



IBM Software Group

# WebSphere Administration writing simple Jython scripts



Jonathan Ricci - [jonricci@us.ibm.com](mailto:jonricci@us.ibm.com)  
Accelerated Value Leader / Specialist  
date: 7/28/2010



AVP Expert Call Series



# Agenda

- WebSphere administration and configurations
- What is wsadmin
  - ▶ AdminControl
  - ▶ AdminConfig
  - ▶ AdminApp
  - ▶ AdminTask
- Wsadmin Interface
- Initiating a script
- Creating Jython Scripts
  - ▶ Using Command Assistance
  - ▶ Modifying script for reuse
- Questions

# WebSphere administration and configuration

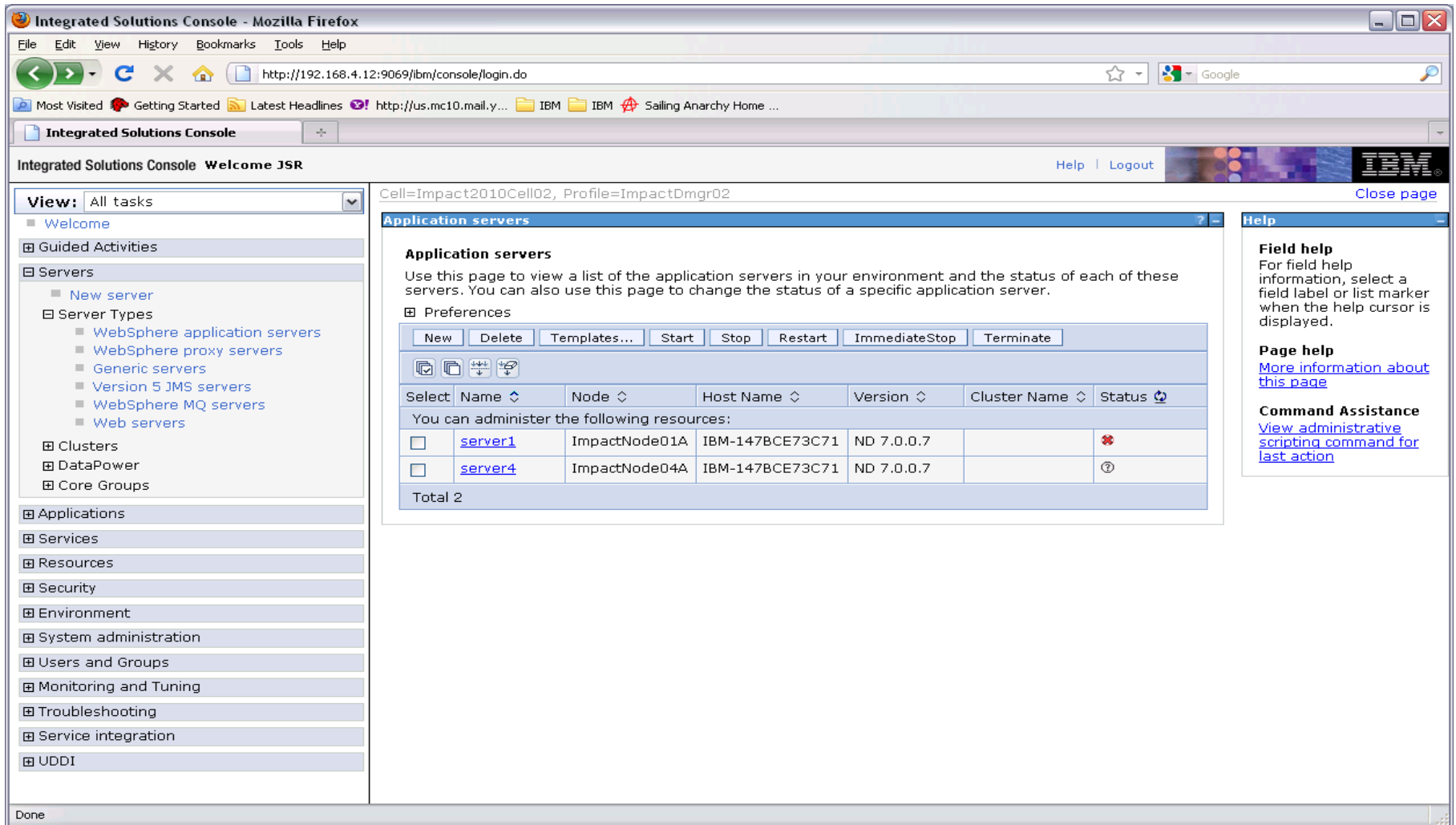
- WebSphere Administration
  - ▶ Administrative Console
  - ▶ Scripting
- Administrative Console
  - ▶ GUI
  - ▶ Changes easily made
  - ▶ View topology
  - ▶ Slow
  - ▶ Prone to typographic errors
  - ▶ Process repeatability
- Administrative Scripting
  - ▶ Speed
  - ▶ Repeatability
  - ▶ Many changes in one script
  - ▶ Reuse
- WebSphere configuration repository
  - ▶ XML based



Configuration files change from version to version

NEVER edit these files!

