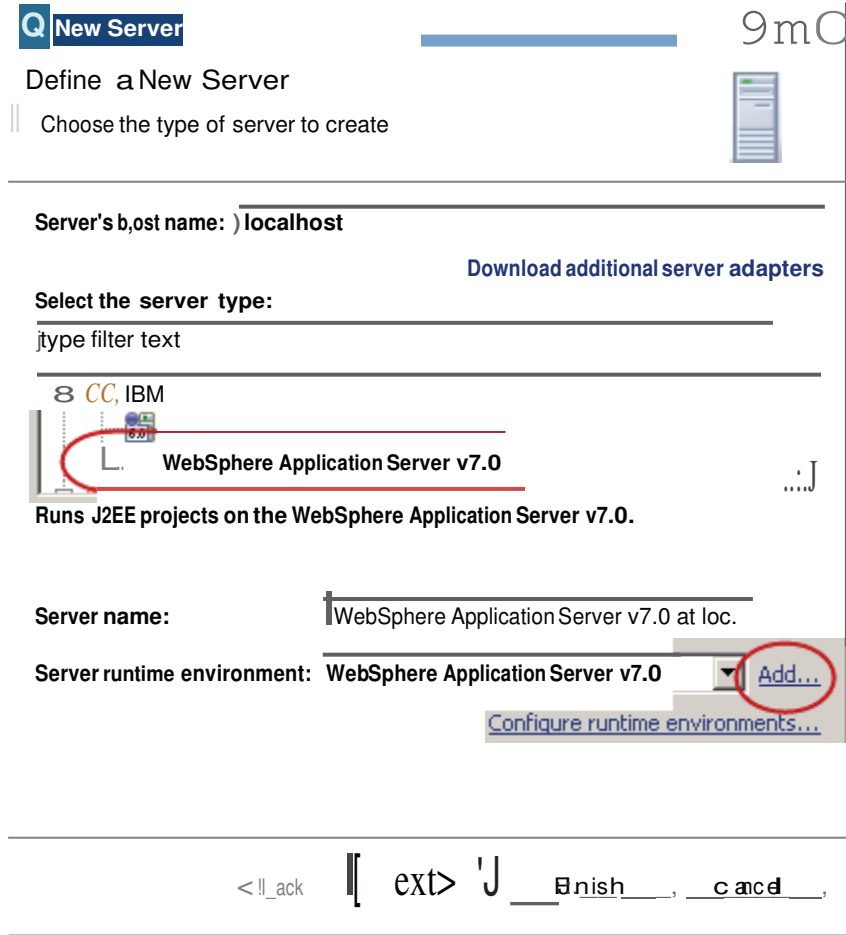
```
manageProfiles –create –profileName AppSrv01
```

   __ c. Wait till profile creation completes. Be patient it may take a few minutes.

____ 2.   Check to see if a server is already been defined on Rational Application Developer.

   __ a. Start Rational Application Developer, open a new workspace. Once the workspace is ready, close the Welcome view.

   __ b. Click the Server tab.

   __ c. If a server already exists in the Server view, it is quite possible that you have installed the WebSphere Test Environment. If there is already a server displaying and you want to use this server for the exercise you may now skip the remaining steps in this part of the lab and continue to the next part of the lab.

____ 3.   If there are no servers displaying at the moment in the Servers view, follow the next few steps to create a definition for the server profile you want to use.

http://www-01.ibm.com/software/support/acceleratedvalue/

Impact2011
Changing the Way Business and IT Leaders Work

__a. On the Server view, right click and select **New** Server from the pop-up menu.

__b. On the New Server wizard select Web-Sphere Application Server v7 as the server type, and then click Add to the right of the Server Runtime environment dropdown.

Q **New Server**

Define a New Server

Choose the type of server to create

Server's b,ost name:) **localhost**

Download additional server adapters

Select the server type:

jtype filter text

8 *CC,* IBM

L. **WebSphere Application Server v7.0**

Runs J2EE projects on the WebSphere Application Server v7.0.

Server name: WebSphere Application Server v7.0 at loc.

Server runtime environment: **WebSphere Application Server v7.0** Add...

Configure runtime environments...

< ‖_ack   ext> 'J ___Finish___ , ___c ance l___ ,

__ c. On the next dialog you point to the location of the runtime for the server you want to use. Click Browse and navigate to <WAS_HOME>. Click OK.
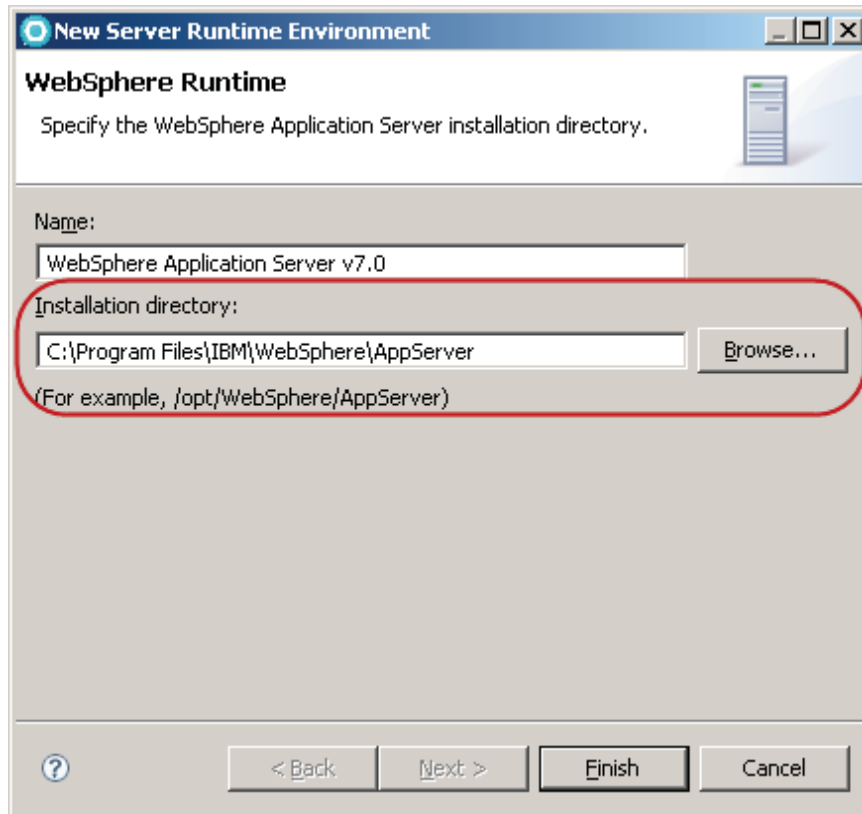
__ d. Click Finish.

__ e. Back on the New Server dialog, click Finish again.

__ f. There should now be a server defined in the Servers view.
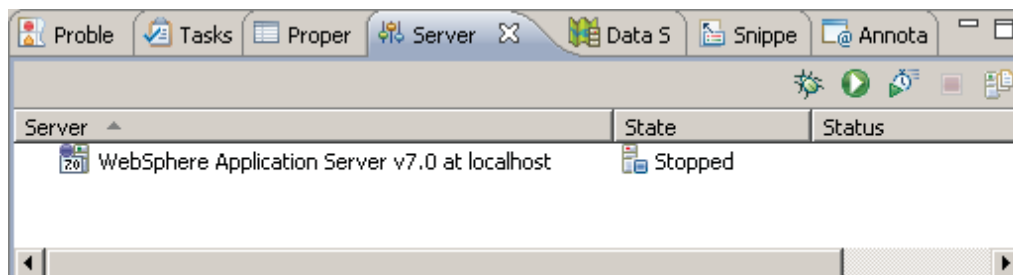
Part 2: Basic tasks using wsadmin

In this part of the exercise you start wsadmin, using Jython as the scripting language, and explore the administrative objects.

____1.   Make sure the application server is up and running To start the server"

    __ a. Change directory to `<PROFILE_HOME>\bin` and start the application server by issuing the following command:

```
startServer.bat server1
```

    __ b. Wait until the server starts.

Start wsadmin.

    __ c. Open a command window.

    __ d. Navigate to :

```
<PROFILE_HOME>\bin
```

    __ e. Start the wsadmin tool by entering the following command:

```
wsadmin –lang jython –username wsadmin –password wsadmin
```

NOTE: if administrative security has not been enabled on your server, you can omit the username and password parameters. If you have multiple servers on a cell you can define which server you are connecting to by specifying the –host and –port parameters. In a federated environment you would connect to the deployment manager instead of to the individual servers.

    __ f. Wait until wsadmin loads the default Jython libraries. This takes a while, but only happens the first time wsadmin is started.

    __ g. Once wsadmin presents its prompt, you can start using it in interactive mode. This is similar to other interactive development environments where you enter a command, it is executed, and the results are shown immediately.

____2.   As you may know, there are five wsadmin administrative objects. They are:

- Help

- AdminControl

- AdminConfig

http://www-01.ibm.com/software/support/acceleratedvalue/

- AdminApp

- AdminTask

Almost all server configurations can be performed through these objects.

Use wsadmin to display the help provided by each of the administrative objects.

__ a. At the wsadmin prompt enter the following command:

```
Help.help()
```

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython                    _ □ X
wsadmin>
wsadmin>
wsadmin>
wsadmin>Help.help()
'WASX7028I: The Help object has two purposes: \n\n\tFirst, provide general help
information for the the objects\n\tsupplied by wsadmin for scripting: Help, Admi
nApp, AdminConfig,\n\tand AdminControl.\n\n\tSecond, provide a means to obtain i
nterface information about\n\tMBeans running in the system.  For this purpose, a
 variety of\n\tcommands are available to get information about the operations,\n
\tattributes, and other interface information about particular\n\tMBeans.\n\n\tT
he following commands are supported by Help; more detailed\n\tinformation about
each of these commands is available by using the\n\t"help" command of Help and s
upplying the name of the command\n\tas an argument.\n\n\nattributes\t\tgiven an
MBean, returns help for attributes\noperations\t\tgiven an MBean, returns help f
or operations\nconstructors\t\tgiven an MBean, returns help for constructors\nde
scription\t\tgiven an MBean, returns help for description\nnotifications\t\tgive
n an MBean, returns help for notifications\nclassname\t\tgiven an MBean, returns
 help for classname\nall\t\t\tgiven an MBean, returns help for all the above\nhe
lp\t\t\treturns this help text\nAdminControl\t\treturns general help text for th
e AdminControl object\nAdminConfig\t\treturns general help text for the AdminCon
fig object\nAdminApp\t\treturns general help text for the AdminApp object\nAdmin
Task\t\treturns general help text for the AdminTask object\nwsadmin\t\t\treturns
 general help text for the wsadmin script\n\t\t\tlauncher\nmessage\t\t\tgiven a
message id, returns explanation and\n\t\t\tuser action message'
wsadmin>
```

___ b. As you can see the output has not been formatted, and it is hard to read. Now try the following command:

```
print Help.help()
```

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython                          _ □ ×
wsadmin>print Help.help()
WASX7028I: The Help object has two purposes:

        First, provide general help information for the the objects
        supplied by wsadmin for scripting: Help, AdminApp, AdminConfig,
        and AdminControl.

        Second, provide a means to obtain interface information about
        MBeans running in the system.  For this purpose, a variety of
        commands are available to get information about the operations,
        attributes, and other interface information about particular
        MBeans.

        The following commands are supported by Help; more detailed
        information about each of these commands is available by using the
        "help" command of Help and supplying the name of the command
        as an argument.

attributes              given an MBean, returns help for attributes
operations              given an MBean, returns help for operations
constructors            given an MBean, returns help for constructors
description             given an MBean, returns help for description
notifications           given an MBean, returns help for notifications
classname               given an MBean, returns help for classname
all                     given an MBean, returns help for all the above
help                    returns this help text
AdminControl            returns general help text for the AdminControl object
AdminConfig             returns general help text for the AdminConfig object
AdminApp                returns general help text for the AdminApp object
AdminTask               returns general help text for the AdminTask object
wsadmin                 returns general help text for the wsadmin script
                        launcher
message                 given a message id, returns explanation and
                        user action message
wsadmin>
```

Using the command print in front of nearly every other command produces formatted output, which is much easier to read.

