WebSphere Application Server Version 6.1
Sales and Technical Enablement Workshop
Systems Management Lab

What this exercise is about

The objective of this lab is to demonstrate the command line assistance features in the adminconsole in WebSphere Application Server V6.1 and the scripting development environment in the AST.

Lab Requirements

- This exercise assumes that WebSphere Application Server V6.1 and that a cell (consisting of a deployment manager and a managed node with a single application server (server1) has been created with administrative security enabled for the administrative userid “jdoe10” and the password of “jdoe10”.

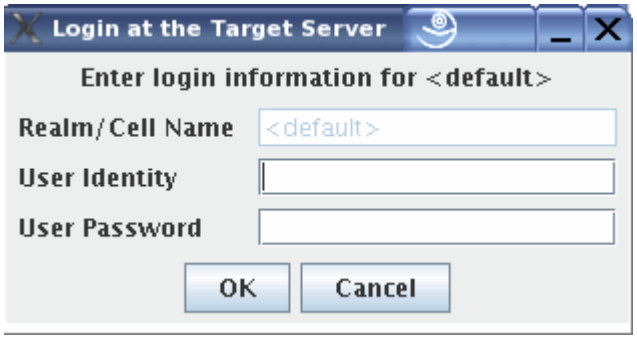
What you should be able to do

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

- Enable and view the output of administrative scripting command in the adminconsole
- Use the administrative scripting command output as the basis for creating wsadmin scripts in the AST.

Part 1: Prepare the Environment

Note: In using the serverStatus command if you do not specify the username and password arguments, and the process is running, you will be prompted with the following dialog



Enter **joe10** for the **User Identity** and **jdoe10** for the **User Password** and click **OK**. If the process is not running the status of the process will be displayed on the command line without requiring authentication

____ 1. Start the deployment manager if necessary

__ a. Navigate to the deployment manager profile's **bin** directory in a command prompt.

```
cd /opt/IBM/WAS61/AppServer/profiles/DMgr01/bin
```

__ b. Use the **serverStatus** command to verify that the deployment manager is stopped.

```
./serverStatus.sh dmgr -username jdoe10 -password jdoe10
```

__ c. If the deployment manager is stopped, use the **startManager** command to start the deployment manager process.

```
./startManager.sh
```

__ d. Look for the following message in the command's output, to indicate successful startup.

```
Server dmgr open for e-business; process id is XXXX.
```

____ 2. Start the node agent if necessary

__ a. Navigate to the managed node profile's **bin** directory in a command prompt.

```
cd /opt/IBM/WAS61/AppServer/profiles/AppSrv01/bin
```

__ b. Use the **serverStatus** command to verify that the node agent is stopped.

```
./serverStatus.sh nodeagent -username jdoe10 -password jdoe10
```

__ c. If the node agent is stopped, use the startNode comment to start the node agent process

```
./startNode.sh
```

__ d. Look for the following message in the command shell output to indicate a successful startup.

```
Server nodeagent open for e-business, process id is XXXX
```

____ 3. Verify that server1 is stopped

__ a. Navigate to the managed node profile's **bin** directory in a command prompt.

```
cd /opt/IBM/WAS61/AppServer/profiles/AppSrv01/bin
```

__ b. Use the **serverStatus** command to verify that the application server is stopped.

```
./serverStatus.sh server1 –username jdoe10 –password jdoe10
```

__ c. If the server status indicates STARTED, then stop the server:

```
stopServer.sh server1 – username jdoe10 –password jdoe10
```

Part 2: Set Up the Console for Command Line Assistance

You will now configure the administration console preferences for command line assistance of some common administration actions.

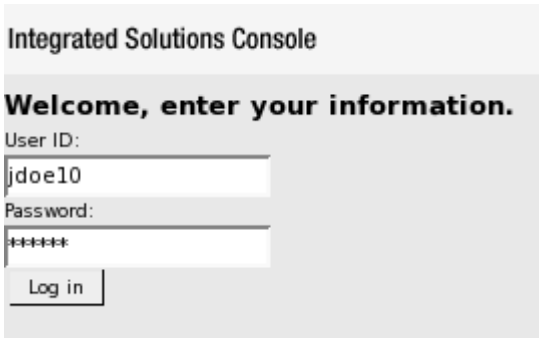
____ 1. From the SLES desktop, locate the panel at the bottom of the workspace and select the Firefox browser icon, and click on the icon as shown below

Note: The browser home page should already be set for https://localhost:9043/ibm/console If the WAS adminconsole doesn't start automatically you will need to enter the URL

____ **2.** When the browser starts if you're presented with the Unknown Authority for the Cert warning as shown below and click on **OK**

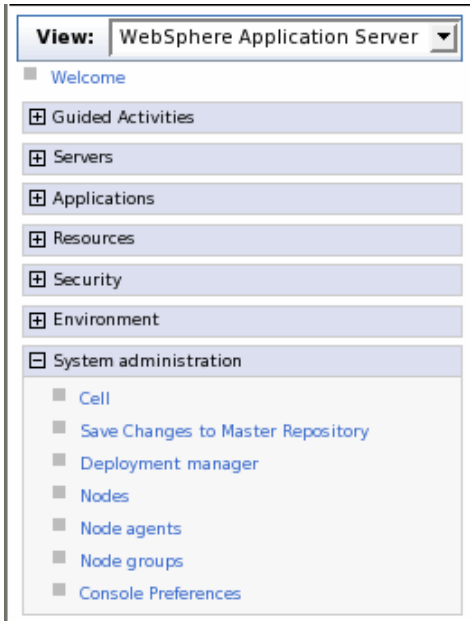
Note: This warning appears because the browser does not have a copy of the self-signed certificate that was created by WAS. If you don't want to see this warning again you can select "**Accept this certificate permanently**" before clicking on **OK.**

____ **3.** Login as the user “jdoe10” with the password “jdoe10” as shown below and click on **Login in**

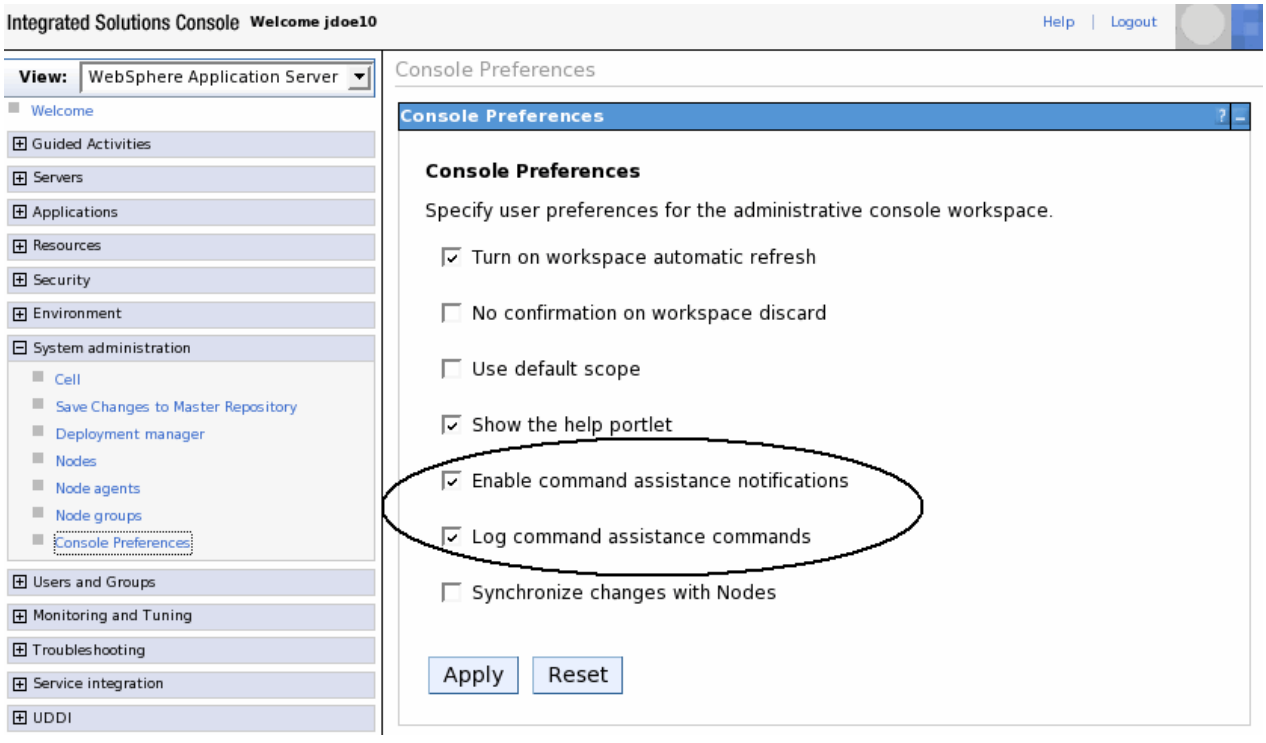


NOTE: While you’ll likely be tempted to do so since the passwords are the same as the User ID, when entering the User ID and password in the console or in command line dialog boxes do not attempt to copy and paste the User ID into the password. In developing this lab material and delivering it, there have been problems using copy and paste for this purpose. It’s not clear if this is related to Linux or VMWare or a combination of the two, but it has occurred often enough and slowed completion of the exercises that it’s best to avoid the problem.