# WebSphere Version 7 Administrative Console



Integrated Solutions Console - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.4.12:9069/ibm/console/login.do

Integrated Solutions Console

Integrated Solutions Console Welcome JSR Help Logout

View: All tasks

- Welcome
- Guided Activities
- Servers
  - New server
  - Server Types
    - WebSphere application servers
    - WebSphere proxy servers
    - Generic servers
    - Version 5 JMS servers
    - WebSphere MQ servers
    - Web servers
  - Clusters
  - DataPower
  - Core Groups
- Applications
- Services
- Resources
- Security
- Environment
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Cell=Impact2010Cell02, Profile=ImpactDmgr02

Application servers

**Application servers**

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

Preferences

New Delete Templates... Start Stop Restart ImmediateStop Terminate

| Select                   | Name    | Node          | Host Name       | Version    | Cluster Name | Status |
|--------------------------|---------|---------------|-----------------|------------|--------------|--------|
| <input type="checkbox"/> | server1 | ImpactNode01A | IBM-147BCE73C71 | ND 7.0.0.7 |              | ✖      |
| <input type="checkbox"/> | server4 | ImpactNode04A | IBM-147BCE73C71 | ND 7.0.0.7 |              | ⓘ      |
| Total 2                  |         |               |                 |            |              |        |

**Field help**

For field help information, select a field label or list marker when the help cursor is displayed.

**Page help**

[More information about this page](#)

**Command Assistance**

[View administrative scripting command for last action](#)

Done



# WebSphere application server.xml

5

# XML Elements

## ■ WebContainer Services element

```
<services xmi:type="applicationserver.webcontainer:SessionManager" xmi:id="SessionManager_1268332207137"
  enable="true"
  enableUrlRewriting="false"
  enableCookies="true"
  enableSSLTracking="false"
  enableProtocolSwitchRewriting="false"
  sessionPersistenceMode="NONE"
  enableSecurityIntegration="false"
  allowSerializedSessionAccess="false"
  maxWaitTime="5"
  accessSessionOnTimeout="true">
  ...
  ...
  ...
</services>
```

### ▶ Note the following:

- Start/end of element
- xmi:type - applicationserver.webcontainer:SessionManager
- xmi:id - SessionManager\_1268332207137

- Attributes

Not all elements contain xmi:type

## What is wsadmin

- Scripting languages for the WebSphere Application Server
  - ▶ Jython
  - ▶ Jacl (default)
- Jython is a java based version of the Python scripting language
- Wsadmin scripting interface for WebSphere V7.0 is based on 5 objects
  - ▶ **AdminControl**: Use to run operational commands.
  - ▶ **AdminConfig**: Use to run configurational commands
  - ▶ **AdminApp**: Use to administer applications.
  - ▶ **AdminTask**: Use to run administrative commands.
  - ▶ **Help**: Use to obtain general help.
- Each of these objects contains a number of functions that can be used to manipulate, view, add or remove environment configuration artifacts
- Configuration types – the wsadmin tool contains over 600 different object types that can be managed
- Mbeans – the wsadmin tool uses WebSphere Mbeans to manage the

environment

## wsadmin AdminControl

- The AdminControl object is used to invoke operational commands on an active environment runtime
- Command examples
  - ▶ Start Server1 in 100 seconds

```
wsadmin>AdminControl.startServer('server1', 100)
```

- ▶ The following lines will synchronize node “MyNode”

```
wsadmin>nodeObj=AdminControl.completeObjectName('type=NodeSync,node=MyNode,*')
```

```
wsadmin>AdminControl.invoke(nodeObj,'sync')
```

- List of all AdminControl commands, infocenter link

- ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome\\_nd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome_nd.html)



## wsadmin AdminConfig

- The AdminConfig object is used to invoke configuration commands and to create or change elements of the WebSphere Application Server configuration
- Command examples

- ▶ List servers

```
wsadmin>print AdminConfig.list('Server')
```

- ▶ Enable HA Manager

```
wsadmin>ha = {HA Manager configuration object for server }  
wsadmin>AdminConfig.modify(ha, [{"enable", setting}])
```

- List of all AdminConfig commands, infocenter link

- ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome\\_nd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome_nd.html)



## wsadmin AdminApp

- The AdminApp object is used to install, modify, and administer applications.
- Command Examples

- ▶ Install and application

```
wsadmin>print AdminApp.install('/apps/myapp.ear')
```