___ c. Now try getting help for the other administrative objects:

- AdminConfig.help()

- AdminControl.help()

- AdminApp.help()

- AdminTask.help()

___ d. You can also request specific help on a particular command, or method, of an administrative object. Try the following:

- AdminTask.help("-commandGroups") to display the command groups available to this object.

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython
wsadmin>print AdminTask.help("-commandGroups")
WASX8005I: Available admin command groups:

AdminAgentNode - Admin Agent Managed Node related tasks
AdminAgentSecurityCommands - Commands used to configure security related items d
uring Admin Agent registration.
AdminReports - Admin configuration reports
AdministrativeJobs - This command group contains all the job management commands
.
AppManagementCommands - Application management commands.
AuditAuthorizationCommands - Audit Authorization Table Commands
AuditEmitterCommands - Commands for managing the Audit service providers.
AuditEncryptionCommands - Commands for managing Security Auditing encryption.
AuditEventFactoryCommands - Commands for managing Security Auditing Event Factor
CellConfigCommands - Commands for Cell Configuration
CentralizedInstallCmds - A group of administrative commands for centralized inst
allation support.
CertificateRequestCommands - Command that manage certificate request.
ChannelFrameworkManagement - A group of admin commands that help in configuring
the WebSphere Transport Channel Service
ClusterConfigCommands - Commands for configuring application server clusters and
 cluster members.
ConfigArchiveOperations - A command group that contains various config archive r
elated operations.
```

- Suppose you are interested in the group of commands for configuring a cluster. Enter the command: AdminTask.help("ClusterConfigCommands") to display the commands available to configure a cluster.

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython
wsadmin>print AdminTask.help("ClusterConfigCommands")
WASX8007I: Detailed help for command group: ClusterConfigCommands

Description: Commands for configuring application server clusters and cluster me
mbers.

Commands:
createCluster - Creates a new application server cluster.
createClusterMember - Creates a new member of an application server cluster.
deleteCluster - Delete the configuration of an application server cluster.
deleteClusterMember - Deletes a member from an application server cluster.
listClusterMemberTemplates - No description available
updateCluster - Updates the configuration of an application server cluster.
```

http://www-01.ibm.com/software/support/acceleratedvalue/

- AdminTask.help("createCluster") provides additional help on how to create a new cluster.

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython

wsadmin>print AdminTask.help("createCluster")
WASX8006I: Detailed help for command: createCluster

Description: Creates a new application server cluster.

Target object:   None

Arguments:
  None

Steps:
  clusterConfig - Specifies the configuration of the new server cluster.
  replicationDomain - Specifies the configuration of a replication domain for t
his cluster.  Used for HTTP session data replication.
  convertServer - Specifies an existing server will be converted to be the firs
t member of cluster.
  eventServiceConfig - Specifies the event service configuration of the new ser
ver cluster.
  promoteProxyServer - If a proxy server was specified for convertServer, apply
 the proxy settings for the contentServer to the cluster.

wsadmin>
```

This information, along with more detailed information from the online documentation, can help you figure out how to create a cluster using scripting. Many scripting operations are simple and straightforward. However, other operations might require a bit of investigation, reading, and trial and error, making it difficult for some users to take full advantage of scripting in their environments. In an effort to help users overcome the complexity of scripting, a new set of script libraries has been introduced in WebSphere Application Server V7. In the next part of the lab, you explore these libraries.

http://www-01.ibm.com/software/support/acceleratedvalue/

Part 3: Exploring the scripting libraries

As mentioned in the previous section, scripting libraries are now provided to take the task of scripting to a higher level of abstraction. Many scripting operations can be performed using a single line from a library function. In this part of the lab you take an initial look at the available libraries and explore what they have to offer.

____1. Open Windows Explorer and navigate to the following folder:

```
<WAS_HOME>\scriptLibraries
```
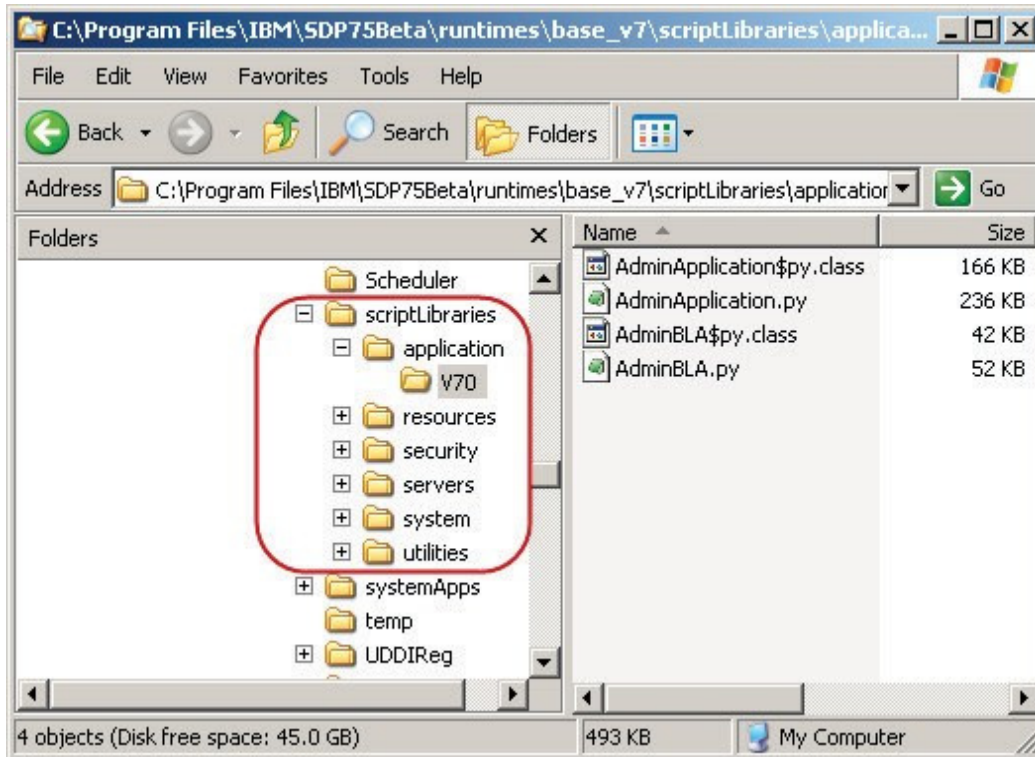
_____ 2.   Open each of the subfolders and discover the name of the libraries:

Library name	
AdminApplication	AdminBLA
AdminJ2C	AdminJDBC
AdminJMS	AdminResources
AdminAuthorizations	AdminClusterManagement
AdminServerManagement	AdminNodeGroupManagement
AdminNodeManagement	AdminLibHelp
AdminUtilities	ApplyPerfTuningTemplates

NOTE: the libraries use the administrative objects you looked at in the previous part of this exercise. All these libraries are loaded when wsadmin starts and are readily available from the wsadmin command prompt, or to be used from your own scripts. Even though source code is provided, it is not meant to be modified by the user. Users of the libraries, call code in the libraries from their own scripts. You may copy parts of the library code to other files and modify the copied code to improve it or better suit your needs.

3. If you are familiar with Jython you can open the libraries in a text editor and look at the code. You will note that the code is well documented and exceptions and other errors are handled gracefully by providing meaningful error messages to the calling scripts.

As an example open the `AdminJDBC.py` file found in the `<WAS_HOME>\scriptingLibrary\resources\JDBC\V70` folder. A few snippets are show below:

```
#################################################################
# Licensed Material - Property of IBM
# 5724-I63, 5724-H88, (C) Copyright IBM Corp. 2005 - All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#
# DISCLAIMER:
# The following source code is sample code created by IBM Corporation.
# This sample code is provided to you solely for the purpose of assisting
# you in the use of the product. The code is provided 'AS IS', without #
warranty or condition of any kind. IBM shall not be liable for any
# damages arising out of your use of the sample code, even if IBM has
# been advised of the possibility of such damages.
#################################################################
#
#---------------------------------------------------------------------
# AdminJDBC.py - Jython procedures for performing JDBCProvider and
# DataSource tasks.
#---------------------------------------------------------------------
#
#   This script includes the following procedures:
#     Ex1: createJDBCProvider
#     Ex2: createJDBCProviderUsingTemplate
#     Ex3: listJDBCProviderTemplates
#     Ex4: createDataSource
#     Ex5: createDataSourceUsingTemplate
#     Ex6: listDataSourceTemplates
#     Ex7: listJDBCProviders
#     Ex8: listDataSources
#     Ex9: help
#
#---------------------------------------------------------------------
```

After the copyright and disclaimer statement, a list of the functions provided by the library is documented. As you can see, these functions expose operations at a much higher level than those provided by the administrative objects, and provide a better abstraction to the script writer. Feel free to examine some of the functions provided by the library.

4. A similar listing of functions available in a library can be obtained using the help() method. At the wsadmin prompt enter:

```
print AdminJDBC.help()
```

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython                          _ □ ×
wsadmin>print AdminJDBC.help()
WASL0007I: The AdminJDBC script library provides the script
        procedures that configure and query Java Database Connectivity (JDBC) pr
ovider and data source settings.

        The AdminJDBC script library provides the following script procedures.
        To display detailed information about each script procedures, use the he
lp command
        for the AdminJDBC script library, specifying the name of the script of i
nterest
        as an argument.


createDataSource:
        Create a new data source in your configuration.
        The script returns the configuration ID of the new data source.

createDataSourceUsingTemplate:
        Use a template to create a new data source in your configuration.
        The script returns the configuration ID of the new data source.

createJDBCProvider:
        Create a new JDBC provider in your environment.
        The script returns the configuration ID of the new JDBC provider.
```

5. As with the administrative objects, you can ask for help on a specific method:

```
print AdminJDBC.help("listJDBCProviders")
```

```
C:\WINDOWS\system32\cmd.exe - wsadmin -lang jython                          _ □ ×
wsadmin>print AdminJDBC.help("listJDBCProviders")
WASL5108I: Procedure: listJDBCProviders

        Optional arguments: jdbcName

        Description: List all JDBCProvider in a cell or list the specific one if
 jdbcName is specified.

        Usage: AdminJDBC.listJDBCProvider()

        Usage: AdminJDBC.listJDBCProvider(jdbcName)
wsadmin>
```

_____ 6.   And you can run the command by entering:

```
print AdminJDBC.listJDBCProviders()
```


_____ 7.   Explore the other files in the library that you might be interested in. Feel free to try the help() or any other library method.

As with any new library system it takes a while to become familiar and comfortable with the functions available. By combining these library functions with your own scripting logic in your Jython scripts you should soon be writing scripts to configure your application servers.

Part 4: Console command assistance

Starting with WebSphere Application Server V6.1, a set of scripting tools was added to the product. These tools appeared as support for Jython scripting in the development tools, like Rational Application Developer and Application Server Toolkit. The development tools now provide a Jython source editor with syntax coloring and command completion, and a Jython source level debugger with breakpoint and variable inspection support.

On the server side, command assistance was introduced by providing the console user with the ability to display the Jython command the console would use to effect a configuration change. The commands displayed can be imported into the current Jython script being edited in the development tool.