____ **4.** Expand the **Systems Administration** tasks in the left panel, and click on **Console Preferences** as shown below

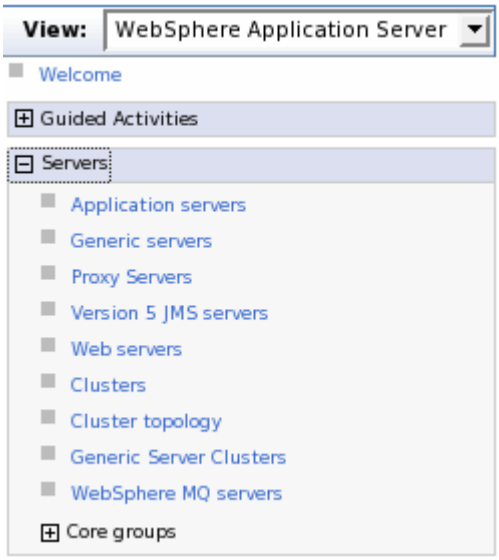


____ **5.** Select **Enable command assistance notifications** and **Log command assistance commands** as shown below and then click on **Apply**



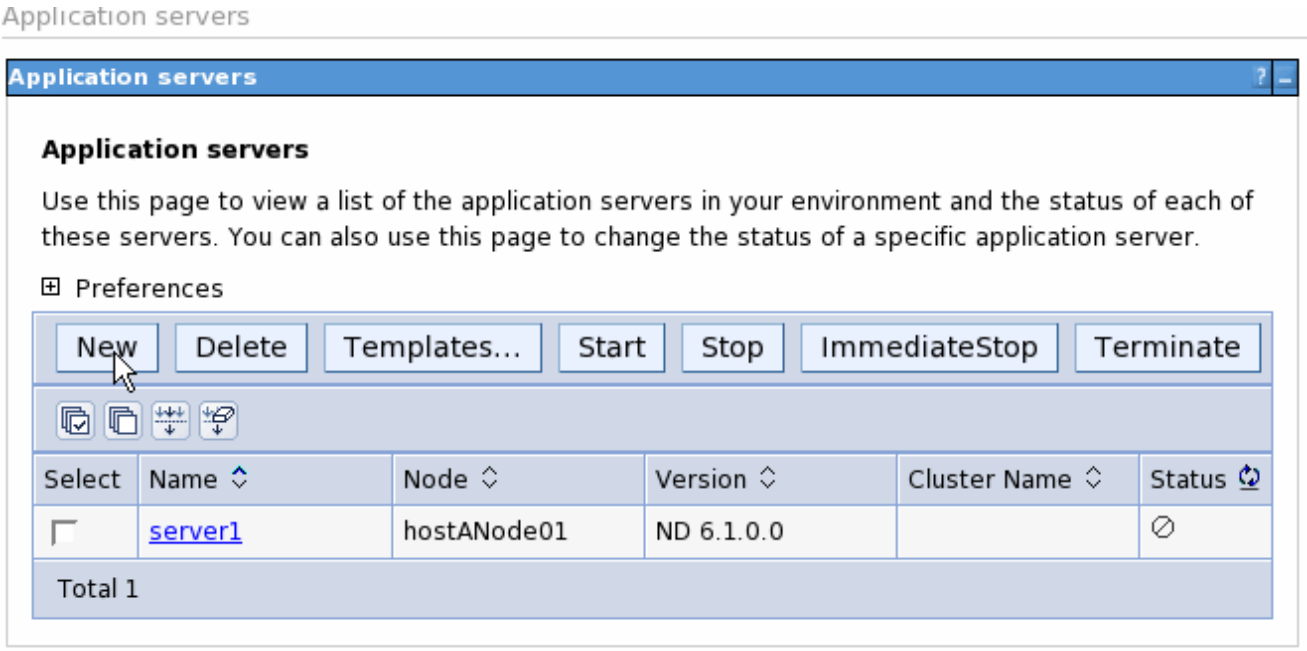
Part 3: Application server creation and application maintenance

____ **1.** In the **Servers** tasks in the left panel, and click on **Application Servers** as shown below



____ 2. Create a new application server

__ a. Click **New** on the right hand side panel.



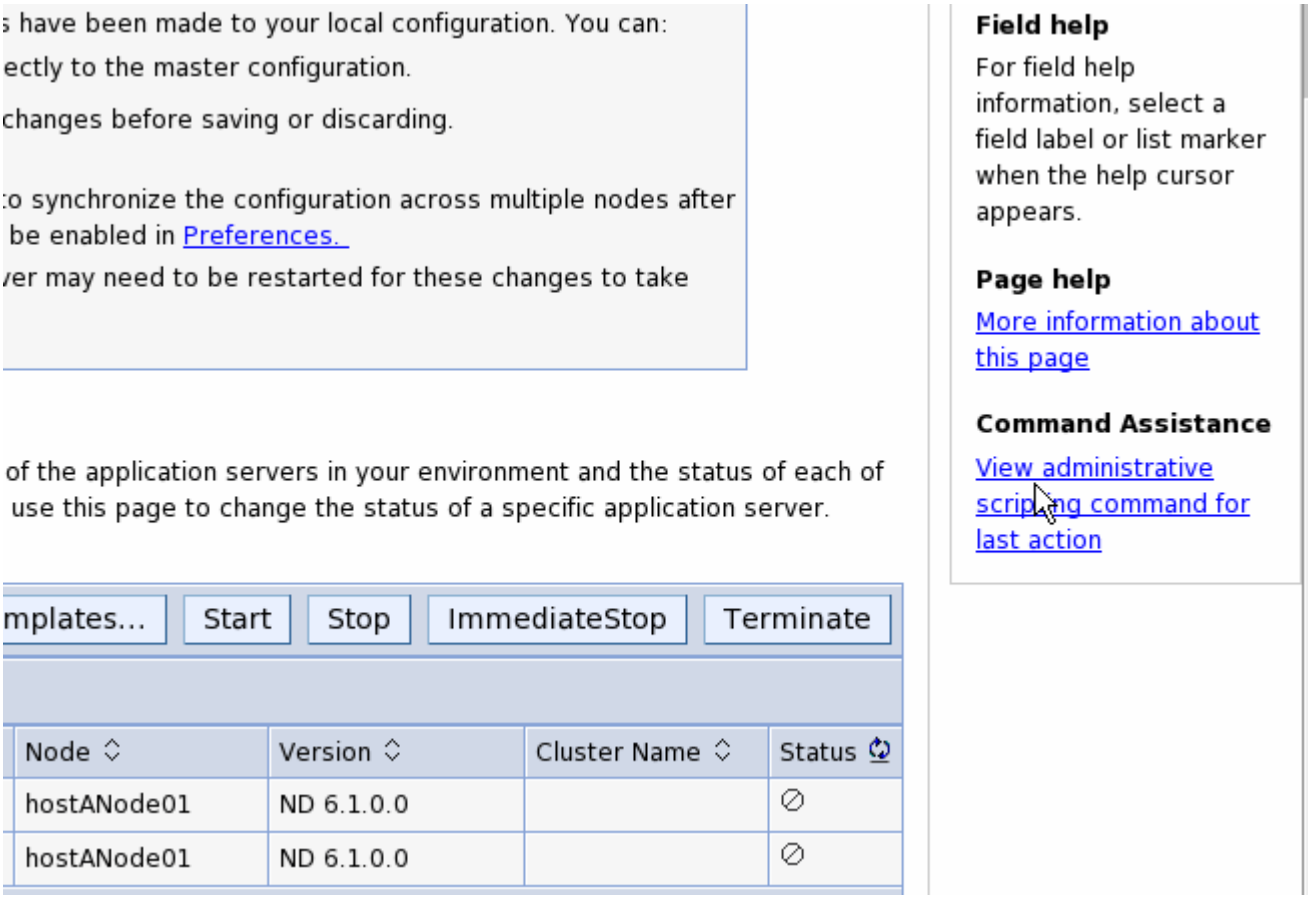
__ b. Enter **STEWServer1** in the server name and click **Next**

__ c. Click **Next** in the Server Template dialog (Accepting the default template)

__ d. Click **Next** in the Server Specific Properties dialog (with the checkbox selected for Generate Unique Ports).

__ e. Click **Finish** on the Confirm new server dialog

__ f. Click **View administrative scripting command for last action** as shown below.



__ g. The results will be shown in a pop up window as shown below

.

Note: While you could cut and paste the command above, since we've also chosen to log the commands we'll be using the logs for creating scripts. Also note that the commands shown above may change in the GA version of WAS V6.1.