- ▶ uninstall an application

```
wsadmin>print AdminApp.uninstall('myApp')
```

- Sample output

```
ADMA5017I: Uninstallation of myapp started.  
ADMA5104I: Server index entry for myCellManager was updated successfully.  
ADMA5102I: Deletion of config data for myapp from config  
repository completed successfully.  
ADMA5011I: Cleanup of temp dir for app myapp done.  
ADMA5106I: Application myapp uninstalled successfully.
```

- List of all AdminApp commands, infocenter link

- ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome\\_nd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome_nd.html)



## wsadmin AdminTask

- The AdminTask object is used to access a set of task-oriented administrative commands that provide an alternative way to access the configuration commands and the running object management commands.
- The AdminTask object provides a large number of commands that perform simple and complex administrative tasks
- The AdminTask object commands work like wizards
- Commands can be invoked in several ways
  - ▶ Interactively
  - ▶ In a script
  - ▶ In a profile
- AdminTask infocenter link
  - ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome\\_nd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welcome_nd.html)



# wsadmin AdminTask

## ■ Command Examples

### ▶ Help command partial list

```
wsadmin>print AdminTask.help('-commandGroups')
WASX8005I: Available admin command groups:
AdminReports - Admin configuration reports
AuthorizationGroupCommands - Authorization Group
AutoGen Commands - Commands for autogenerating LTPA password and server Id.
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 related operations.
ConfigLimits - No description available
CoreGroupBridgeManagement - A group of administrative commands that help in configuring core groups.
```

### ▶ List port management commands using help

```
wsadmin>print AdminTask.help('PortManagement')
WASX8007I: Detailed help for command group: PortManagement
Description: A group of admin commands that help in managing WebSphere ports
Commands:
listApplicationPorts - Displays a list of ports that is used to access the specified application, including
the node name, server name, named endpoint, and host and port values.
listServerPorts - Displays a list of ports that is used by a particular server, including the node name,
server name, named endpoint, and host and port values.
modifyServerPort - Modifies the host or port of the named endpoint that is used by the specified server.
```

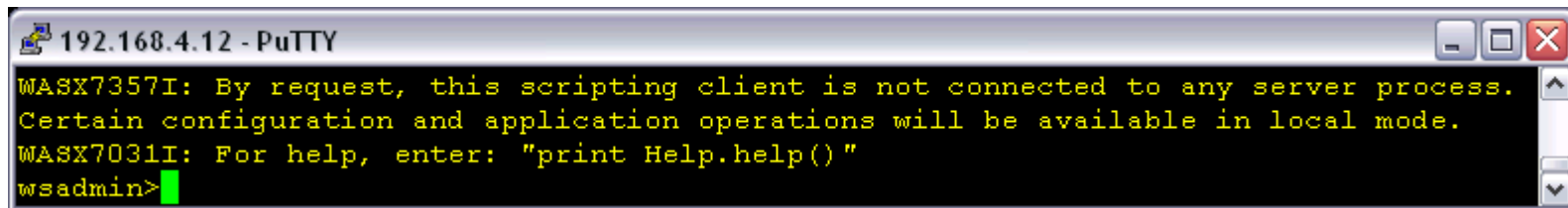




## wsadmin interface

- WebSphere Scripting Interface
- Located in the bin directory of any WebSphere server
  - `WebSphere/AppServer/profiles/{profile Name}/bin`
- The wsadmin tool can be used in 3 modes
  - ▶ Interactive – like working at a UNIX or DOS prompt
  - ▶ Scripted – using scripts and profiles
  - ▶ Local
- Local mode – scripting client with WebSphere not running

```
.../bin>wsadmin -lang jython -conntype NONE
```



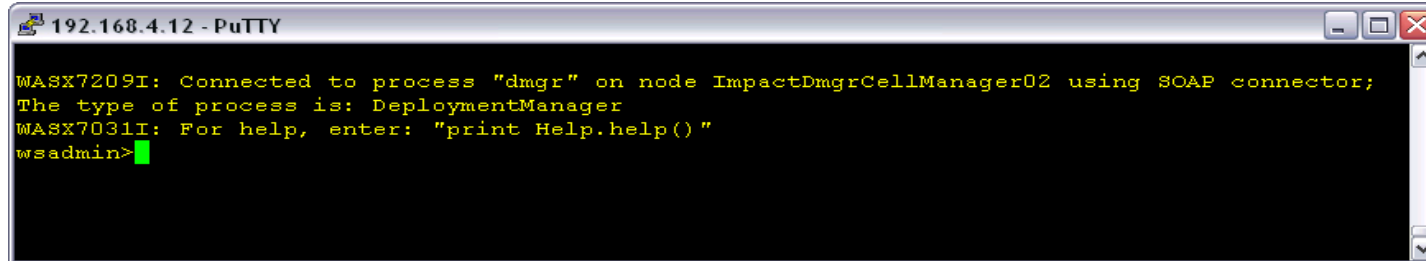
```
192.168.4.12 - PuTTY
WASX7357I: By request, this scripting client is not connected to any server process.
Certain configuration and application operations will be available in local mode.
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
```