Although no new tools have been introduced in WebSphere Application Server V7, the existing tools have been improved. The command assistance feature has been considerably improved to cover most administrative commands started from the console.

In this part of the exercise you explore these tools.

_____ 1.   Enable command assistance preferences in the console.

___ a. Make sure your server is up and running.

Start the server: <PROFILE_HOME>/bin/startServer.bat server1

___ b. Log in to the administrative console.

Use Firefox to login to the WAS Admin Console
- http://localhost:9060/ibm/console

___ c. On the left navigation bar, expand System administration, click Console Preferences.

d. Check Enable command assistance notifications and Log command assistance commands.
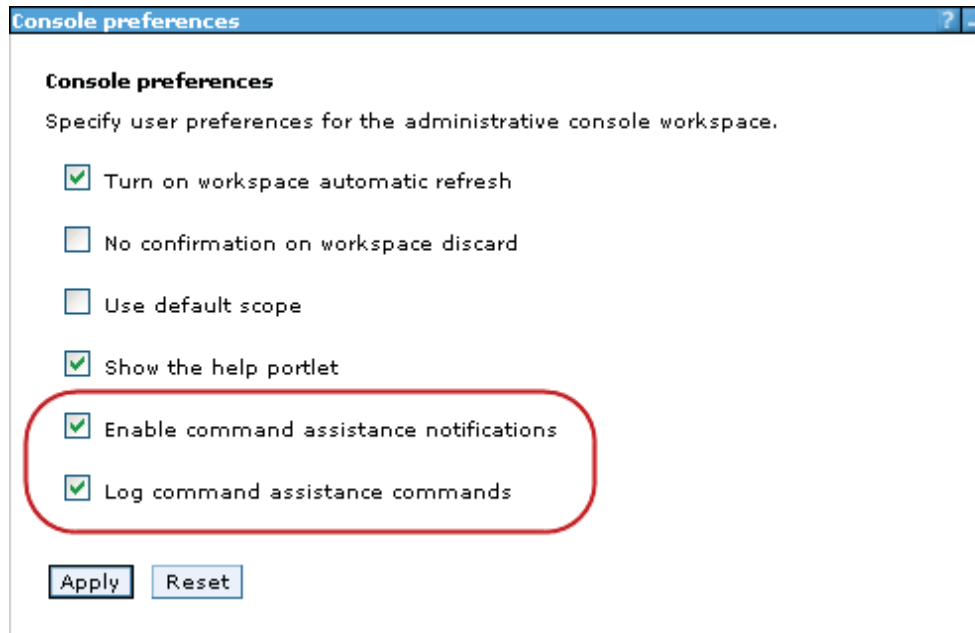
NOTE: the first option only works with the complementing option in your IBM development tool. You enable that feature in Rational Application Developer later in this exercise. The second option allows you to save the command assistance output to a log file, where you can review or edit it later. This file is placed in the server logs folder with a name of: `CommandAssistanceJythonCommands_.log`.

e. Click Apply.

2. See how command assistance works by following the next few steps. From the left navigation bar expand Resources, expand JDBC and click JDBC providers.

3. On the right side, under Help, note the link View administrative scripting command for last action.

http://www-01.ibm.com/software/support/acceleratedvalue/

____ 4.   Click View administrative scripting command for last action.

The wsadmin command shown above is the equivalent command to perform the action that the console application used to get the list of JDBC providers currently defined on the server.

The command was also saved on the `CommandAssistanceJythonCommands_.log` file for future reference.

____ 5.   Try other operations on the console and see what commands are constructed. Note that not all operations will result in the link to display the generated wsadmin command.

http://www-01.ibm.com/software/support/acceleratedvalue/

Part 5: Writing Jython scripts

There are a few options for writing Jython scripts. You can use a plain old text editor, or you can use a development environment such as Rational Application Developer. Whichever method you choose has its advantages and disadvantages.

Writing the actual scripts in a development environment has some productivity enhancements, such as code completion, color syntax and an integrated source level debugger. These are very nice features that can help you get your code up and running faster.

Running a Jython script under wsadmin can cause some productivity problems. The wsadmin tool is a Java program, and as such, it runs in a Java Virtual Machine (JVM). Starting and stopping a JVM is costly in terms of time and resources. Normally the JVM starts, then wsadmin loads, run the script, and exits, shutting down the JVM in the process. Every time you change the script and want to test it you go through this costly operation.

There is a way to circumvent this problem when running wsadmin from the command line, unfortunately this is harder to implement when using the development environment.

To illustrate this point follow the steps below:

1. Using a text editor (Notepad, WordPad) create a new file in the `c:\scripts` folder. Create the folder if it does not exist. Name this file `test.py`.

2. Type the following line in the `test.py` file and save it, leaving the editor open:

    ```
    print "This is a Jython script"
    ```

3. To run the script enter the following command:

    ```
    wsadmin –lang jython –f test.py
    ```

    The wsadmin tool starts, loads `test.py`, executes it and exits. If you make a change to the file and need to run it again, you re-enter the same command and the cycle repeats. The concern is the time it takes for wsadmin to start.

    If you are at the test stage and making many small changes frequently, the time to start and stop wsadmin can be considerable.
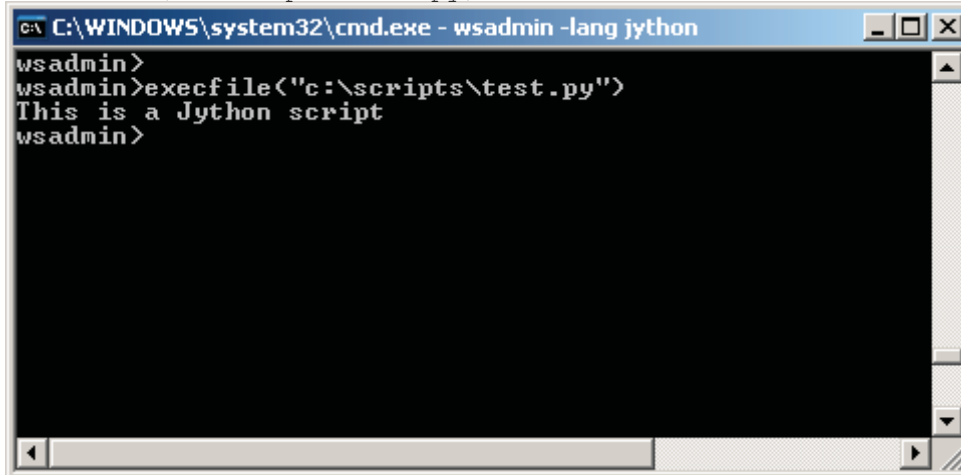
    While testing and making frequent changes to your scripts you might consider using either a plain text editor or Rational Application Developer's Jython editor to create and make changes to the script; and running the script in wsadmin in the following manner:

4. Start wsadmin using the following command:

    ```
    wsadmin –lang jython
    ```

http://www-01.ibm.com/software/support/acceleratedvalue/

5. Use the `execfile()` Jython function to load and run the script you are working on. In your case enter:
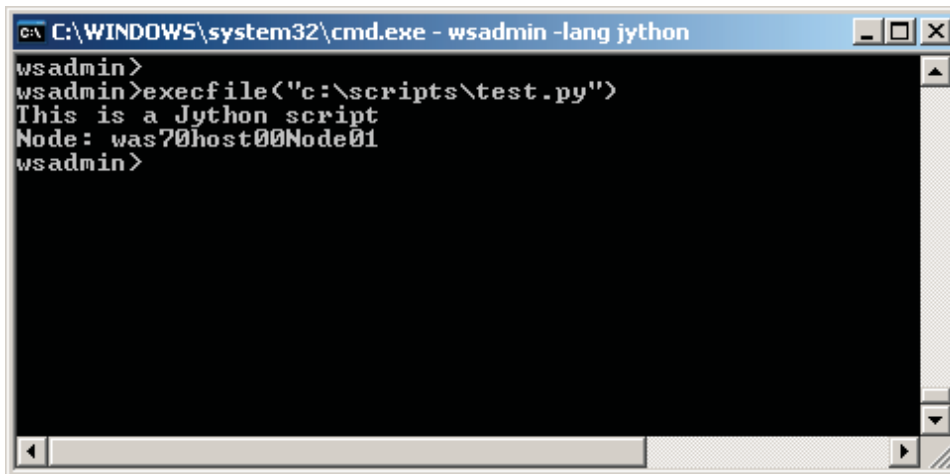
```
execfile(c:\scripts\test.py)
```


6. Without exiting wsadmin, add the following three lines to the script:

```
#Get Node name
nodeName = AdminControl.getNode()
print("Node: " + nodeName)
```

7. Switch back to the wsadmin prompt and recall the execfile() function (using the up arrow key), press the Enter key to re-run the script.

8. You can repeat this process until development of the script is complete.

Note: when using this method wsadmin only needs to be started once. This is the preferred method to test your scripts while in development mode. If you need to debug a script, beyond using print statements, you will be well served by Rational Application Developer's source level debugger. When using the development workbench, you will need to wait for wsadmin to restart, but it is well worth it! You will use these tools in the next section of the lab.


Part 6: Tools for writing and testing scripts

Writing administrative scripts requires you to use the Jython language for program flow and logic, and the administrative objects and libraries to configure WebSphere Application Server. It is not the intention of this exercise to teach you the Jython programming language, its syntax and constructs; there are many online and printed resources to do so. Having said that, Jython can be very easy to learn and use, as its syntax is similar to other languages you may already know.

In this part of the lab you write a very simple Jython script to interrogate the application server. You use Rational Application Developer's Jython tools to do so.

_____1.   Make sure the application server is up and running.

_____2.   Start Rational Application Developer if it is not already running, open a new workspace. Once the workspace is ready, close the Welcome view.

_____3.   Create a new Jython project named Scripts.

___ a. From the main menu click File -> New -> Other.

___ b. In the New dialog select Jython -> Jython project. Click Next.
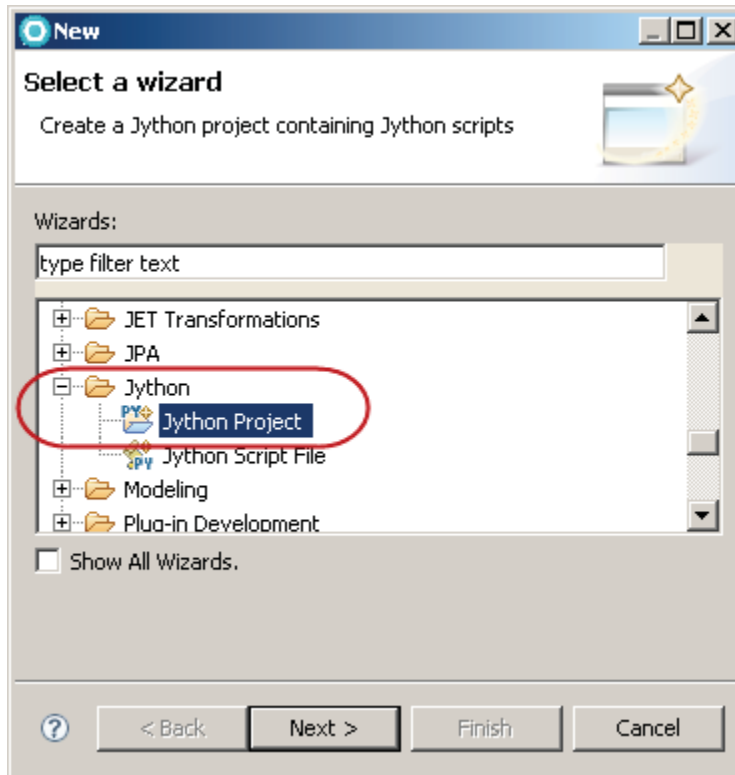
___ c. On the next page type Scripts for the name and click Finish.