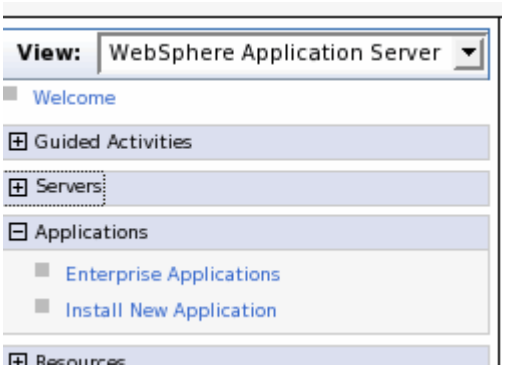
__ h. Close the pop up window

__ i. Repeat steps 2a – 2G for a server named **STEWServer2**

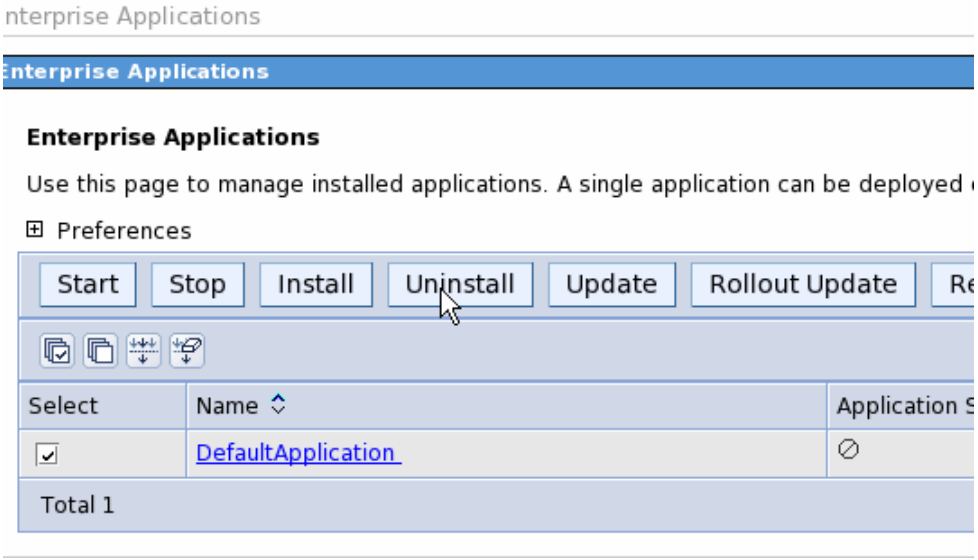
__ j. **Save** your changes

____ 3. Uninstall the Default Application

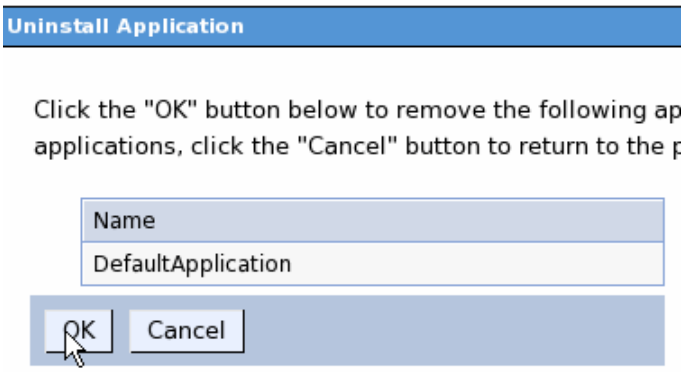
__ **a.** In the **Applications** tasks in the left panel, and click on **Enterprise Applications** as shown below



__ b. Select the **checkbox** for the Default Application and Click **Uninstall**



__ c. Click **OK** to confirm the uninstall as show below



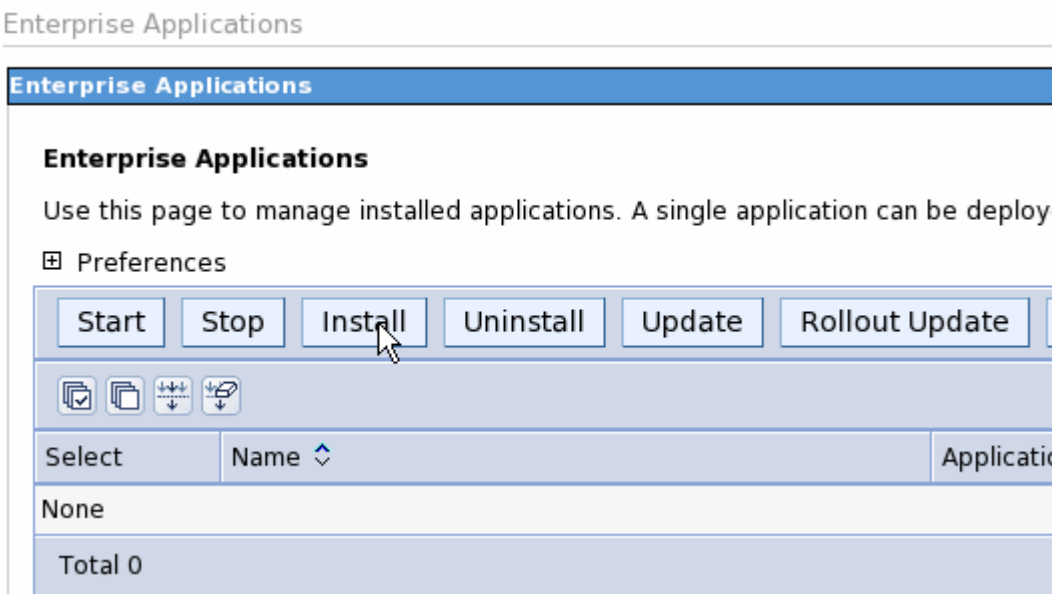
__ d. Click **View administrative scripting command for last action** and view the results. You will notice that AdminApp.list() is shown. That's because the listing of the applications was the last administrative action. The application uninstall was performed prior to the listing of the applications in the console.

__ e. Close the pop window that the scripting command is displayed in

__ f. **Save** your changes

____ **4.** Install the Default Application

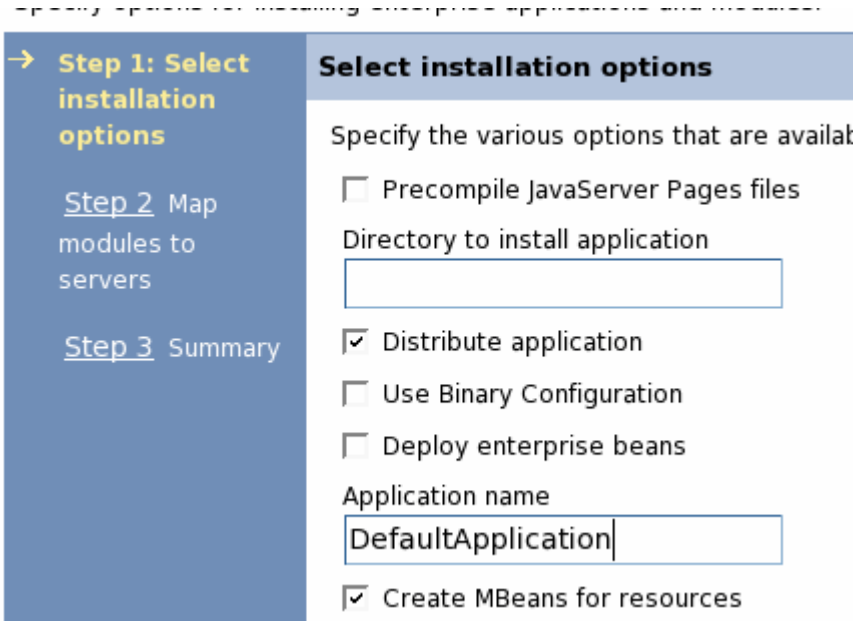
__ **a.** Click **Install** on the Enterprise Applications dialog in the right hand panel as shown below



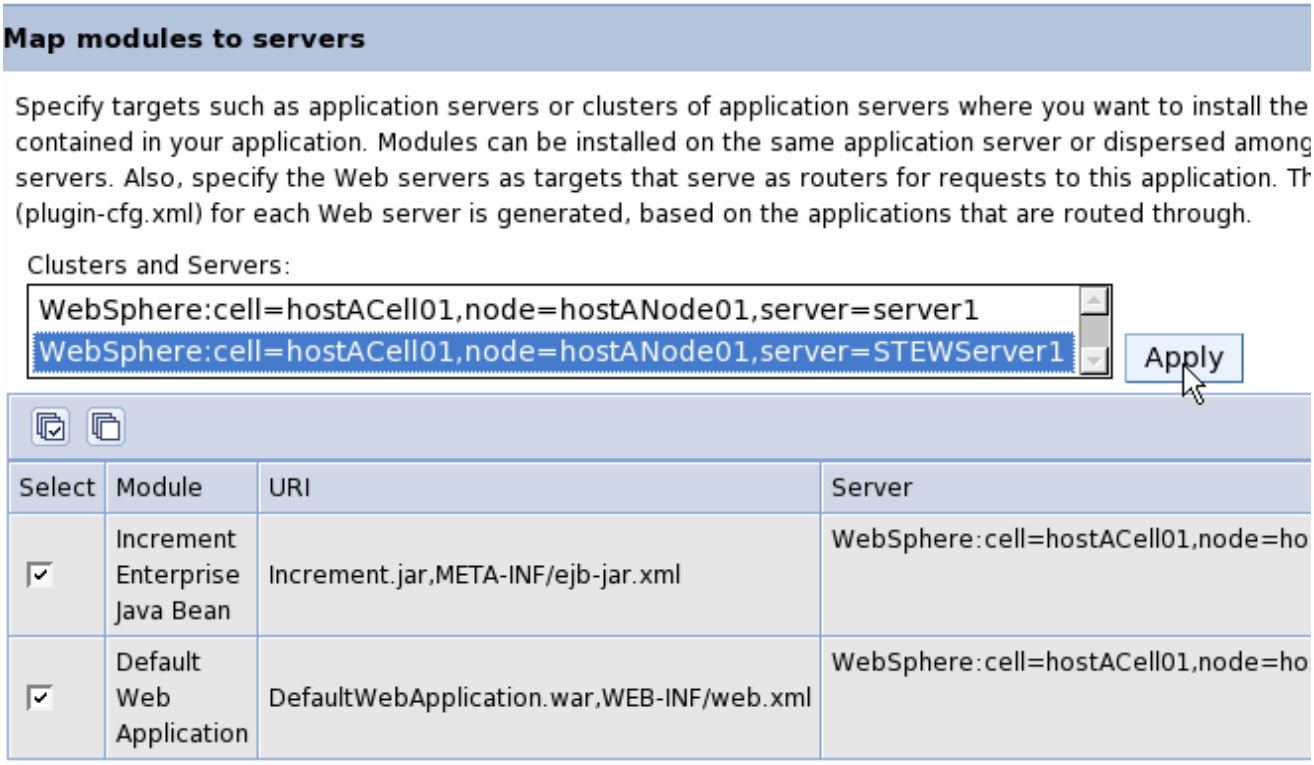
__ b. Using the local file system **browse** to **/opt/IBM/WAS61/AppServer/installableApps/DefaultApplication.ear** and then click on **Open** to close the browse dialog