- ▶ Not recommended if server is running

## wsadmin interface

- Interactive Mode – working from a prompt

```
/{profile Name}/bin>wsadmin -lang jython
```



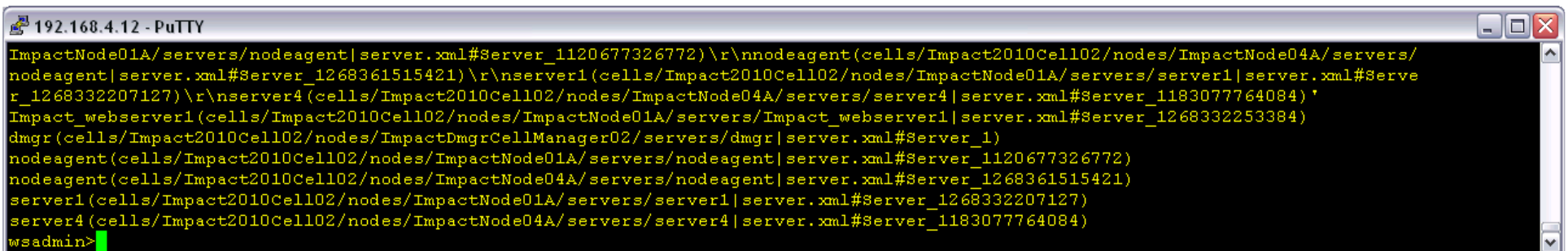
```
192.168.4.12 - PuTTY
WASX7209I: Connected to process "dmgr" on node ImpactDmgrCellManager02 using SOAP connector;
The type of process is: DeploymentManager
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
```

- Can be started from any bin directory, if other then the Deployment Manager set DMGR port number to connect

```
/{profile Name}/bin>wsadmin -lang jython -port 1234
```

- Jython commands can be entered just like a UNIX prompt

```
wsadmin>print AdminConfig.list('Server')
```



```
192.168.4.12 - PuTTY
ImpactNode01A/servers/nodeagent|server.xml#Server_1120677326772)\r\nnodeagent(cells/Impact2010Cell102/nodes/ImpactNode04A/servers/
nodeagent|server.xml#Server_1268361515421)\r\nserver1(cells/Impact2010Cell102/nodes/ImpactNode01A/servers/server1|server.xml#Serve
r_1268332207127)\r\nserver4(cells/Impact2010Cell102/nodes/ImpactNode04A/servers/server4|server.xml#Server_1183077764084)'
Impact_webserver1(cells/Impact2010Cell102/nodes/ImpactNode01A/servers/Impact_webserver1|server.xml#Server_1268332253384)
dmgr(cells/Impact2010Cell102/nodes/ImpactDmgrCellManager02/servers/dmgr|server.xml#Server_1)
nodeagent(cells/Impact2010Cell102/nodes/ImpactNode01A/servers/nodeagent|server.xml#Server_1120677326772)
nodeagent(cells/Impact2010Cell102/nodes/ImpactNode04A/servers/nodeagent|server.xml#Server_1268361515421)
server1(cells/Impact2010Cell102/nodes/ImpactNode01A/servers/server1|server.xml#Server_1268332207127)
server4(cells/Impact2010Cell102/nodes/ImpactNode04A/servers/server4|server.xml#Server_1183077764084)
wsadmin>
```

## Initiating wsadmin scripts

- Initiating a jython script with wsadmin

- ▶ Works in either platform

- unix

```
wsadmin.sh -lang jython -f \MyScriptsDirectory\myJython.py
```

- windows

```
wsadmin -lang jython -f /MyScriptsDirectory/myJython.py
```

- ▶ Start at the bin directory of the deployment manager

- ▶ -f : indicates the fully qualified jython script file

- ▶ -lang: indicates the language of the script

- ▶ Other commonly used options

- -conntype : SOAP, RMI, NONE
- -host : host name
- -port : port number of Deployment Manager process
- -profile : used when running a scripting profile/framework

- Can be saved in a shell or batch file

- To see full list of initiation options type

```
\bin>wsadmin -help
```



## Creating Jython Scripts

- There are several options to create scripts with WebSphere V7
  - ▶ Creating scripts from scratch
  - ▶ Command Assistance
- Create a script that will list all Web container custom properties for all servers in an environment
- When creating Jython scripts remember
  - ▶ Case sensitive
  - ▶ Indent sensitive
  - ▶ Uses Python key words/commands
  - ▶ Uses Python grammar
  - ▶ Jython is a scripting language that is compiled by Java
    - Java functions can be utilised within scripts