http://www-01.ibm.com/software/support/acceleratedvalue/

___ d. The Scripts Jython projects now exist in the workspace.
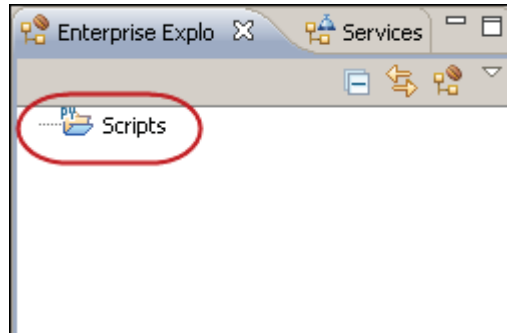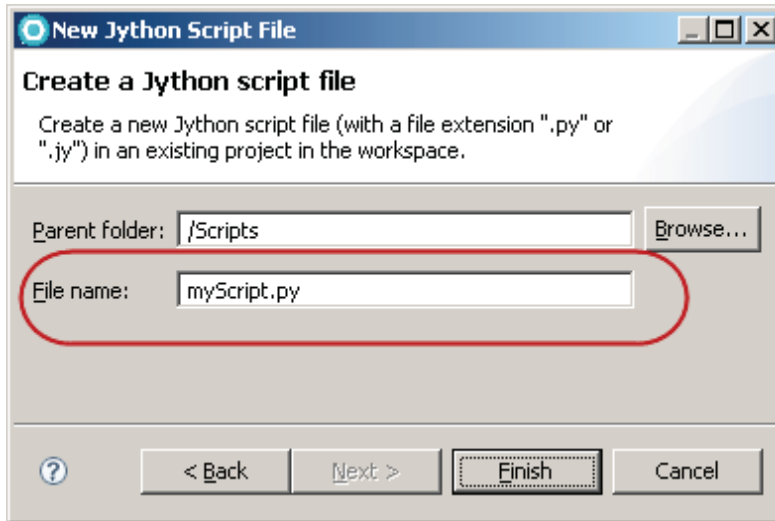

____ 4. Create a new Jython script file named myScript.

___ a. Select the newly created Scripts Jython project folder. Right click on Scripts to bring up a pop-up menu and select: New -> Other -> Jython -> Jython Script File.

___ b. Click Next.

http://www-01.ibm.com/software/support/acceleratedvalue/
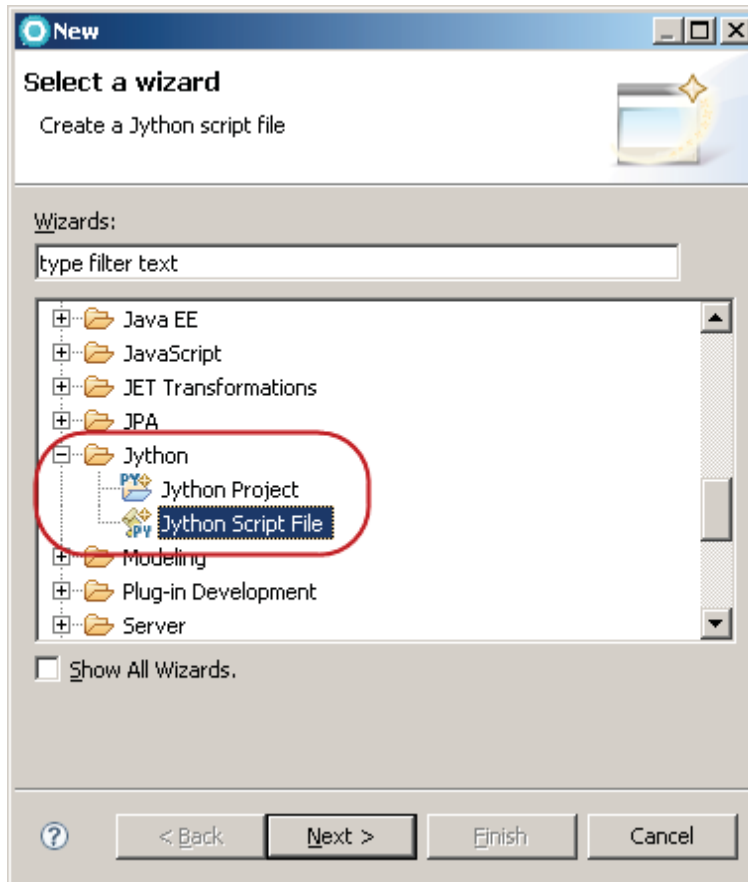
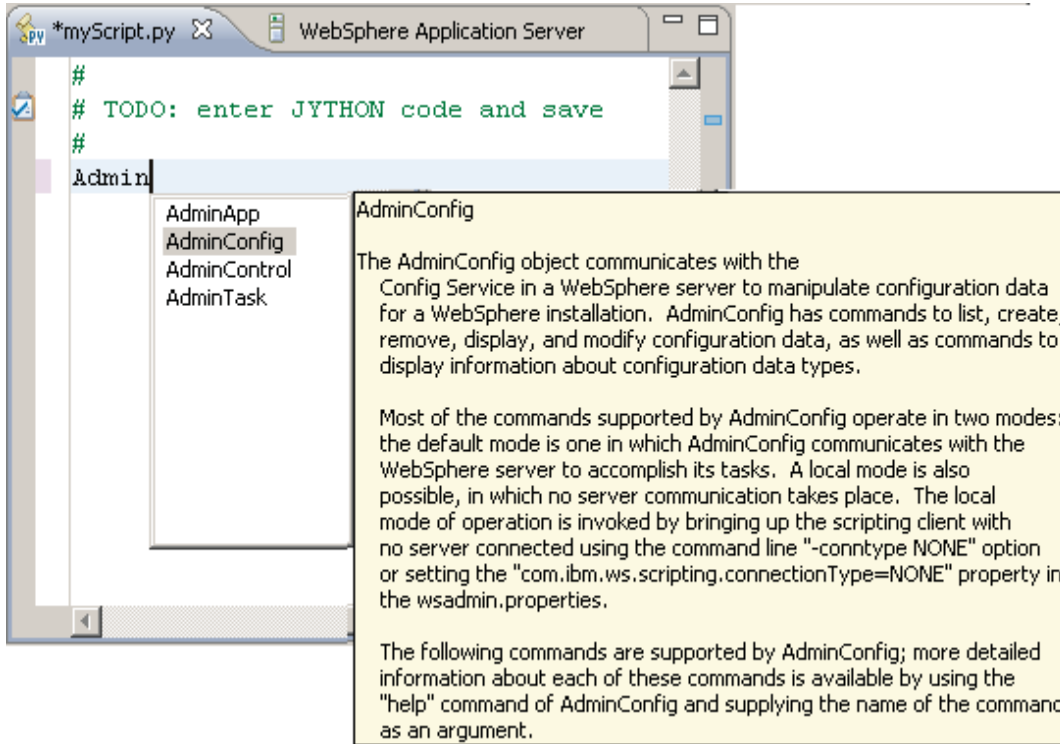____ c. Type myScript.py for the file name.

____ d. Click Finish.

____ e. The file is created and it opens in the Jython editor.

____5. One of the recommended actions in every script is to discard any pending changes that may not have been saved from a previous execution of the script. The command is:
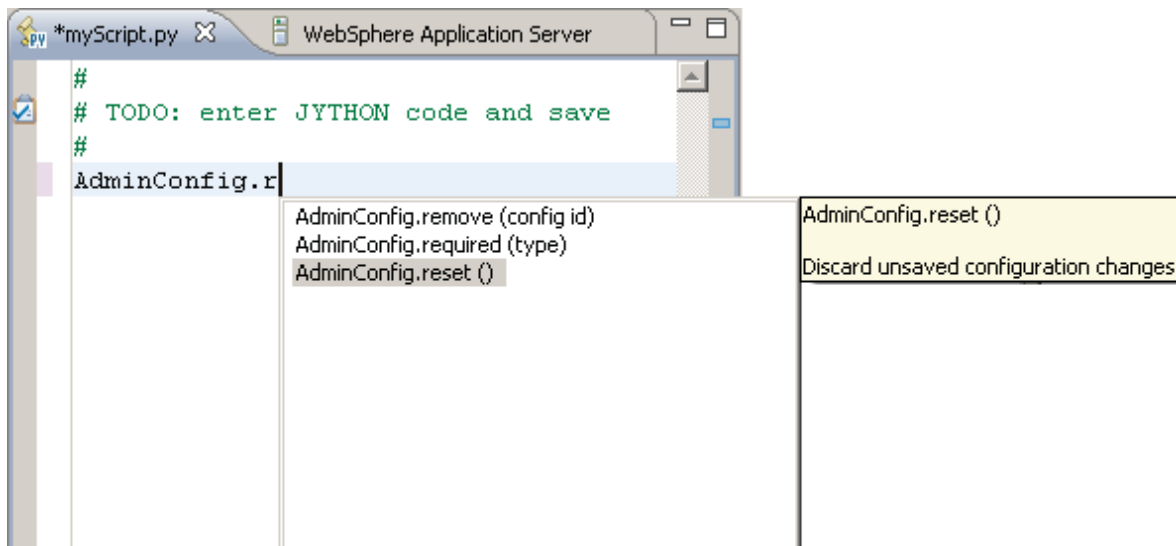
```
AdminConfig.reset()
```

Do not enter the command all at once. Doing it slowly demonstrates the command completion functionality of Rational Application Developer.

___ a. On the Jython editor view, enter `Admin` and press Ctrl-space. A list of the known administrative objects pops-up; as you move from object to object, using the up and down arrow keys, a description of each object is shown.

```
*myScript.py         WebSphere Application Server

#
# TODO: enter JYTHON code and save
#
Admin

        AdminApp          AdminConfig
        AdminConfig
        AdminControl      The AdminConfig object communicates with the
        AdminTask           Config Service in a WebSphere server to manipulate configuration data
                            for a WebSphere installation.  AdminConfig has commands to list, create,
                            remove, display, and modify configuration data, as well as commands to
                            display information about configuration data types.

                            Most of the commands supported by AdminConfig operate in two modes:
                            the default mode is one in which AdminConfig communicates with the
                            WebSphere server to accomplish its tasks.  A local mode is also
                            possible, in which no server communication takes place.  The local
                            mode of operation is invoked by bringing up the scripting client with
                            no server connected using the command line "-conntype NONE" option
                            or setting the "com.ibm.ws.scripting.connectionType=NONE" property in
                            the wsadmin.properties.

                            The following commands are supported by AdminConfig; more detailed
                            information about each of these commands is available by using the
                            "help" command of AdminConfig and supplying the name of the command
                            as an argument.
```

___ b. Select AdminConfig and press Enter.

___ c. To select the reset() method of the AdminConfig object enter a period and type the letter r. Now press Ctrl-space to invoke command completion.

```
*myScript.py         WebSphere Application Server

#
# TODO: enter JYTHON code and save
#
AdminConfig.r

        AdminConfig.remove (config id)      AdminConfig.reset ()
        AdminConfig.required (type)
        AdminConfig.reset ()                Discard unsaved configuration changes
```

http://www-01.ibm.com/software/support/acceleratedvalue/

_____ d. From the list select the reset() method.

_____ 6. Enter the next two lines to list the applications currently installed on the server:

```
print "Applications installed: "
print AdminApp.list()
```