__ c. Click **Next**

__ d. Enter **DefaultApplication** in the Application name as shown below, then click **Next**



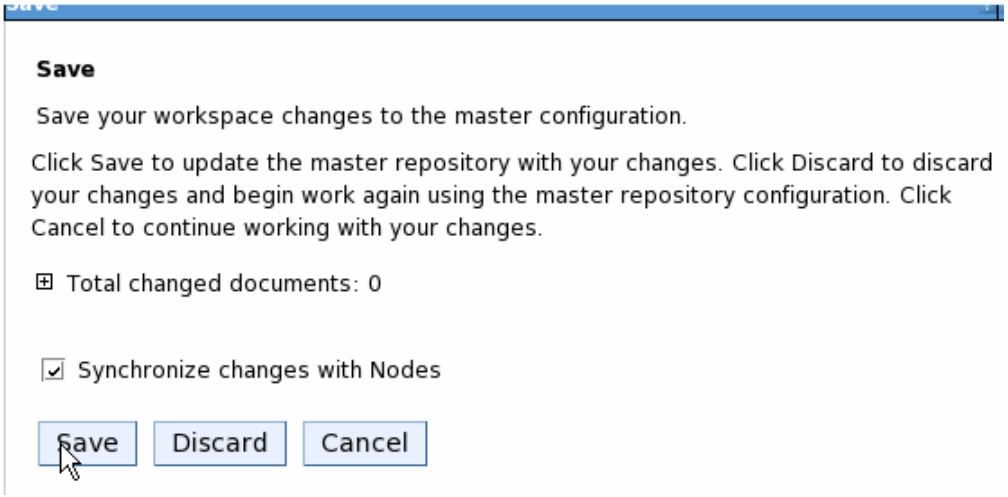
__ e. Map the application to **STEWServer1** by selecting both checkboxes, highlighting **STEWServer1** and clicking **Apply**



__ f. Click **Next**

__ g. Click **Finish** to complete the installation

__ h. **Select** Synchronize changes with Nodes as shown below and **Save** your changes

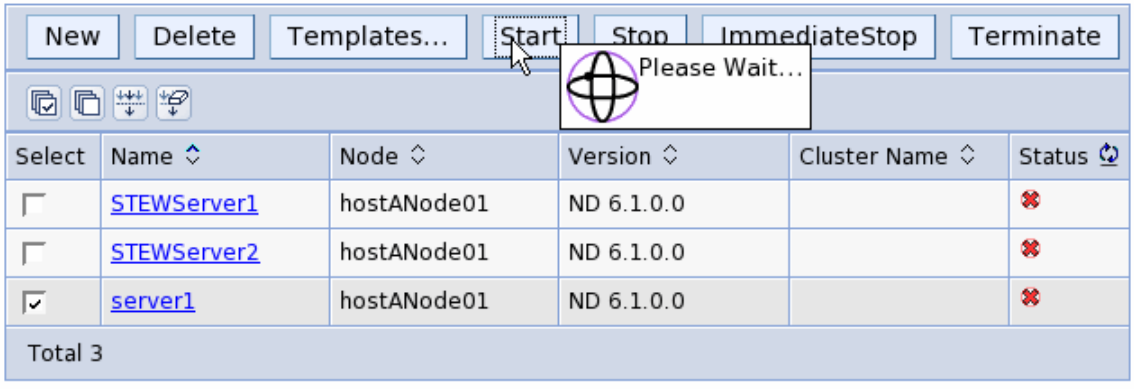


NOTE: The option to **Synchronize changes with Nodes** is no longer the default in the **Save** dialog in WAS V6.1 unlike prior releases. Be sure that you synchronize changes with the nodes in the cell prior to trying to administer objects; otherwise the node configuration will not reflect the configuration in the deployment manager repository.

____ 5. Start and Stop the Default Server

__ **a.** In the **Servers** tasks in the left panel, and click on **Application Servers**

__ b. In the right hand panel **select** Server1 and Click **Start** as shown below



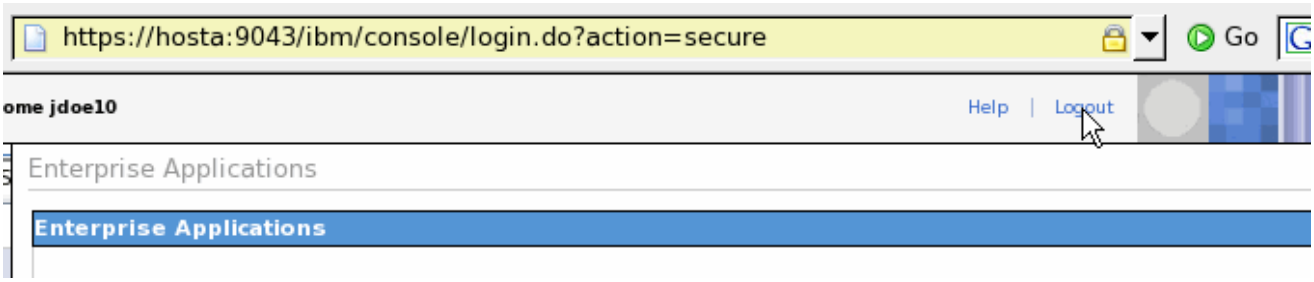
__ c. Wait until the server starts successfully as shown below

__ d. In the right hand panel **select** Server1 and Click **Stop**

__ e. Click **OK** to stop the server

__ f. Click **OK** once the server has stopped.

____ 6. Click **Logout** of the console as shown below and close the browser

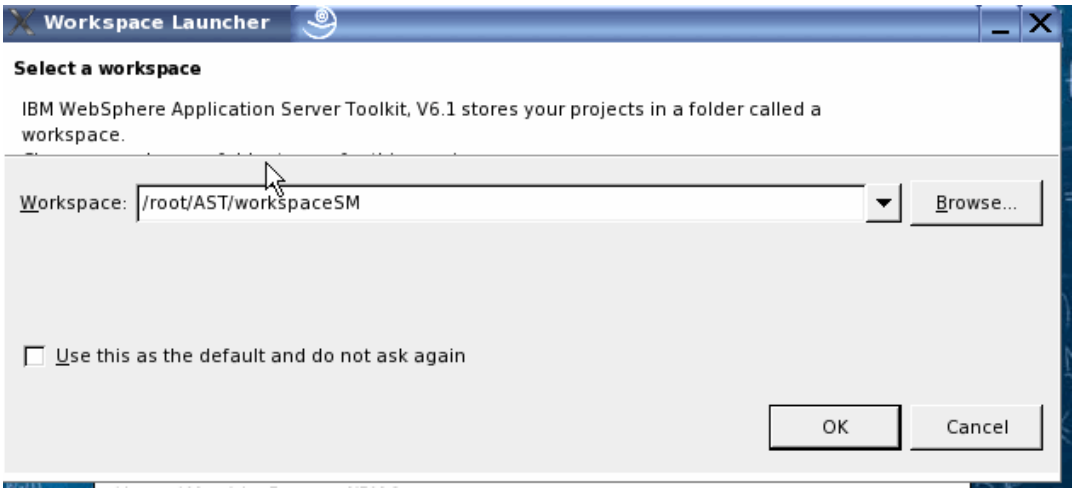


Part 4: Script creation in AST

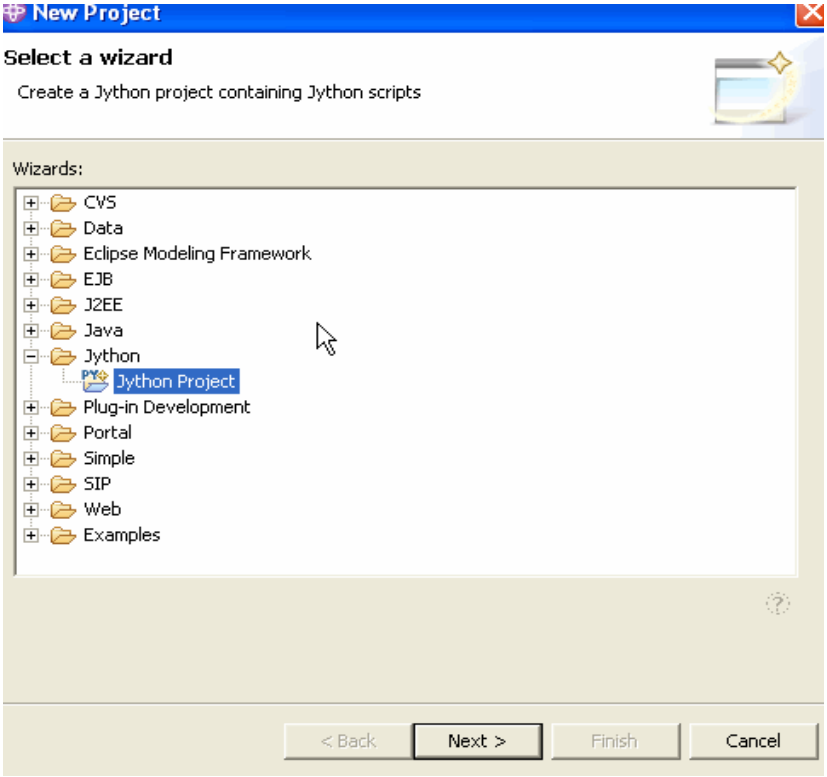
____ 7. Open a command shell and cd to **/opt/IBM/AST61**