# Command Assistance

- New in WebSphere V7
- Provides assistance for scripting from the administrative console
  - ▶ The command assistant will show a line of Jython code for some administrative functions
  - ▶ Logs commands to a log file
  - ▶ Commands can be used at the wsadmin prompt or placed into a script.
- Enabled through Console preferences
  - ▶ System Administration > Console Preferences
  - ▶ Check boxes
    - Enable command assistance notifications
    - Log command assistance commands
  - ▶ Command Assistance log is located in in the log directory of the process in which the console is running in
    - Log for deployment manager is located in

[/WebSphere/AppServer/Profiles/Dmgr/logs/dmgr/commandAssistanceJythonCommands\\_{user}.log](#)



# Creating a jython script using Command Assistance

- List Web container custom properties
  - ▶ Go to

Server Types > WebSphere Application Servers > {server Name} > Web container settings > Web container > custom properties

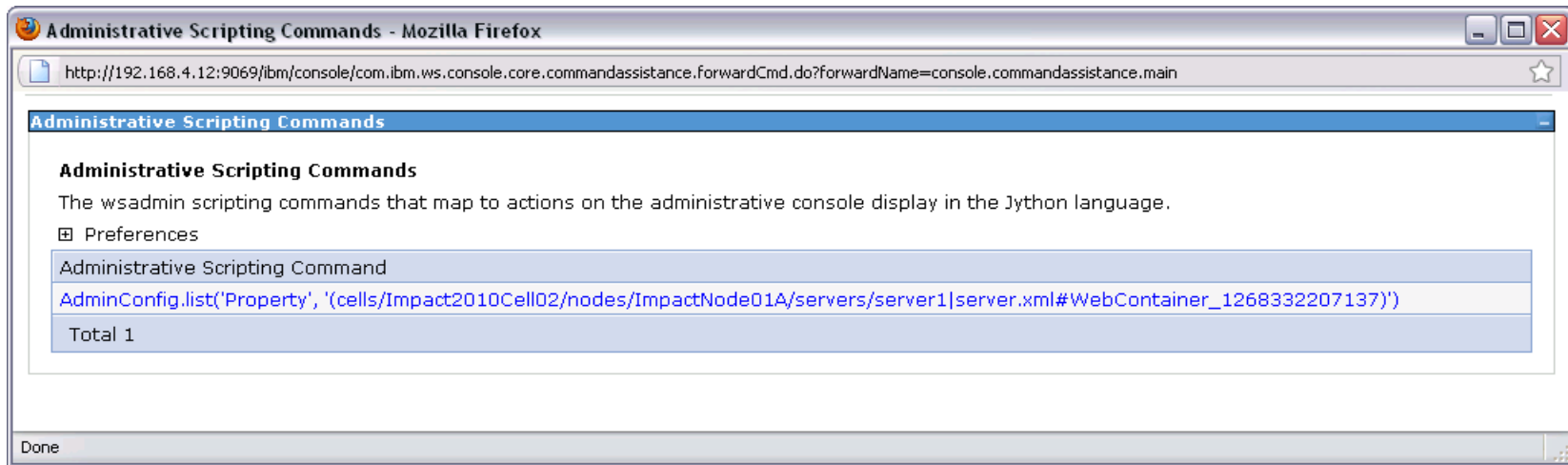
The screenshot shows the Integrated Solutions Console interface in Mozilla Firefox. The browser address bar shows the URL: `http://192.168.4.12:9069/ibm/console/login.do`. The page title is "Integrated Solutions Console - Welcome jsr". The left navigation pane shows the "Servers" section expanded to "WebSphere application server". The main content area displays the "Application servers > server1 > Web container > Custom properties" page. The page includes a "Preferences" section with "New" and "Delete" buttons, and a table of custom properties. The table has columns for "Select", "Name", "Value", and "Description". One property is listed: "myWebcontainer\_property" with value "my\_wc\_value" and description "new Web container property". The table shows "Total 1" rows. On the right side, there is a "Help" panel with sections for "Field help", "Page help", and "Command Assistance".

| Select                   | Name                                    | Value       | Description                |
|--------------------------|---|-------------|----------------------------|
| <input type="checkbox"/> | <a href="#">myWebcontainer_property</a> | my_wc_value | new Web container property |

- ▶ On the screen shot above you can see the custom property along with command assistance on the right hand side of the screen

# Creating a jython script using Command Assistance

- Selecting command assistance brings up the following dialog box



# Creating a jython script using Command Assistance

## ■ Command assistance log file

```
# [7/15/10 15:49:08:296 EDT] Application servers > server1 > Web container > Custom properties > New
AdminConfig.create('Property',
' (cells/Impact2010Cell102/nodes/ImpactNode01A/servers/server1|server.xml#WebContainer_1268332207137) ',
' [[validationExpression "" ] [name "myWebcontainer_property"] [description "new Web container property"] [value "my_wc_value"]
[required "false"]] ')

# [7/15/10 15:49:09:899 EDT] Application servers > server1 > Web container > Custom properties
AdminConfig.save()

# [7/15/10 15:49:10:950 EDT] Application servers > server1 > Web container > Custom properties
AdminConfig.list('Property', ' (cells/Impact2010Cell102/nodes/ImpactNode01A/servers/server1|
server.xml#WebContainer_1268332207137) ')
```