_____ 7. Press Ctrl-s to save the file.

_____ 8. In the Enterprise Explorer view, right click myScript.py and select Run As -> Administrative Script.

http://www-01.ibm.com/software/support/acceleratedvalue/
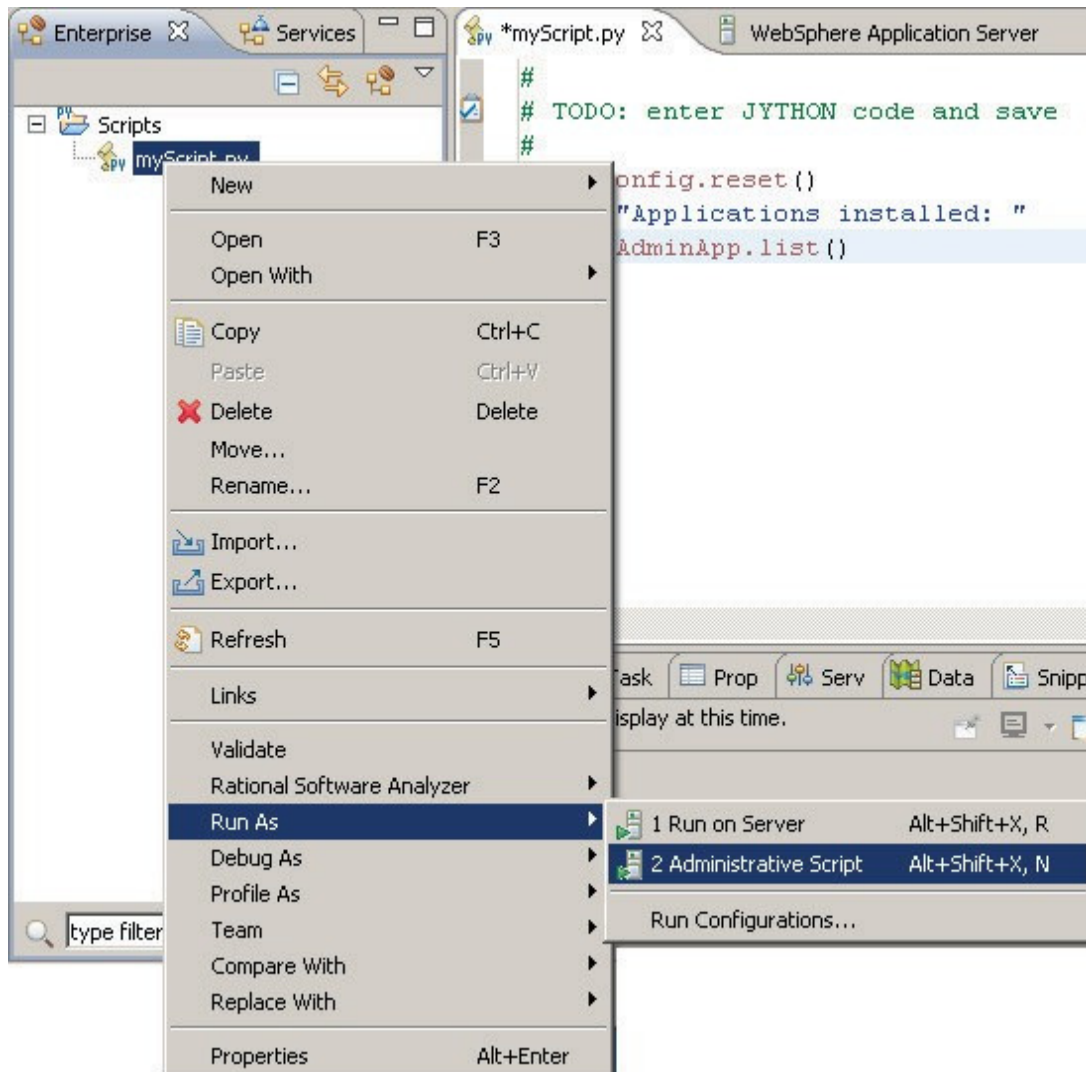
9. From the Edit Configuration panel, select WebSphere Application Server v7 in the Scripting runtime dropdown menu. In the WebSphere profile dropdown menu, choose the name of your profile. If security is enabled on the application server enter a valid User Id and Password.

Note: in your environment the WebSphere profile name may be different than the profile name shown below.
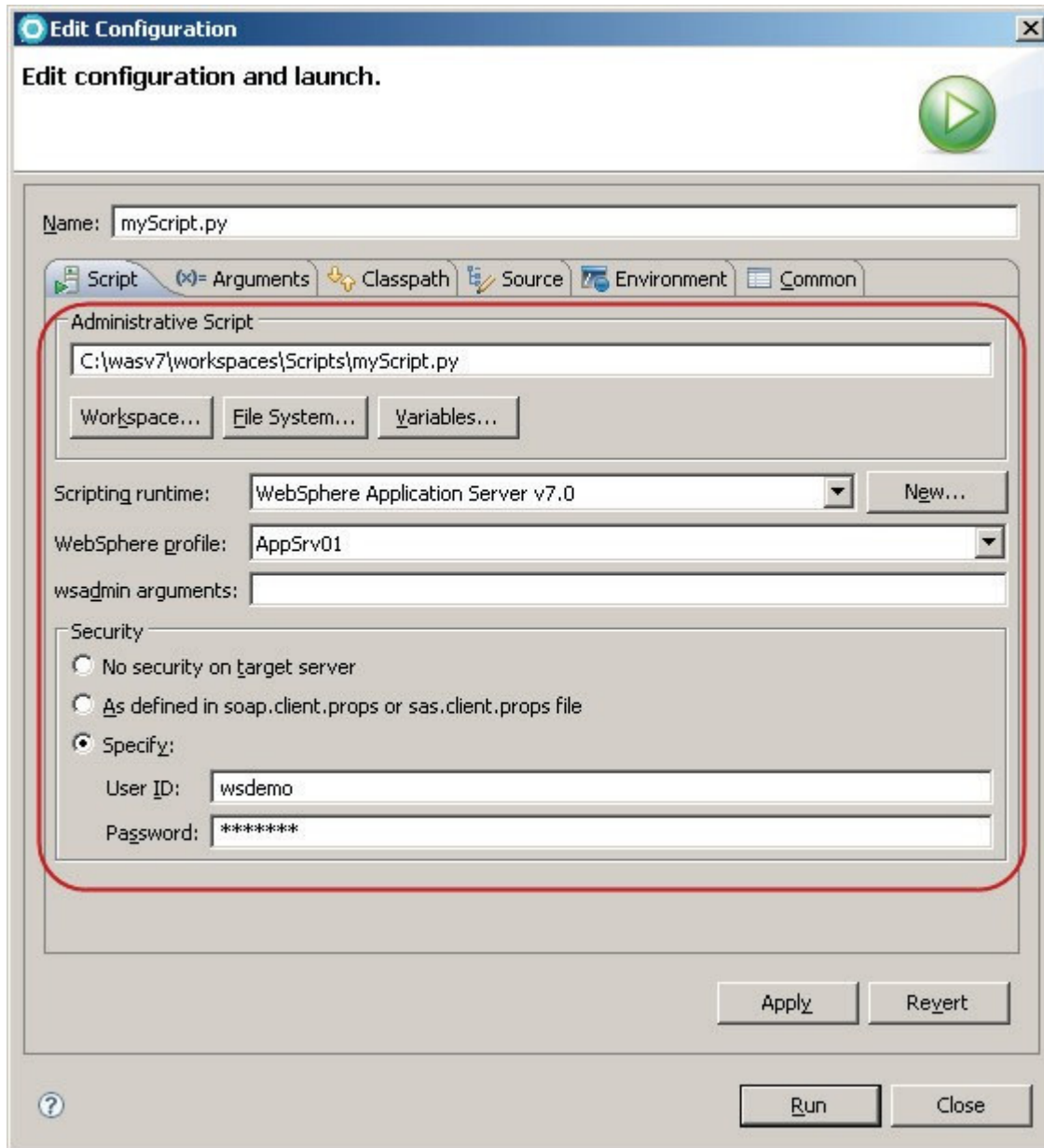
10. Click Run.

_____11.   Observe what happens next:

   __ a. Rational Application Developer starts wsadmin and passes the name of the script as the argument.

   __ b. The first time wsadmin starts it loads the Jython libraries. This can take some time, but it only happens once.

   __ c. The script is loaded and it runs.

   __ d. The output of the script appears in the Rational Application Developer's Console view.

```
Proble   Console    Tasks   Propert   Servers   Data S

<terminated> myScript.py [WebSphere Administrative Script] C:\Program Files\IBM\'\java\jr

webtools.det

*sys-package-mgr*: processing new jar, 'C:\Program
*sys-package-mgr*: processing new jar, 'C:\Program
*sys-package-mgr*: processing new jar, 'C:\Program
WASX7209I: Connected to process "server1" on node
Applications installed:
DefaultApplication

IBMUTC

ivtApp

query
```
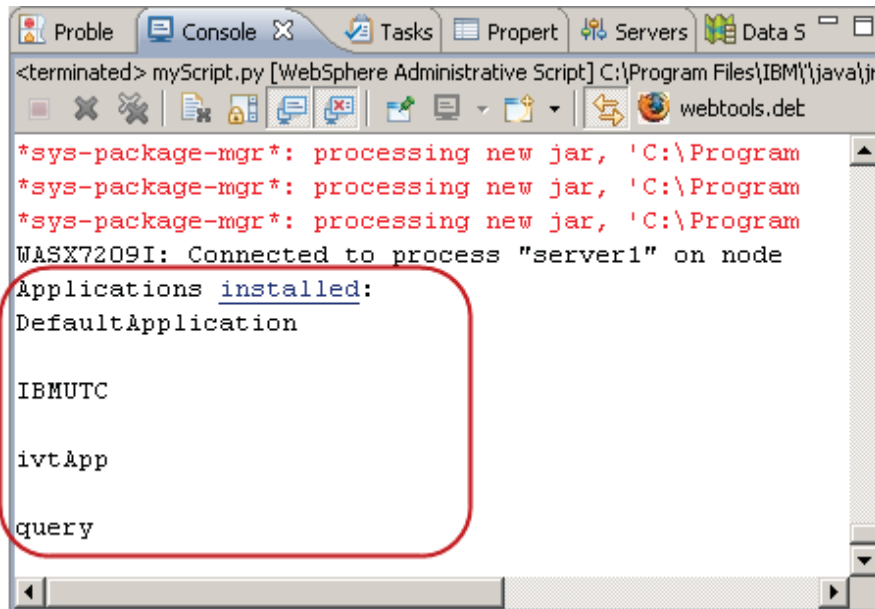
_____12.   Console command assistance allows you to capture actions being performed in the administrative console as Jython statements. In this portion of the exercise, you will use console command assistance to include some more complex commands in your Jython script.