__ a. Start the AST with the command ./ast

__ b. Specify **/root/AST/workspaceSM** as shown below

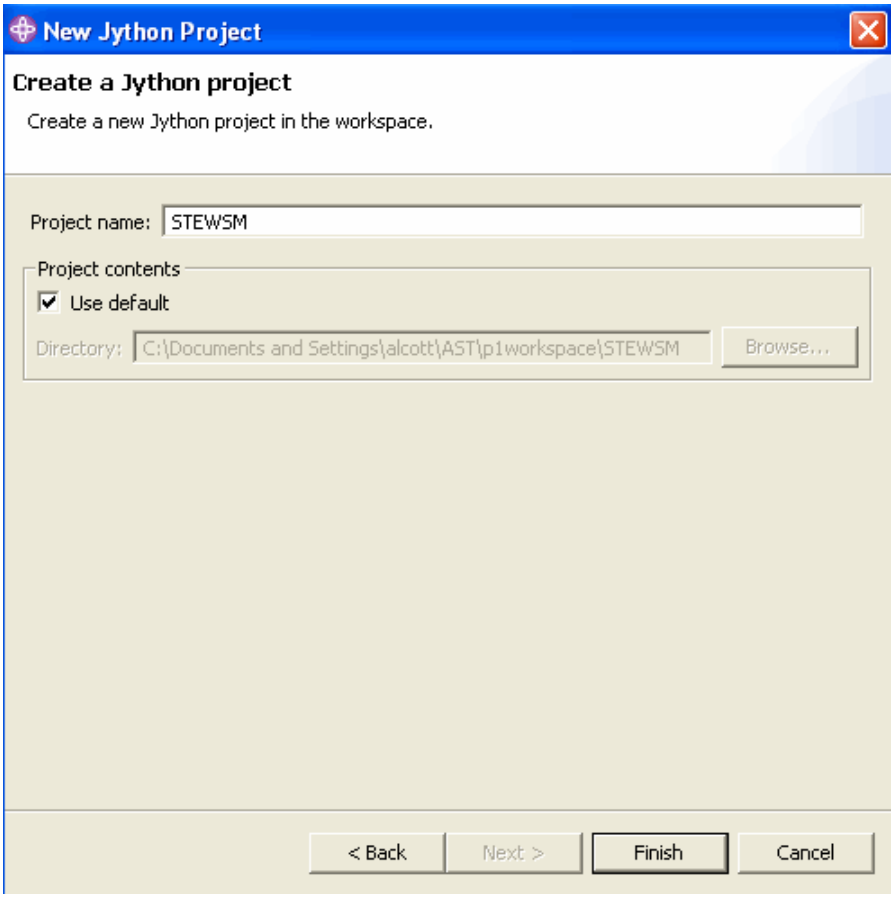


__ **c.** Create a new Jython Project. Select **File -> New Project -> Jython ,** the highlight **Jython Project** as shown below

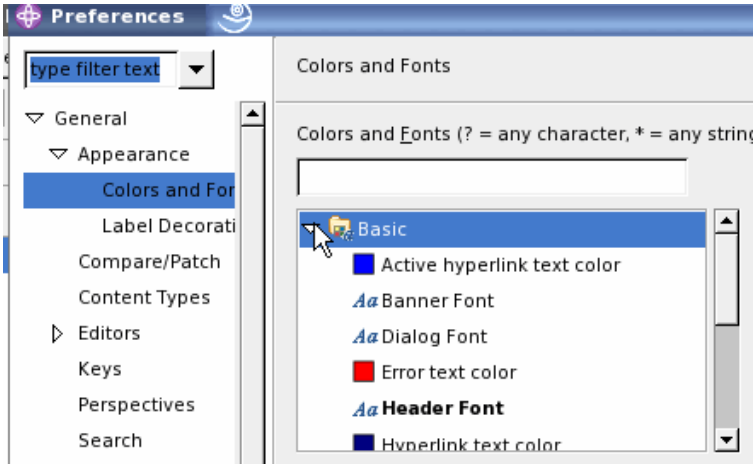


__ d. Click **Next**

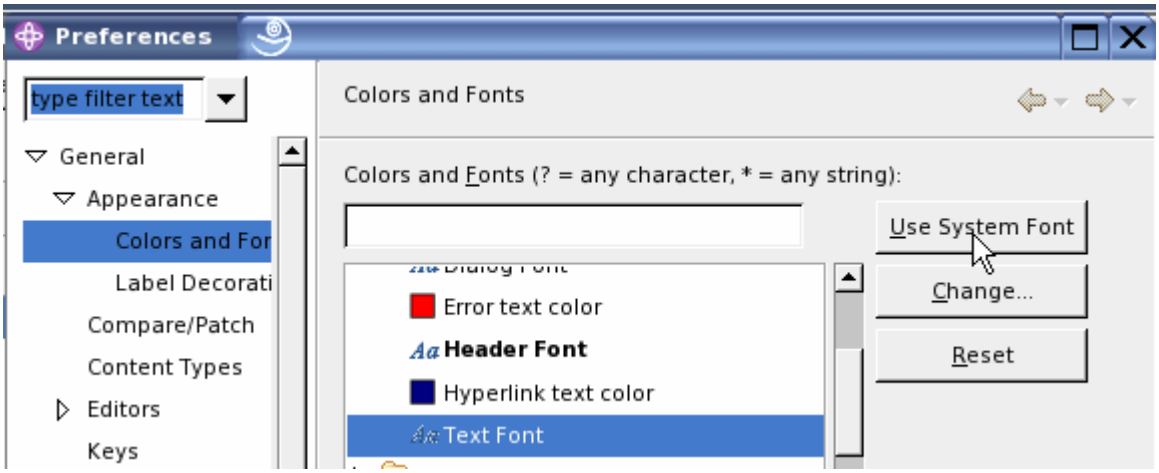
__ e. Enter **STEWSM** in the Project Name field and click **Finish** as shown below



NOTE: In order to improve the readability of the text in the script files you might wish to specify "System Fonts". You can do so by navigating to **Window -> Preferences -> General -> Appearance -> Colors and Fonts** then expand **Basic** as shown below

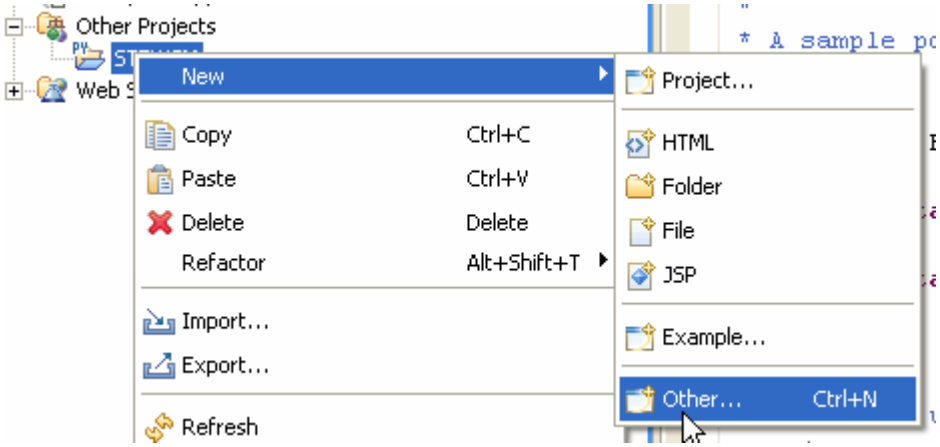


Then highlight TextFont and Click on Use System Font as shown below, then Click **OK.**

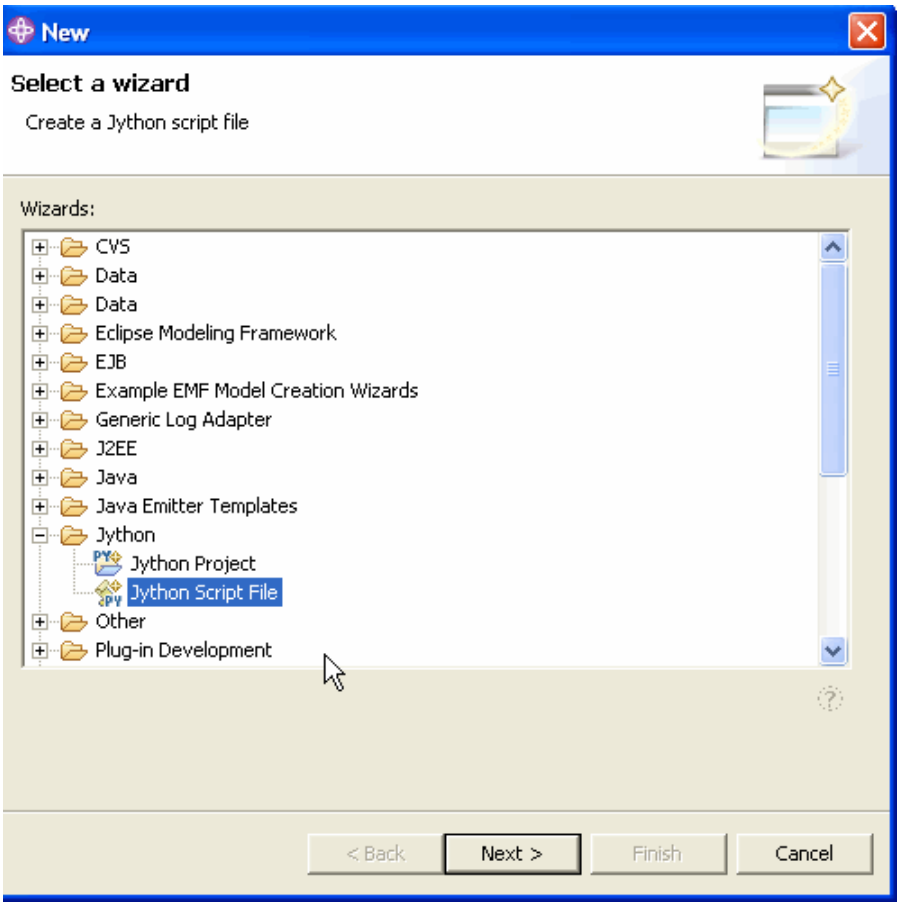


____ 8. Add a script to the Project to Install the Default Application to STEWServer1

__ a. In the Project Explorer perspective Expand **Other Projects,** highlight **STEWSM,** Select **New** and highlight **Other** as shown below.