## ■ Creating jython script

- ▶ Create and place file in a scripts directory  
`/data/scripts/myJython.py`
- ▶ Copy the necessary commands into that file – output from command assistance
- ▶ Add any necessary commands to break down element attributes
- ▶ Add any formatting - print statements





# Creating a jython script using Command Assistance

- Script to list Web container custom properties
  - ▶ server.xml from server1, Web container custom property element

```
<components xmi:type="applicationserver.webcontainer:WebContainer" xmi:id="WebContainer_1268332207137"
  enableServletCaching="false" disablePooling="false">
  ...
  <properties xmi:id="Property_1279223348286" name="myWebcontainer_property" value="my_wc_value"
    description="new Web container property" required="false"/>
</components>
```

- ▶ Command from command assistance log

```
AdminConfig.list('Property',
  '(cells/Impact2010Cell102/nodes/ImpactNode01A/servers/server1|server.xml#WebContainer_1268332207137)')
```



# Creating a jython script using Command Assistance

- Jython script - `/data/scripts/myJython.py`

```
# [7/15/10 15:49:10:950 EDT] Application servers > server1 > Web container > Custom properties
# script name
print "Script myJython.py"

# retrieve custom property element
webContainerPropList = AdminConfig.list('Property',
    '(cells/Impact2010Cell102/nodes/ImpactNode01A/servers/server1|server.xml#WebContainer_1268332207137)')

# assign attributes to variables
property_name = AdminConfig.showAttribute(webContainerPropList, "name")
property_value = AdminConfig.showAttribute(webContainerPropList, "value")
property_descript = AdminConfig.showAttribute(webContainerPropList, "description")

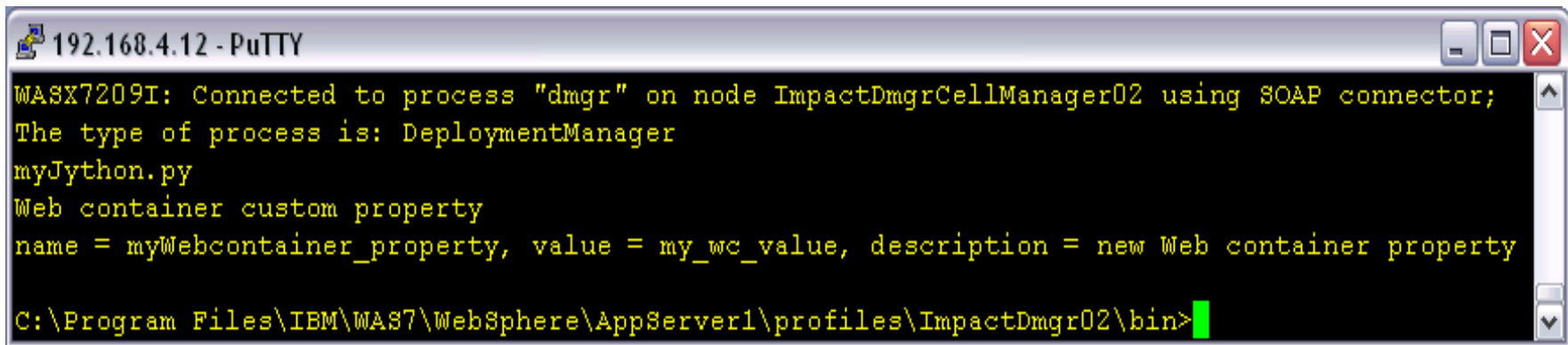
# Display Web container custom property
print "Web container custom property"
print "name = " + property_name + ", value = " + property_value + ", description = " + property_descript
```

## Creating a jython script using Command Assistance

- Execute Jython script

```
...\bin> wsadmin -lang jython -f /data/scripts/myJython.py
```

- Output from script



```
192.168.4.12 - PuTTY
WASX7209I: Connected to process "dmgr" on node ImpactDmgrCellManager02 using SOAP connector;
The type of process is: DeploymentManager
myJython.py
Web container custom property
name = myWebcontainer_property, value = my_wc_value, description = new Web container property
C:\Program Files\IBM\WAS7\WebSphere\AppServer1\profiles\ImpactDmgr02\bin>
```

- Things to note

- ▶ Script will only display one custom property
- ▶ Script will only work for server1 Web container
- ▶ Command assistance will only provide command to retrieve specific xml element