   __ a. Remember that earlier in the exercise you enabled command assistance on the administrative console's preferences.

http://www-01.ibm.com/software/support/acceleratedvalue/

___ b. In Rational Application Developer bring the Servers view into the foreground. Select the
   application server, right click, and from its pop-up menu select Administration -> WebSphere
   console command assist
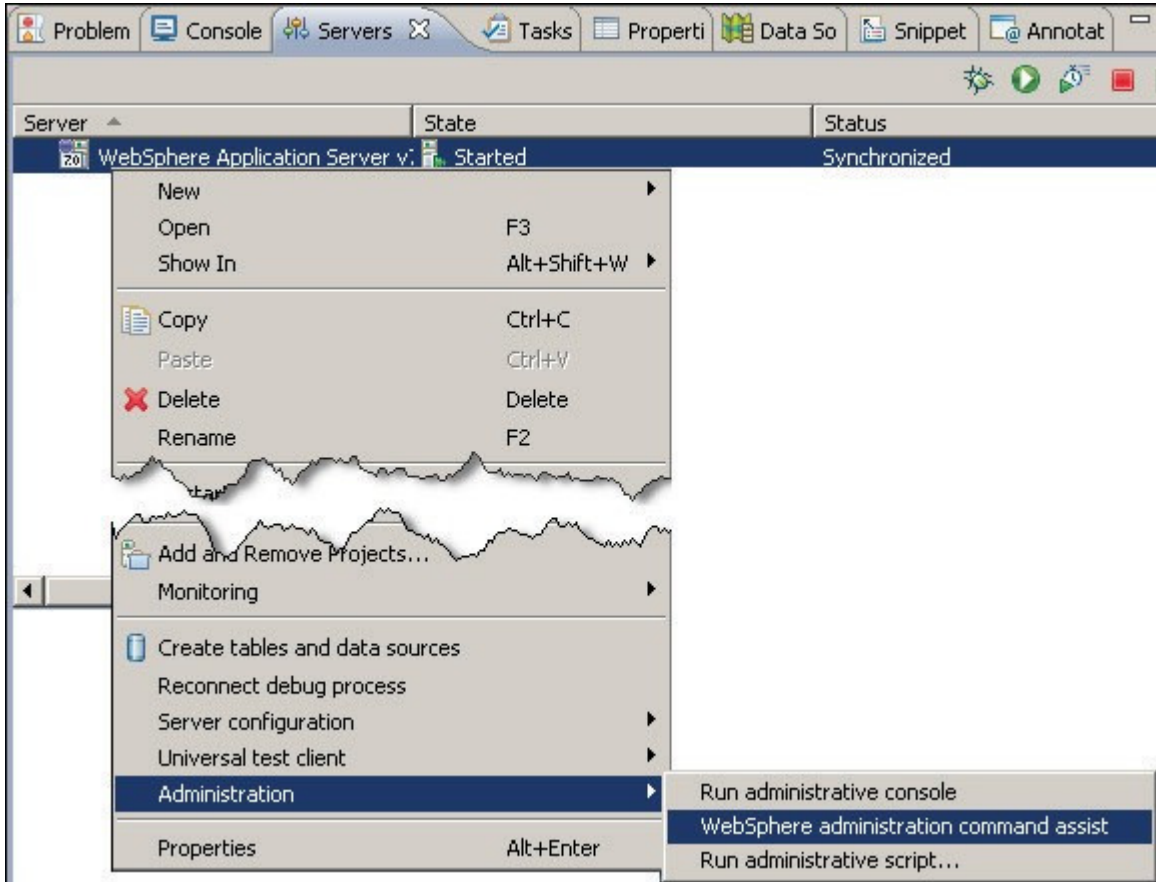
___ c. The WebSphere Administration Command view opens. On this view select the server to
   monitor by clicking the Select Server to Monitor icon and selecting your server from the list.
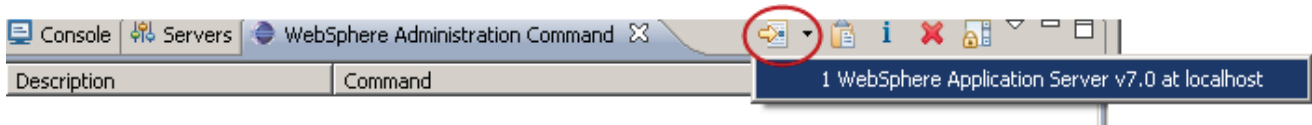
Make sure that after you select the server, a checkmark appears indicating the connection has
been made.
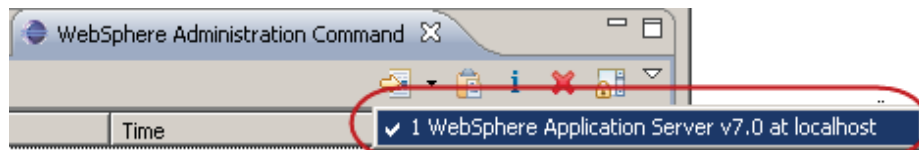
_____ 13. Open a browser to the administrative console; you may open it in a separate browser or within
       Rational Application Developer. Login to the console application using valid credentials.

http://www-01.ibm.com/software/support/acceleratedvalue/

____14. The next statement you insert into the script is fairly complex and you would have to invest a fair amount of time looking through the documentation to figure out all the parameters and the proper syntax for the statement to work as desired. The statement performs a very common task in WebSphere; setting up WebSphere variables, in this case the path to the DB2 JDBC driver.

__ a. From the left navigation bar expand Environment and click WebSphere variables.

__ b. From the Scope dropdown list, select All scopes.

__ c. Find the DB2_JDBC_DRIVER_PATH variable and click its link.

__ d. In the Value field enter the location where the DB2 drivers are located:

```
C:\Program Files\IBM\SQLLIB\java
```


Note: You may not have the driver installed on your machine. Since the driver is not really used anywhere this fact does not matter. This is just a sample on how to configure a WebSphere environment variable.

__ e. Click OK.

http://www-01.ibm.com/software/support/acceleratedvalue/

___ f. Do not save the changes. Instead look at the rightmost part of the page and click the View administrative scripting command for last action link to display the command constructed to make the change effective in the configuration.

The first line represents the command to modify the configuration to define the WebSphere variable which points to the DB2 JDBC driver. The second line is the previous command used by the console application to display the list of WebSphere variables.
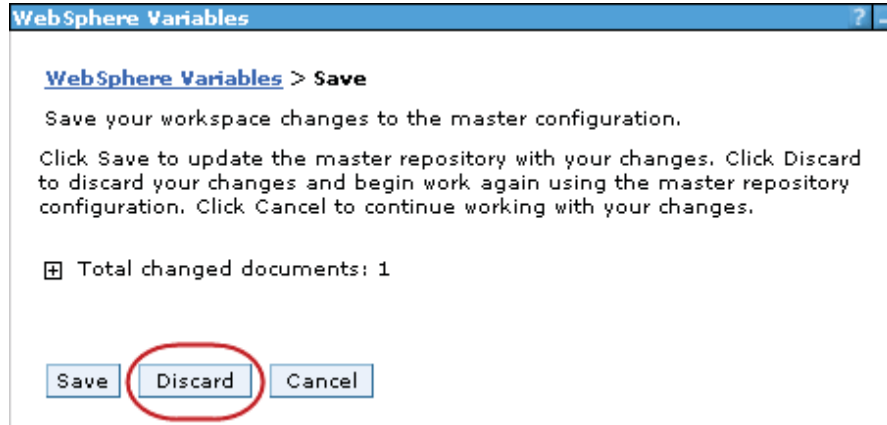
___ g. Close the Administrative Scripting Commands pane.

___ h. Back in the administrative console click the link to Review the changes you just made. Do not save.

http://www-01.ibm.com/software/support/acceleratedvalue/

i. In the Confirmation dialog, click Discard.


15. Switch to the WebSphere Administration Command view in Rational Application Developer and insert the command to alter the WebSphere variable into the script.

a. Place the cursor inside the Jython editor below the last existing line.

b. Select the AdminConfig.modify command, which appears in the WebSphere Administration Command view.

c. Click the Insert into Editor icon.
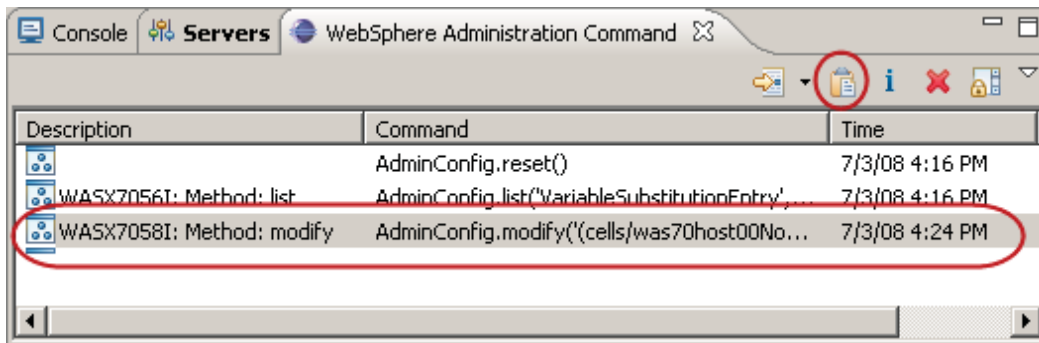
d. The command should now appear in your script at the line where the cursor was placed.

```
*myScript.py
#
# TODO: enter JYTHON code and save
#
AdminConfig.reset()
print "Applications installed: "
print AdminApp.list()

AdminConfig.modify(\
    '(cells/was70host00Node01Cell/nodes/was70host00Node01
    '[[symbolicName "DB2_JDBC_DRIVER_PATH"] [description
```

e. Make the changes permanent. After the last line insert the following commands:

```
print "Saving configuration"
AdminConfig.save()
```

f. Save the script.

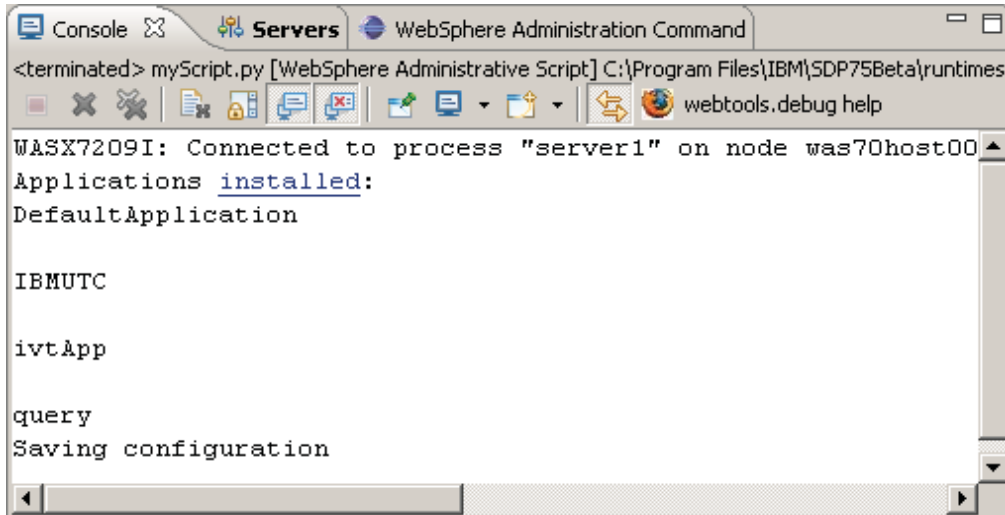16. Test the script and verify that the variable has been properly set.

a. Using the WebSphere administrative console, navigate to the WebSphere variables page, make sure the scope is set to All scopes.

b. Find the DB2_JDBC_DRIVER_PATH variable and make sure that no value has been set.

Select	Name	Value
	You can administer the following resources:	
☐	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps
☐	CONNECTJDBC_JDBC_DRIVER_PATH	
☐	CONNECTOR_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedConnectors
☐	DB2390_JDBC_DRIVER_PATH	
☐	DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	
☐	DB2UNIVERSAL_JDBC_DRIVER_PATH	
☐	DB2_JDBC_DRIVER_PATH	
☐	DEPLOY_TOOL_ROOT	${WAS_INSTALL_ROOT}/deploytool/itp

c. Logout of the console. It is always a good idea to be logged out when running administrative scripts.

34

___ d. Back in Rational Application Developer select the myscript.py script, right click, and from its pop-up menu click Run As -> Administrative Script.