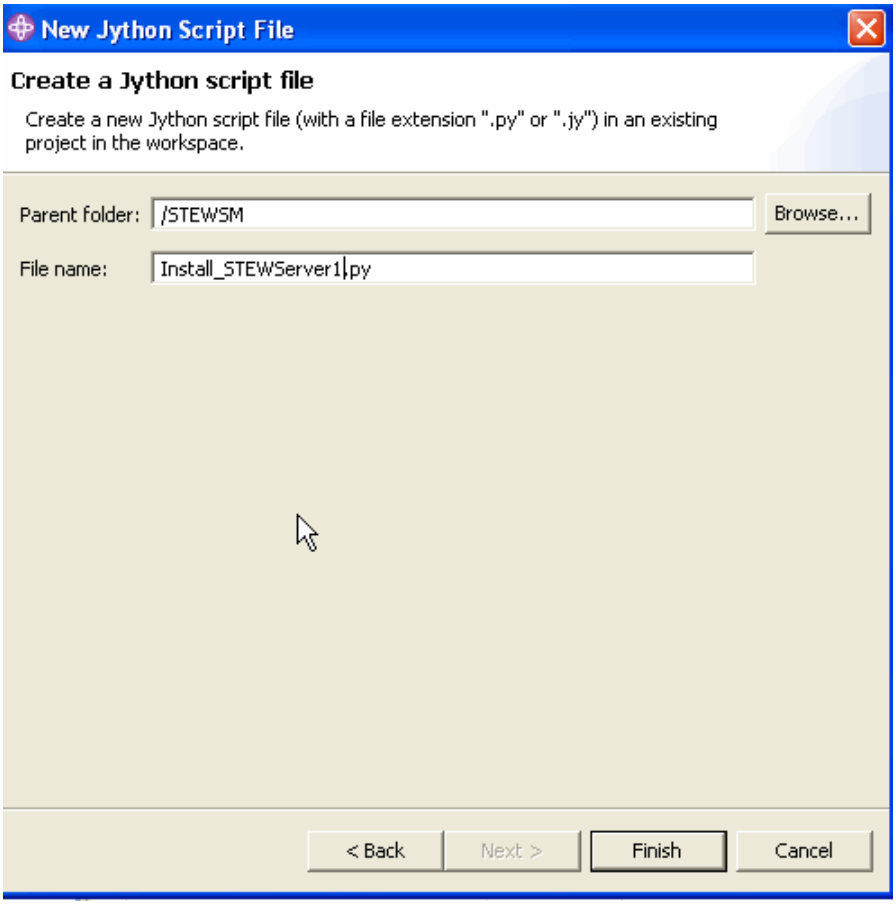


__ b. Select **Jython Script File** from the Wizard dialog as shown below



__ c. Click **Next**

__ d. Enter **Install_STEW1Server.py** in the File Name field as shown below and click **Finish**



__ e. Open a shell and cd to **/opt/IBM/WAS61/AppServer/profiles/Dmgr01/logs/dmgr**

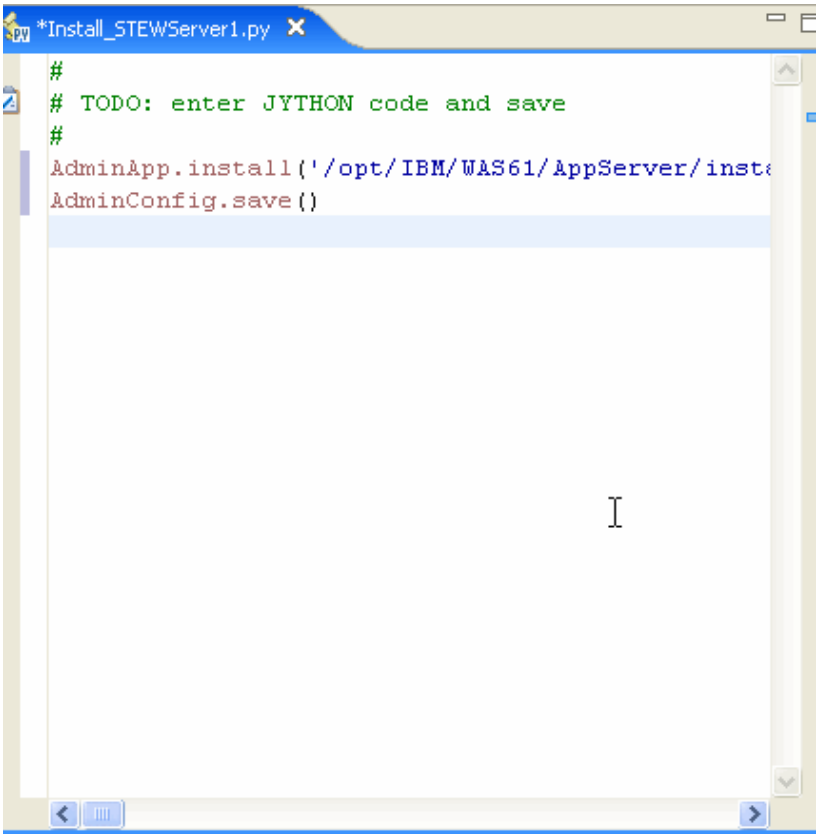
__ f. Use an editor (vi or Kate are available)to open the file **commandAssistanceJythonCommands_jdoe10.log**

__ g. Locate the entry for

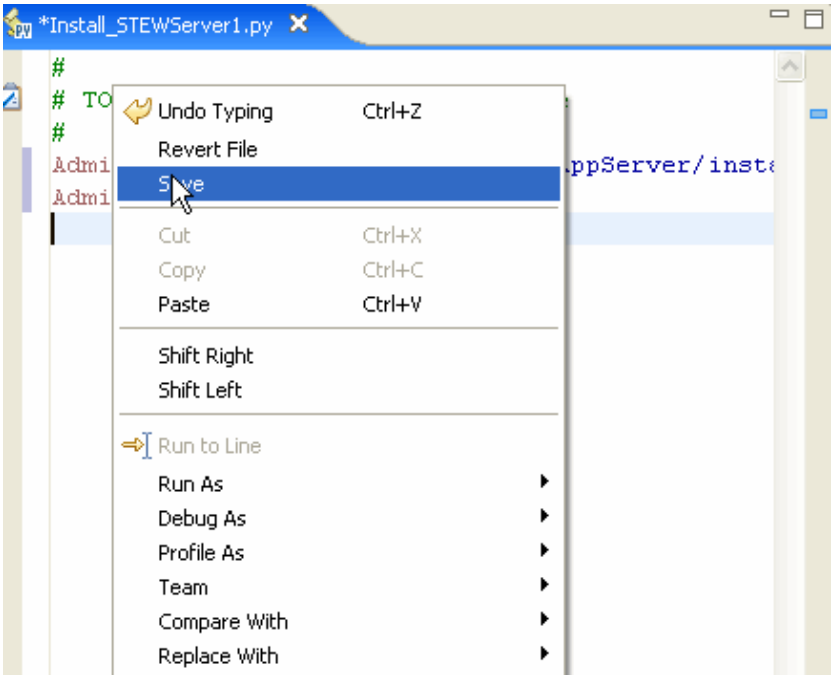
```
AdminApp.install('/opt/IBM/WAS61/AppServer/installableApps/DefaultApplication.ear
', '[  -nopreCompileJSPs -distributeApp -nouseMetaDataFromBinary -nodeployejb -
appname DefaultApplication -createMBeansForResources -noreloadEnabled -nodeployws
-validateinstall warn -noprocessEmbeddedConfig -filepermission
.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755 -noallowDispatchRemoteInclude -
noallowServiceRemoteInclude -MapModulesToServers [[ "Increment Enterprise Java
Bean" Increment.jar,META-INF/ejb-jar.xml
WebSphere:cell=hostACell01,node=hostANode01,server=STEWServer1 ][ "Default Web
Application" DefaultWebApplication.war,WEB-INF/web.xml
WebSphere:cell=hostACell01,node=hostANode01,server=STEWServer1 ]]]' )
```

This is the command that installed the Default Application to STEWServer1

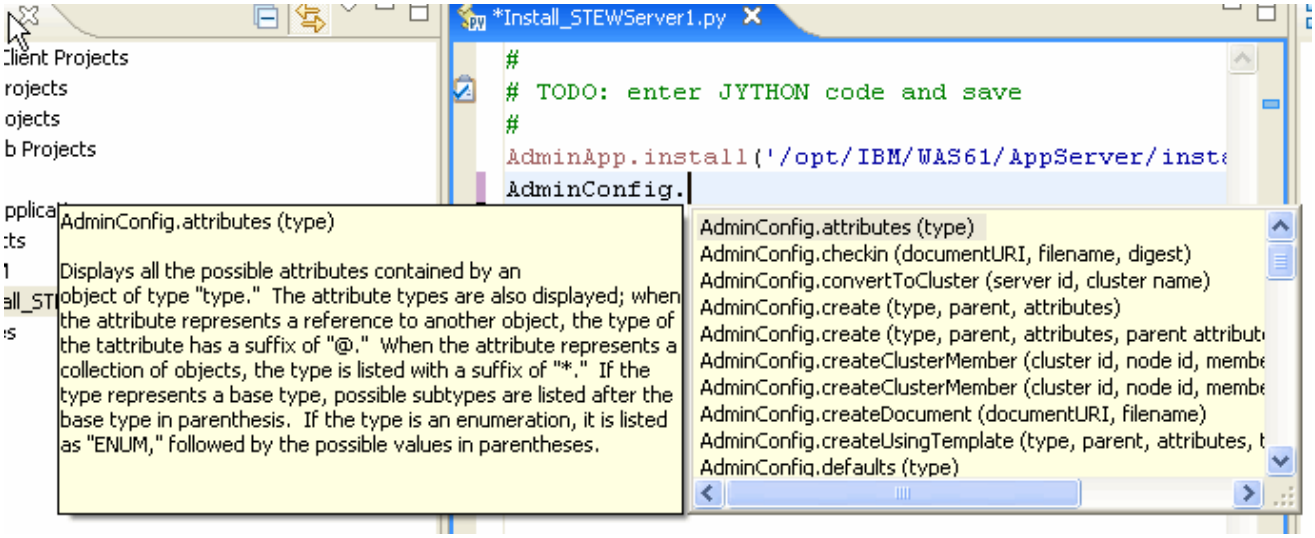
__ h. Copy the command above and the AdminConfig.save() command that follows it and paste it into the Install_STEWServer1.py script in the AST as shown below



__ i. Click the **right mouse button** in the script, which will display the dialog shown below and select **Save**



____ 9. When you create your next scripts in steps 4, 5 and 6 if you start to type a command you and use the CNTL and SPACE BAR keys to see all the syntax and attributes for the wsadmin as shown below.