# Modifying jython.py – list multiple custom properties

- Changes to myJython.py
  - ▶ LineSeperator
  - ▶ For statement
    - *webContainerProp* variable
    - indent

```
# [7/15/10 15:49:10:950 EDT] Application servers > server1 > Web container > Custom properties
# script name
print "Script myJython.py"
#define a line seperator
lineSeperator = java.lang.System.getProperty('line.seperator'):
# retrieve custom property element
webContainerPropList = AdminConfig.list('Property',
' (cells/Impact2010Cell102/nodes/ImpactNode01A/servers/
server1|server.xml#WebContainer_1268332207137) ')
# scroll through list of custom properties
for webContainerProp in webContainerPropList.split(lineSeperator)
# assign attributes to variables
property_name = AdminConfig.showAttribute(webContainerPropList, "name")
property_value = AdminConfig.showAttribute(webContainerPropList, "value")
property_descript = AdminConfig.showAttribute(webContainerPropList, "description")
# Display Web container custom property
print "Web container custom property"
print " - name = " + property_name + ", value = " + property_value + ", description = " + property_descript
#end for loop
```

# Modifying jython.py – list multiple custom properties

- New list of properties

Integrated Solutions Console - Welcome jsr

| Select                   | Name                                      | Value         | Description                |
|--------------------------|---|---------------|----------------------------|
| <input type="checkbox"/> | <a href="#">myWebcontainer_property</a>   | my_wc_value   | new Web container property |
| <input type="checkbox"/> | <a href="#">myWebcontainer_property_2</a> | my_wc_value_2 | Web container property 2   |
| <input type="checkbox"/> | <a href="#">myWebcontainer_property_3</a> | my_wc_value_3 | Web container property 3   |
| Total 3                  |   |               |                            |

```

192.168.4.12 - PuTTY
WASX7209I: Connected to process "dmgr" on node ImpactDmgrCellManager02 using SOAP connector; The type of process is:
DeploymentManager
Script myJython.py
Web container custom property
- name = myWebcontainer_property, value = my_wc_value, description = new Web container property
Web container custom property
- name = myWebcontainer_property_2, value = my_wc_value_2, description = Web container property 2
Web container custom property
- name = myWebcontainer_property_3, value = my_wc_value_3, description = Web container property 3
C:\Program Files\IBM\WAS7\WebSphere\AppServer1\profiles\ImpactDmgr02\bin>
  
```

# Modifying jython.py – work with multiple servers

- Changes
  - ▶ Servers list - “Server” type
  - ▶ Web container list - “WebContainer” type
  - ▶ try/except statements
  - ▶ New variables: **ServerIDList, serverID, webContainerID**