___ e. Watch the console output. The script should run and exit without any errors.

```
Console  ⚙ Servers  ● WebSphere Administration Command

<terminated> myScript.py [WebSphere Administrative Script] C:\Program Files\IBM\SDP75Beta\runtimes\
  ■  ✕  ✖  ▤ ▦ ▤ ▣  ▦ ▤ ▾ ▭ ▾  ⤢ 🔴 webtools.debug help

WASX7209I: Connected to process "server1" on node was70host00 ▲
Applications installed:
DefaultApplication

IBMUTC

ivtApp

query
Saving configuration
◄                                                               ► ▼
```

___ f. Log back into the administrative console and verify that the WebSphere variable was set correctly.

| Select	Name ↕	Value ↕
	You can administer the following resources:	
☐	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps
☐	CONNECTJDBC_JDBC_DRIVER_PATH	
☐	CONNECTOR_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedConnectors
☐	DB2390_JDBC_DRIVER_PATH	
☐	DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	
☐	DB2UNIVERSAL_JDBC_DRIVER_PATH	
☐	DB2_JDBC_DRIVER_PATH	C:\Program Files\IBM\SQLLIB\java
☐	DEPLOY_TOOL_ROOT	${WAS_INSTALL_ROOT}/deploytool/itp

Part 7: Debugging scripts

This part of the exercise takes a simplistic view at the basics of debugging Jython scripts. The built-in source level debugger is used to step through and set a breakpoint on a new script.

_____ 1.   Create a new Jython script under the Scripts folder. Name this script debugTest.py.

Note: if you need help creating the file, refer to the previous part of this exercise.

_____ 2.   Turn on line numbering for the file.

___ a. Click anywhere in the text area of the editor.

___ b. Right click to bring up the pop-up menu for the editor.

___ c. Click Preferences.
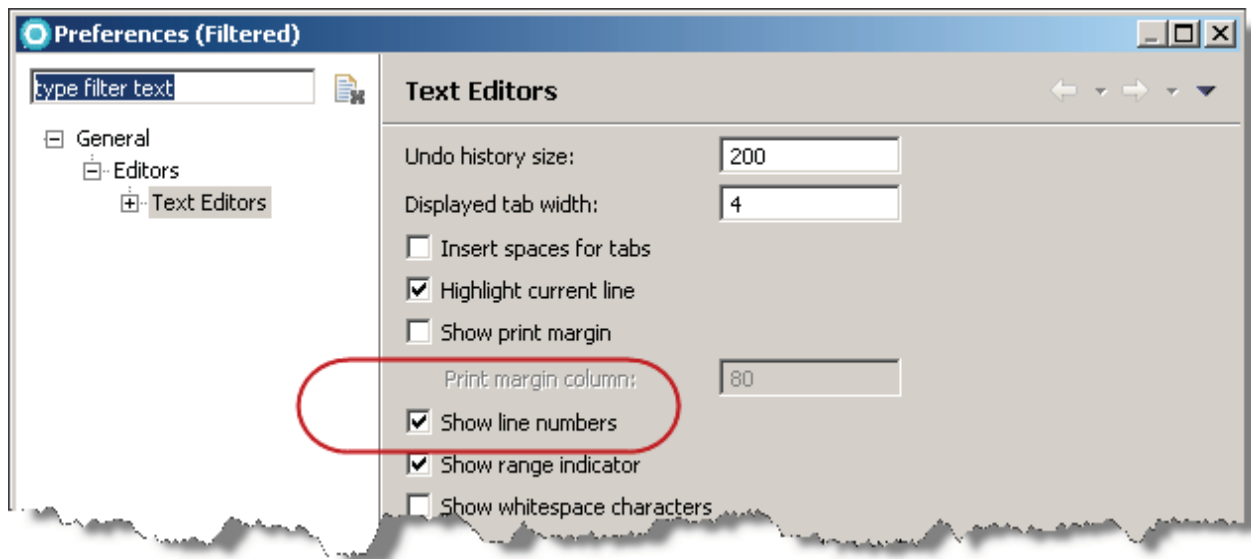
___ d. Select Show line numbers.

___ e. Click OK.

http://www-01.ibm.com/software/support/acceleratedvalue/

3. Enter the following lines:

```
1  #
2  # TODO: enter JYTHON code and save
3  #
4
5  #Get Node name
6  nodeName = AdminControl.getNode()
7  print("Node: " + nodeName)
8
9  i = 0
10 fruits = ["apple", "banana", "pear", "orange"]
11
12 for fruit in fruits:
13         i = i + 1
14         print i, ") ",  fruit
```

Let us examine the code:

- Lines 1 through 5 are comments.

- Lines 6 and 7 get the application server's node name and print it out to the console.

- Line 9 defines the variable i which is initialized to zero.

- Line 10 defines a Jython list named fruits and is initialized with the values of some fruits.

- Lines 12 through 14 define a Jython for loop that cycles through each of the elements of the fruits list. Each element of the list is assigned to the fruit variable while looping. The loop also increments the value of i. Finally it prints the elements of the list and performs some basic formatting.

Note: Jython's syntax is sensitive to indentation, notice how the body of the for loop is indented from the loop declaration. In Jython, levels of indentation delimit blocks of code.

4. Save the file.

5. Set a breakpoint to prevent execution of the script to complete without debugging. Normally you place breakpoints in areas of your code that are giving you problems. In this case, add a breakpoint on line 6 of the script.

   a. Place the mouse pointer on the marker bar just to the left of line 6.

   b. Double click to set a breakpoint. A sphere should now show besides line 6.

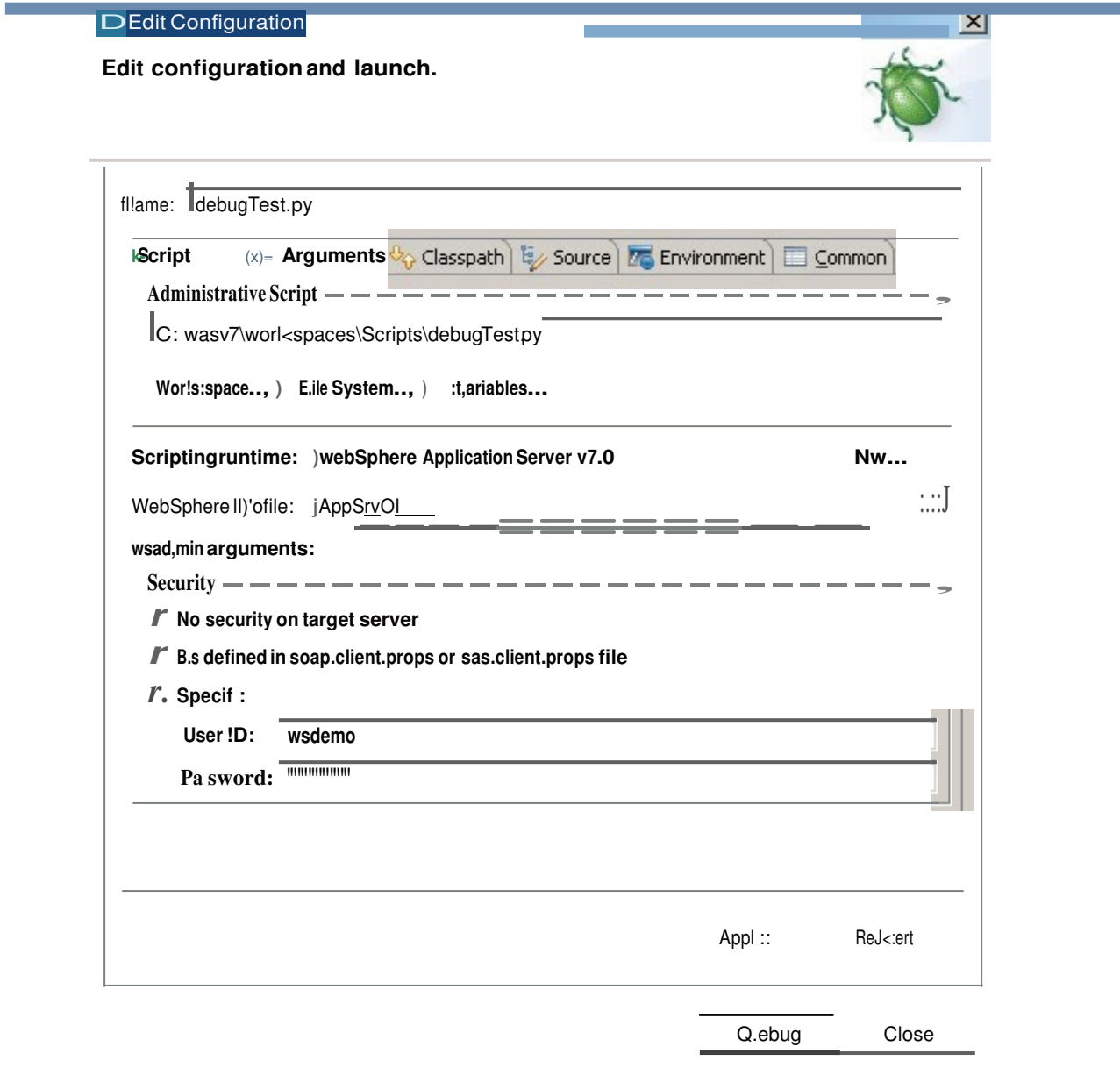Note: you may also set and reset breakpoints from the context menu of the marker bar.

http://www-01.ibm.com/software/support/acceleratedvalue/

```
  1  #
  2  # TODO: enter JYTHON code and save
  3
      Breakpoint set

  6  nodeName = AdminControl.getNode()
  7  print("Node: " + nodeName)
  8
  9  i = 0
 10  fruits = ["apple", "banana", "pear", "orange"]
 11
 12  for fruit in fruits:
            + 1
      Marker bar    i, ") ",  fruit
```

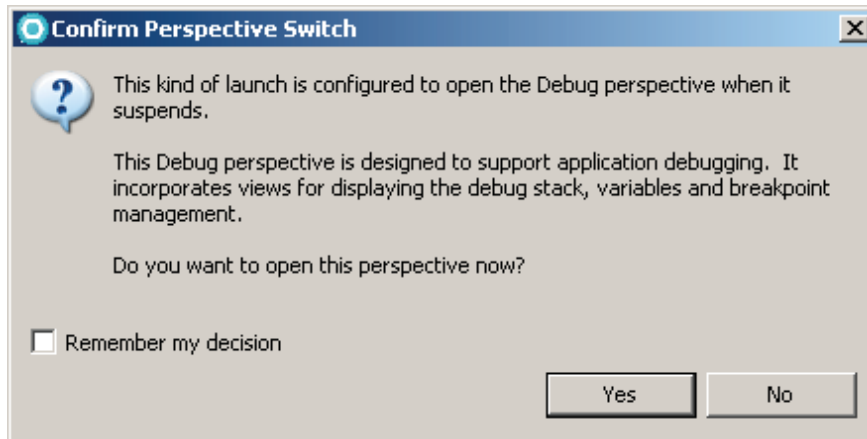_____6.   Select the debugTest.py file on the left view and from its pop-up menu select Debug As ->
         Administrative Script.

http://www-01.ibm.com/software/support/acceleratedvalue/

7. On the Edit Configuration dialog select WebSphere Application Server for the Scripting runtime, select your current WebSphere profile.If security is enabled on the application server enter a valid User Id and Password.

Note: in your environment the WebSphere profile name may be different than the profile name shown below.

Edit Configuration

Edit configuration and launch.

fl!ame: debugTest.py

| Script | (x)= **Arguments** | Classpath | Source | Environment | Common |

Administrative Script

C: wasv7\worl<spaces\Scripts\debugTestpy

Wor!s:space..,) E.ile System..,) :t,ariables…

Scripting runtime:)webSphere Application Server v7.0   Nw…

WebSphere II)'ofile: jAppSrvOI___   :..:]

wsad,min arguments:

Security

r **No security on target server**

r **B.s defined in soap.client.props or sas.client.props file**

r. **Specif :**

User !D: **wsdemo**

Pa sword: """""""""""

Appl ::   ReJ<:ert

Q.ebug   Close

http://www-01.ibm.com/software/support/acceleratedvalue/

____8.   Click Debug. The Confirm Perspective Switch dialog appears.

____9.   Click Yes to switch to the Debug perspective.

The Jython debugger loads and the script starts running. The script continues to run until the first breakpoint is found, or the script ends. In this case the script is paused before line 6 is run.