____ 10. Create a script to uninstall the Default Application

__ a. Repeat steps 2a through 2c above.

__ b. Enter **Uninstall_STEW1Server.py** in the File Name field and click **Finish**

__ c. Enter the commands

AdminApp.uninstall('DefaultApplication')

AdminConfig.save()

__ d. Save your work as before.

Note: You will notice that uninstall does not require the server name as an argument. This is because a given application name can exist only once in a WAS cell. You might also notice that this command is not in the log. As mentioned previously Command Assistance is only available for some of the console actions. The reason for creating two scripts for this purpose is to provide different scripts to different security roles to leverage the WAS V6.1 instance based security and only install and uninstall applications that are in a specific server.

____ 11. Create a script to install the Default Application in STEWServer2

__ a. Repeat steps 2a through 2c above.

__ b. Enter **Install_STEWServer2.py** in the File Name field and click **Finish**

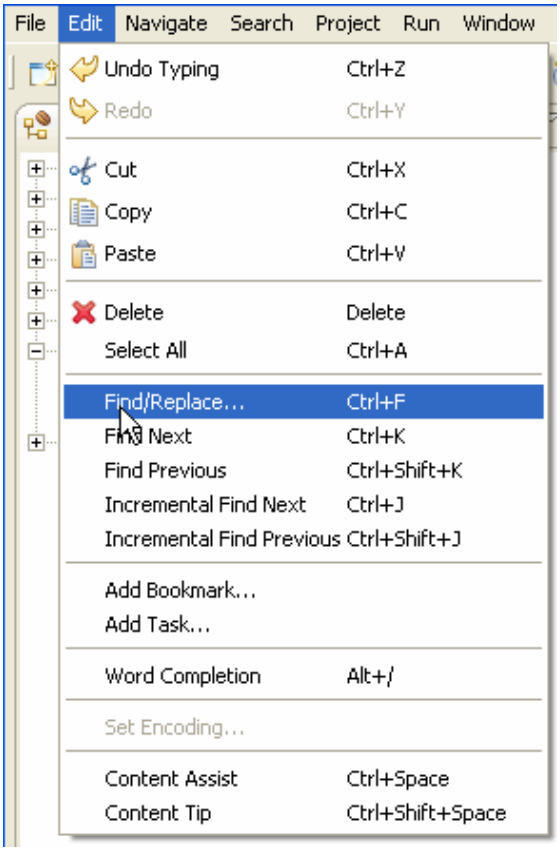
__ c. Return to the shell where you have a copy of **commandAssistanceJythonCommands_jdoe10.log** opened

__ d. Locate the entry for

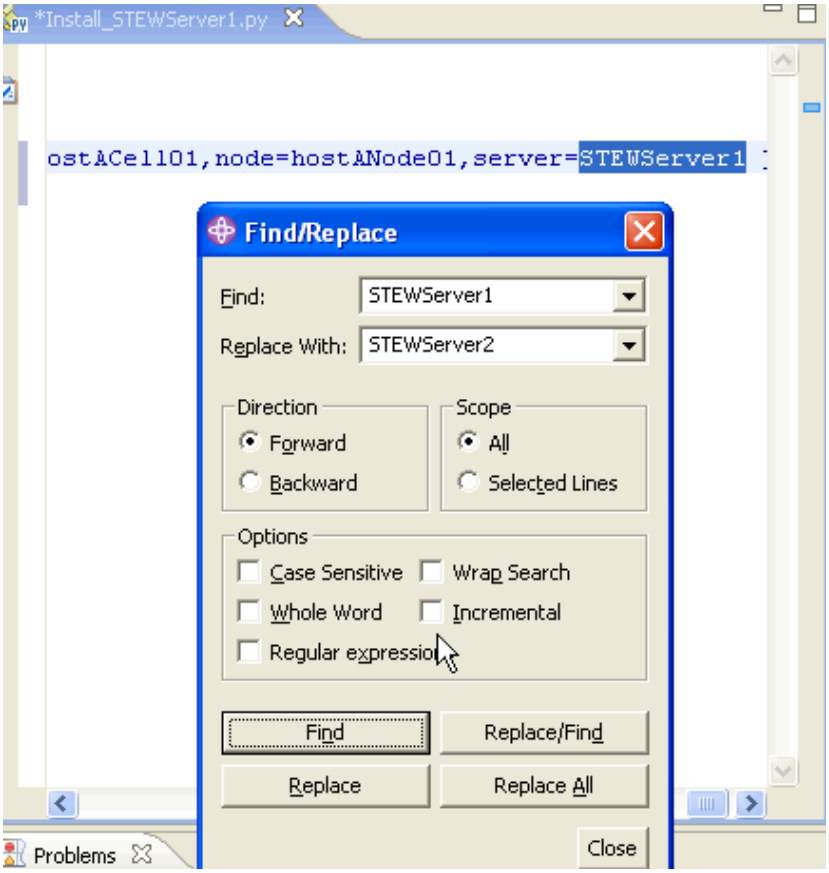
```
AdminApp.install('/opt/IBM/WAS61/AppServer/installableApps/DefaultApplication.ear
', '[  -nopreCompileJSPs -distributeApp -nouseMetaDataFromBinary -nodeployejb -
appname DefaultApplication -createMBeansForResources -noreloadEnabled -nodeployws
-validateinstall warn -noprocessEmbeddedConfig -filepermission
.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755 -noallowDispatchRemoteInclude -
noallowServiceRemoteInclude -MapModulesToServers [[ "Increment Enterprise Java
Bean" Increment.jar,META-INF/ejb-jar.xml
WebSphere:cell=hostACell01,node=hostANode01,server=STEWServer1 ][ "Default Web
Application" DefaultWebApplication.war,WEB-INF/web.xml
WebSphere:cell=hostACell01,node=hostANode01,server=STEWServer1 ]]]' )
```

__ e. As before copy and paste this and the AdminConfig.save() command into the Install_STEWServer2.py script that you created in the AST

__ f. Invoke the Find/Replace dialog **Edit -> Find/Replace** dialog as shown below



__ **g.** Use the Find/Replace dialog to Find **STEWServer1** and Replace it with **STEWServer2.** This occurs in two places, **be sure to change both of these.**



__ **h.** Save your work as before.

____ 12. Create a script to uninstall the Default Application

__ a. Repeat steps 2a through 2c above.

__ b. Enter **Uninstall_STEW2Server.py** in the File Name field and click **Finish**

__ c. Enter the commands

AdminApp.uninstall('DefaultApplication')

AdminConfig.save()

__ d. Save your work as before.

Note: The reason for creating two scripts for this purpose is to provide different scripts to different security roles to leverage the WAS V6.1 instance based security and only install and uninstall applications that are in a specific server.

____ 13. Create a scripts to start and stop a server

__ a. Locate the command below in the log and copy and paste it into a new script named **Start_Server1.py**

```
AdminControl.invoke('WebSphere:name=NodeAgent,process=nodeagent,platform=co
mmon,node=hostANode01,diagnosticProvider=true,version=6.1.0.0,type=NodeAgen
t,mbeanIdentifier=NodeAgent,cell=hostACell01,spec=1.0','launchProcess','[se
rver1]','[java.lang.String]')
```

Note: An alternative, and much shorter, version of the above is

AdminControl.startServer('server1','hostANode01')

Either version will work correctly.

Whichever version, long or short, of the AdminControl syntax you use for this script you should add a "print" in front of the "AdminControl" e.g

print AdminControl.startServer('server1','hostANode01')

Doing so will provide status in the AST console when the command executes

---

__ b. **Save** your work

__ c. Locate the command below in the log and copy and paste it into a new script named **Stop_Server1.py**

```
AdminControl.invoke('WebSphere:name=server1,process=server1,platform=proxy,nod
e=hostANode01,j2eeType=J2EEServer,version=6.1.0.0,type=Server,mbeanIdentifier=
cells/hostACell01/nodes/hostANode01/servers/server1/server.xml#Server_11491721
01764,cell=hostACell01,spec=1.0,processType=ManagedProcess', 'stop')
```

Note: An alternative, and much shorter, version of the above is

AdminControl.stopServer('server1,'hostANode01')

Either version will work correctly.

Whichever version, long or short, of the AdminControl syntax you use for this script you should add a "print" in front of the "AdminControl" e.g.

print AdminControl.stopServer('server1,'hostANode01')