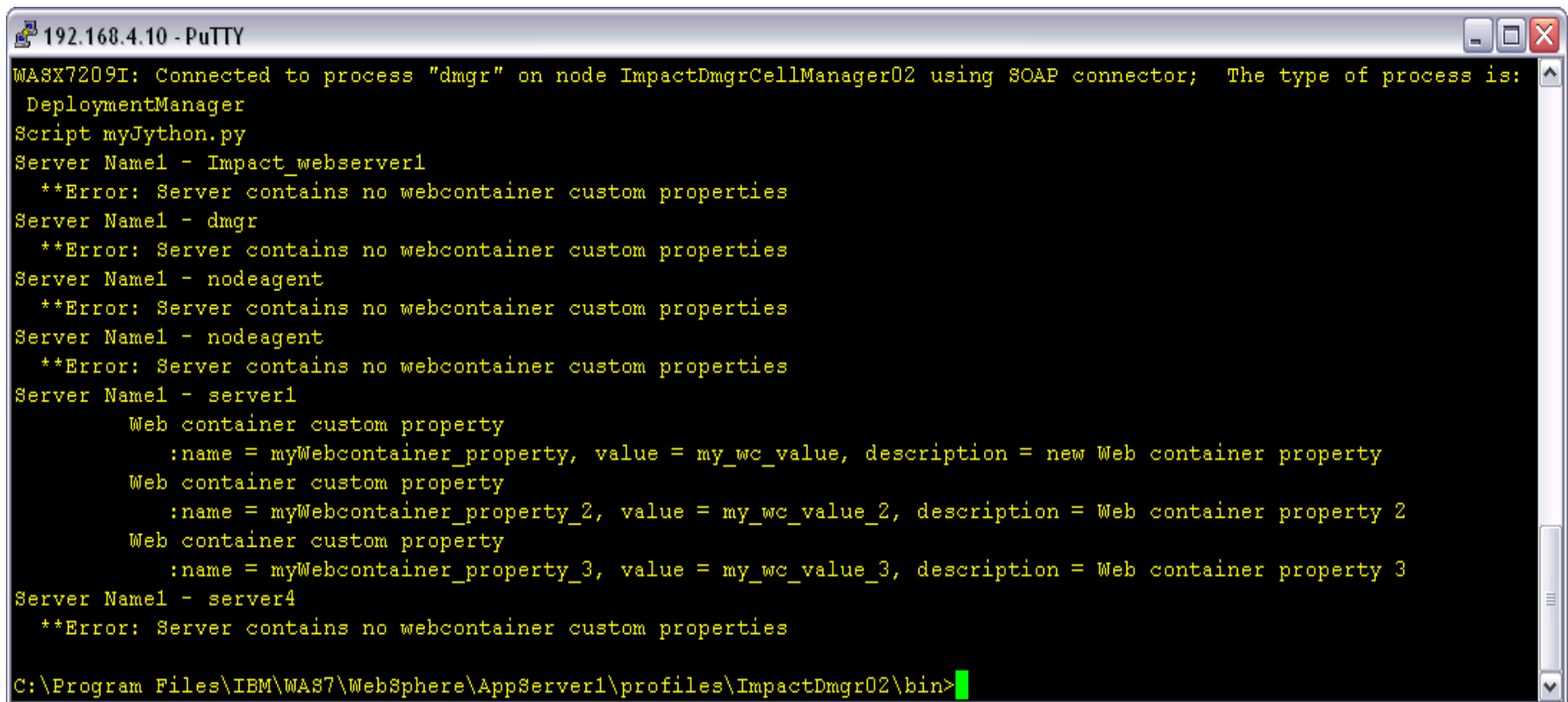
```
# [7/15/10 15:49:10:900 EDT] Application Servers > Server1 > Web Container > Custom properties
# script name
print "Script myJython.py"
#define a line seperator
lineSeperator = java.lang.System.getProperty('line.seperator');
ServerIDList = AdminConfig.list("Server")
for serverID in ServerIDList.split(lineSeperator):
    print "Server Name - " + AdminConfig.showAttribute(serverID, "name")
    try:
        webContainerID = AdminConfig.list("WebContainer", serverID)
        #retrieve custom property element list
        webContainerPropList = AdminConfig.list('Property', webContainerID)
        # scroll through list of custom properties
        for webContainerProp in webContainerPropList.split(lineSeperator)
            # assign attributes to variables
            property_name      = AdminConfig.showAttribute(webContainerPropList, "name")
            property_value     = AdminConfig.showAttribute(webContainerPropList, "value")
            property_descript  = AdminConfig.showAttribute(webContainerPropList, "description")
            # Display Web container custom property
            print "Web container custom property"
            print " : name = "+property_name+", value = "+property_value+", description = "+property_descript
        #end for loop
    #end try
    except:
        print " **Error: Server contains no WebContainer or WebContainer custom properties"
    #end except
#end for loop
```



# Modifying jython.py – work with multiple servers

- Output

- ▶ Not all servers contain a Web container or custom properties
- ▶ All processes are considered servers, note server names



```
192.168.4.10 - PuTTY
WASX7209I: Connected to process "dmgr" on node ImpactDmgrCellManager02 using SOAP connector; The type of process is:
DeploymentManager
Script myJython.py
Server Name1 - Impact_webserver1
  **Error: Server contains no webcontainer custom properties
Server Name1 - dmgr
  **Error: Server contains no webcontainer custom properties
Server Name1 - nodeagent
  **Error: Server contains no webcontainer custom properties
Server Name1 - nodeagent
  **Error: Server contains no webcontainer custom properties
Server Name1 - server1
  Web container custom property
    :name = myWebcontainer_property, value = my_wc_value, description = new Web container property
  Web container custom property
    :name = myWebcontainer_property_2, value = my_wc_value_2, description = Web container property 2
  Web container custom property
    :name = myWebcontainer_property_3, value = my_wc_value_3, description = Web container property 3
Server Name1 - server4
  **Error: Server contains no webcontainer custom properties

C:\Program Files\IBM\WAS7\WebSphere\AppServer1\profiles\ImpactDmgr02\bin>
```



# Appendix

- Jython Development Tools Presentation

- ▶ [http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was\\_v6/was/6.1/DevelopmentTools/WASv61\\_JythonTools/player.html](http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/DevelopmentTools/WASv61_JythonTools/player.html)

- Info center – getting started with scripting

- ▶ [http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was\\_v6/was/6.1/DevelopmentTools/WASv61\\_JythonTools/player.html](http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/DevelopmentTools/WASv61_JythonTools/player.html)

- RedPapers: WebSphere Application Server V7: Administration with Scripting

- ▶ <http://www.redbooks.ibm.com/abstracts/redp4576.html?Open>

- WebSphere Application Server Administration using Jython: by Bob Gibson

- ▶ <http://www.ibmpressbooks.com/bookstore/product.asp?isbn=9780137009527>





Questions ?