____10. Once in the debugger you may control the flow of the script with the F5, F6, F7 and F8 keys, or the icons on the toolbar. These keys and icons provide the Step Into, Step Over, Step Return and Resume functionality. Hover the mouse pointer over the toolbar icons to identify them.

____11. On the Source window you see the script and on the Marker bar an arrow shows the next statement to be run.

____12. The Variables view shows the variables currently in scope. Note that in this release you are limited to inspecting variables, their values cannot be changed.

____13. The Breakpoints view allows you to view and alter existing breakpoints.

http://www-01.ibm.com/software/support/acceleratedvalue/

14. The Console view shows any output produced by the script.
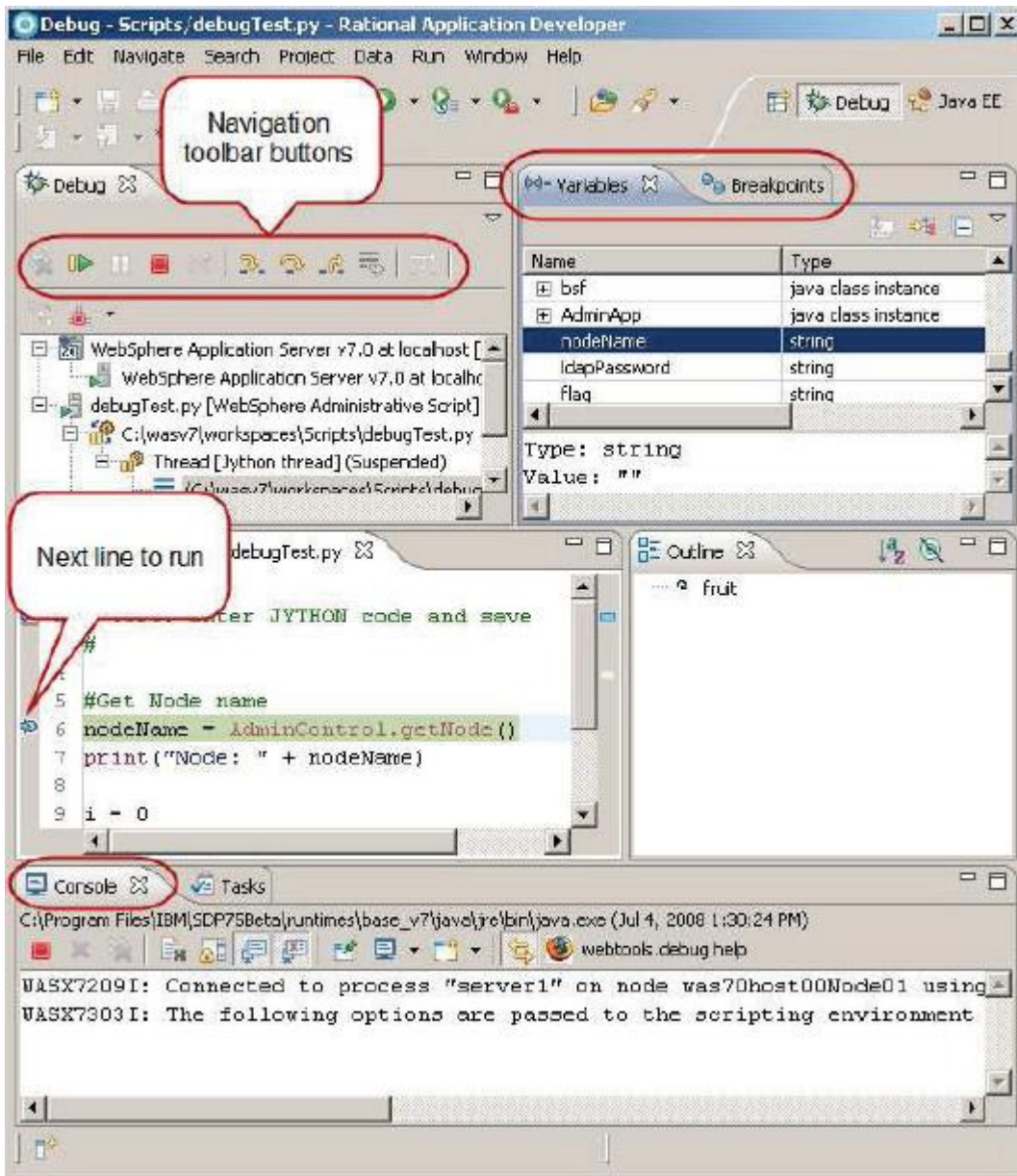
15. Run line 6 by pressing F6 or clicking the Step Over icon. Notice that now the Variables view shows the correct value of the nodename variable.
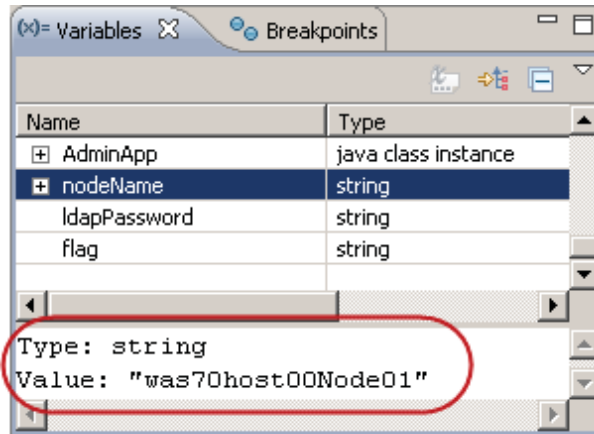
16. Step over the next line, and notice that the print statement runs and its output is displayed on the Console view.

http://www-01.ibm.com/software/support/acceleratedvalue/
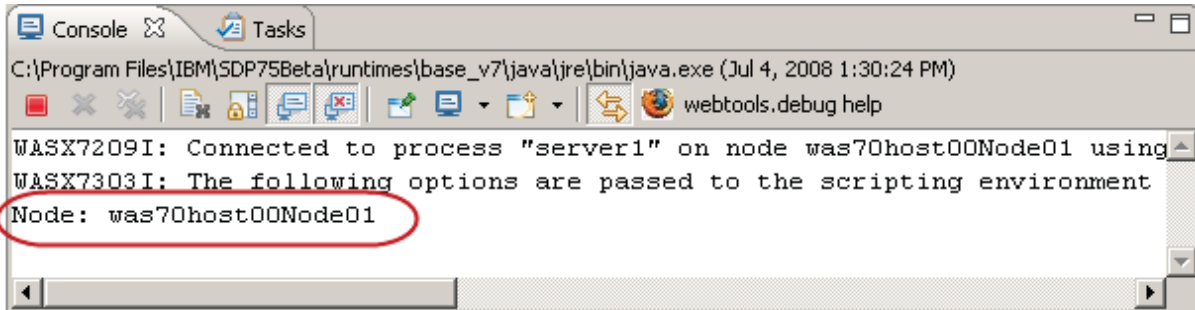
_____17. Step through the script watching the Variables and Console views to see the program flow. After the first pass of the loop your Debug perspective should look like the image below.
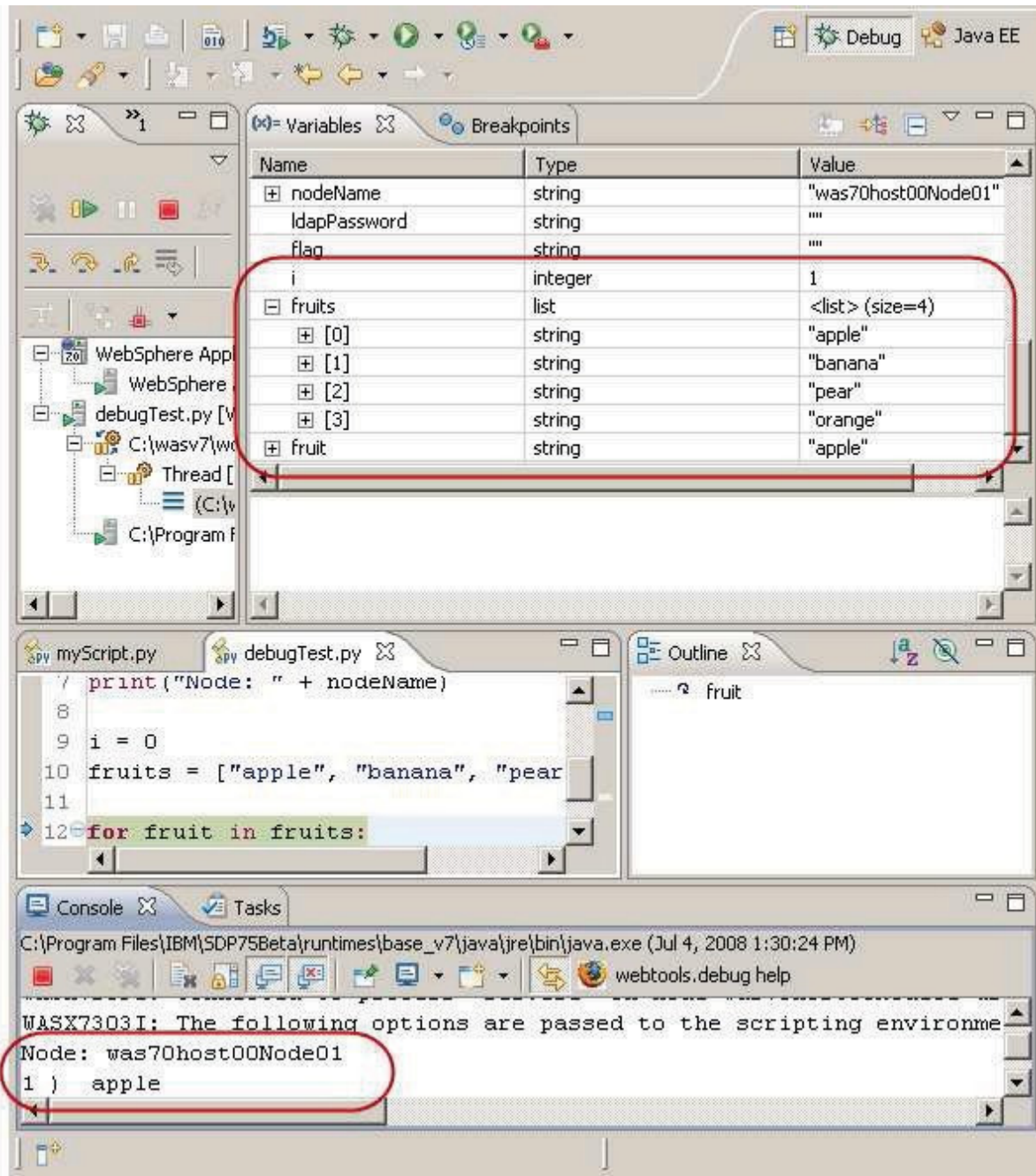
_____18. Continue stepping through the remainder of the script.

_____19. Upon reaching the end of the script the debugger thread terminates.

http://www-01.ibm.com/software/support/acceleratedvalue/

What you did in this exercise

In this exercise you looked at the basics of writing and debugging Jython scripts. You also looked at the administrative objects provided by WebSphere Application Server and the new scripting libraries.

You ran some simple scripts using wsadmin and created, ran and debugged a Jython script from Rational Application Developer.

http://www-01.ibm.com/software/support/acceleratedvalue/