Doing so will provide status in the AST console when the command executes

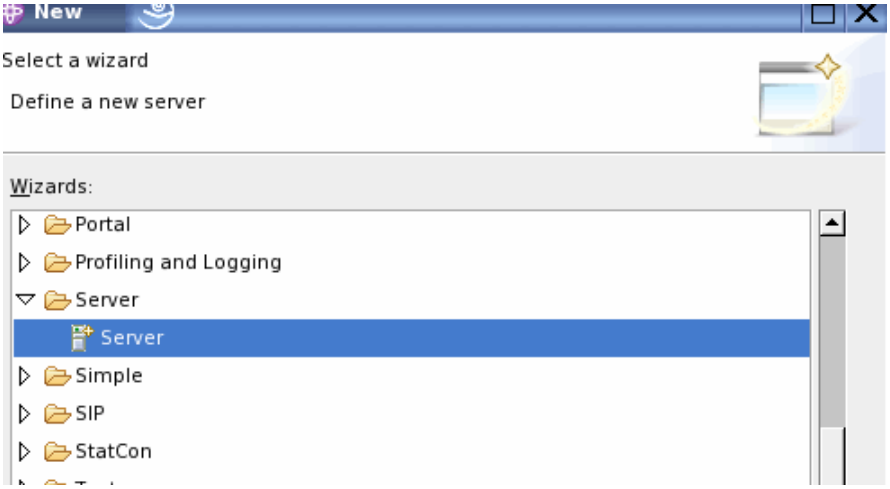
---

__ d. **Save** your work

Part 5: Script Execution/Testing in AST

____ 1. Define a external server in the AST to run the scripts in.

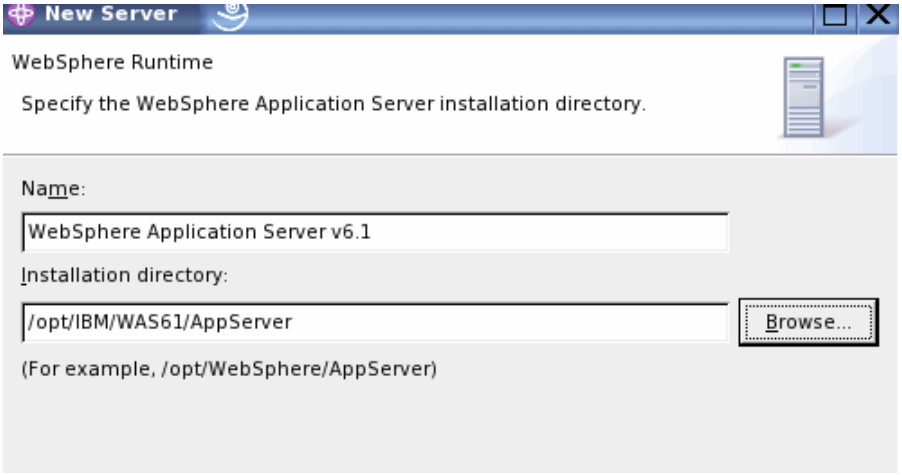
__ **a.** Create a new Server, Select **File -> New -> Other -> Server ->** and highlight **Server** as shown below



__ **b.** Click **Next**

__ **c.** Click **Next**

__ **d. Browse** to /**opt**/**IBM**/**WAS61**/**AppServer** as shown below



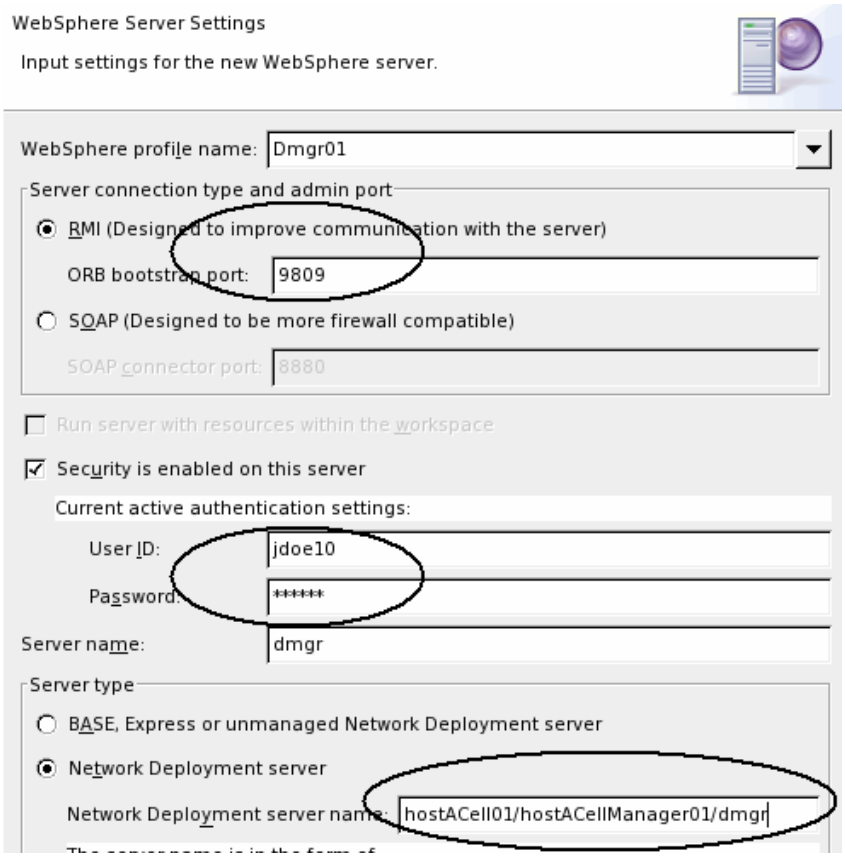
__ **e.** Click **Next**

__ **f.** Fill in the values as shown below

1) 9809 for the **ORB bootstrap port**

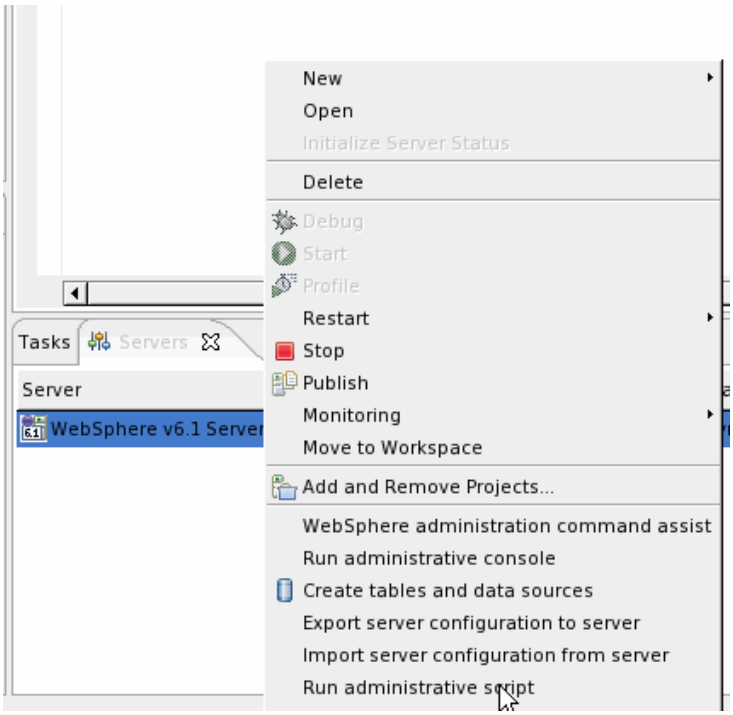
2) jdoe10 for the **User ID** and **password**

3) hostACell01/**hostACellManager01**/**dmgr** for the **Network Deployment server name**

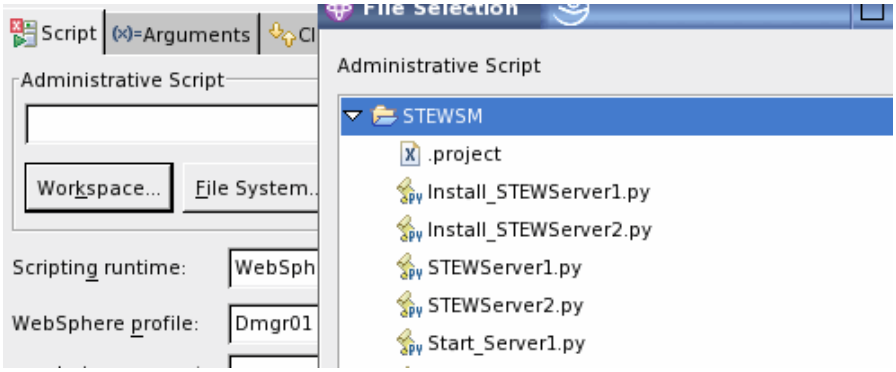


__ **g.** Click **Finish**

____ **2.** **Highlight** the WebSphere v6.1 Server, then click **Run Administrative script**



__ **a.** Click **Workspace,** the expand **STEWSM** and select **Start_Server1.py**



__ **b.** Click **OK**

__ **c.** Select **Specify** for Security and Click **Apply** as shown below

__ **d.** Click **Run**

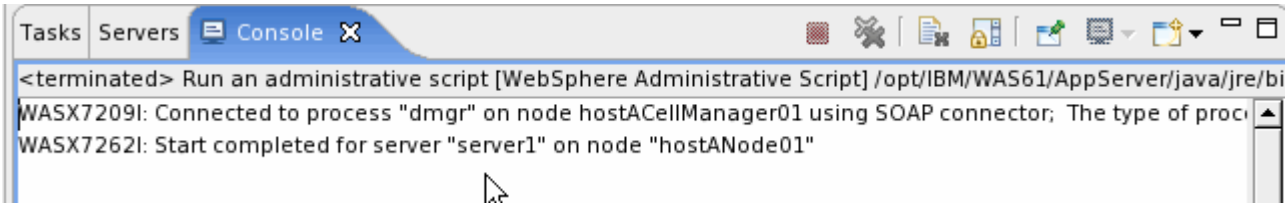
__ **e.** The Console should become visible as shown below and once server1 has started the second message will be displayed if you placed a “print” before “AdminControl”.

Note: If you use the shorter version of the command

AdminControl.startServer(‘server1,’hostANode01’)

Then you will see the message below. If you use the long version with uses “AdminControl.invoke” then all you will see displayed is “true” when AdminControl.invoke completes

---



Note: You may see some number of messages “sys-package-msg processing new jar <file name>” as shown below. This is normal the first time you run an administrative command.

```
WASX7209I: Connected to process "dmgr" on node hostACellManager01 using SOAP connector;  The type of process is: Deployr
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/optionalLibraries/jython/jython.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/lib/startup.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/lib/bootstrap.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/lib/j2ee.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/lib/lmproxy.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/lib/urlprotocols.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/java/lib/tools.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/deploytool/itp/batchboot.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/deploytool/itp/batch2.jar'
*sys-package-mgr*: processing new jar, '/opt/IBM/WAS61/AppServer/plugins/com.ibm.ws.runtime.coregroupbridge_6.1.0.jar'
```

____ **3.** Repeat steps 2a – 2e for the script **Stop_Server1.py**

What you did in this exercise

In Parts 2 and 3 you learned how to enable and view the command assistance features in WAS V6.1 and installed an application using the improved installation dialogs. In Parts 4 and 5 you used the output to create Jython scripts in the AST using the command assistance output and to test them.