

IBM WebSphere Application Server V7 – Lab exercise

WebSphere Application Server Network Deployment V7

Properties file configuration

What this exercise is about	1
Lab requirements	1
What you should be able to do	2
Introduction	2
Exercise instructions	3
Part 1: Extract a configuration properties file from a server.....	4
Part 2: Modify and apply a properties file to a server	5
Part 3: Use filters to extract properties.....	14
What you did in this exercise	18

What this exercise is about

Configuration information for WebSphere Application Server is composed of many files, in XML and other formats. These configuration files are spread across many directories in the application server's configuration tree. These files are made up mostly of complex configuration objects which are difficult for people to read, understand, and edit directly. While there are tools available to manage the server configuration, it would be useful to have a simple, readable file that reflects the configuration of your environment. WebSphere Application Server V7 provides a new set of utilities for working with a server's configuration using properties files. You can create a properties file of human readable key-value pairs based on your environment, make modifications to that file, and then apply the updated properties to a server. The objective of this lab is to provide you with an understanding of this new technique for administering your environment.

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.

- A system that meets that requirements for running WebSphere Application Server Version 7.0, with approximately 500 MB of disk space for creating profiles
- The most current version of WebSphere Application Server V7
- Two application server profiles with administrative security disabled and with the administrative console and the default application deployed – Part 1 of the lab walks you through creating these profiles

What you should be able to do

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

- Use properties file configuration tools to extract the configuration of your existing environment
 - Apply configuration information from a properties file to a server
-

Introduction

In this lab, you will work through some samples based on the properties file configuration utilities that were introduced in WebSphere Application Server V7.

Part 1: Extract a configuration properties file from a server

In the first portion of the lab, you will create the profiles that are used throughout the exercises. You will also learn how to use the `extractConfigProperties` command to create a configuration properties file based on your cell environment.

Part 2: Modify and apply a properties file to a server

Once you have the properties file for your cell, you can modify the file and apply those changes to your environment. In this portion of the exercise, you will update the properties file to contain new ports and virtual host information, apply those changes to your environment, and then verify that the changes were applied successfully.

Part 3: Use filters to extract properties

In the previous sections, you used a large properties file that contained all of the configuration information for an entire cell. It is also possible to use filters when extracting a properties file. In this section of the exercise, you will extract a properties file that contains information about the Java Virtual Machine configuration for your server, modify the properties, and apply the changes to the server.

Exercise instructions

Some instructions in this lab may be 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	Sample Windows location	Sample AIX® or UNIX® location
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/usr/WebSphere/AppServer /opt/WebSphere/AppServer

Note for Windows users: When directory locations are passed as parameters to a Java program such as wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\LabFiles70\ with C:/LabFiles70/

Special instructions:

- This lab is heavily command-based. In the lab instructions, these commands will often span multiple lines. In all cases, when you are typing in the commands, you should type them as one continuous entry, with no new line characters.
- In most cases, commands are given with no file extension (that is, .bat or .sh are omitted). Use the appropriate command structure for your operating system.

Part 1: Extract a configuration properties file from a server

- ___ 1. The exercises in this lab require that you have an application server profile created with the following features: administrative security is disabled, and the administrative console and default application are deployed on the server. In this step, you will create this profile.

- ___ a. Open a command prompt on your system and navigate to the WebSphere Application Server V7 bin directory.

```
cd <WAS_HOME>\bin
```

- ___ b. Use the `manageprofiles` command to create the first profile, `Props1`. Use the appropriate command extension (for example, `.bat` or `.sh`) and directory separator for your platform. The administrative console and the default application are required for this lab, but will be deployed by default to this profile because it is an application server profile. If you are running on the Windows platform and your fully-qualified directory name includes the space character, you need to enclose the path in quotation marks, as shown in the example below.

```
manageprofiles -create -profileName Props1 -templatePath
"<WAS_HOME>\profileTemplates\default" -enableAdminSecurity false
```

- ___ c. The command above may take several minutes to complete. When the profile creation has completed successfully, a message similar to this one will be displayed in your console:

```
INSTCONFSUCCESS: Success: Profile Props1 now exists. Please consult
C:\ProgramFiles\IBM\WebSphere\AppServer\V7\profiles\Props1\logs\About
ThisProfile.txt for more information about this profile.
```

- ___ 2. Go into `wsadmin` and extract the properties for the cell. The `wsadmin` commands in this section are `jython`-based.

- ___ a. Open a command prompt and navigate to the bin directory of the `Props1` profile.

```
cd <WAS_HOME>/profiles/Props1/bin
```

- ___ b. Start `wsadmin`. Use the appropriate command extension (for example, `.sh` or `.bat`) for your platform.

```
wsadmin -lang jython -conntype none
```

- ___ c. Wait for `wsadmin` to start. You may see several messages scroll through your display about `.jar` file processing. When `wsadmin` has started, you will see a prompt like the one below:

```
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
```

- ___ d. Properties files can be extracted for a variety of configuration attributes at different levels – a cell, a node, a server, one container in that server, and others. If you do not specify a particular type of configuration data to gather, the command will extract configuration information for the entire cell. Run the command below to create a properties file, `props1.props`, for profile `Prop1`'s cell

```
AdminTask.extractConfigProperties('-propertiesFileName props1.props')
```

- ___ e. Wait for the extraction to complete. When it's done, you will see two single-quote characters in your command prompt. Since you did not specify an absolute path for the file, the configuration file was generated in the current working directory, which is the profile's bin directory.

Part 2: Modify and apply a properties file to a server

1. Before modifying a properties file, it's a good practice to create a back up copy of that file. In this exercise, you will not be directly using the backup file, but it is still a best practice to create one.
 - a. Locate the properties – props1.props – that you created in the previous section of the exercise. You can find it here: <WAS_HOME>/profiles/Props1/bin/props1.props
 - b. Open a command prompt on your system and navigate to the location of the properties file.


```
cd <WAS_HOME>/profiles/Props1/bin
```
 - c. Create a back up copy of the properties file. Use the appropriate command below, based on your operating system.

```
copy props1.props props1_backup.props
```



```
cp props1.props props1_backup.props
```



2. Now you can modify the properties file to include the updated properties you want to apply to your server. In this exercise, you will be setting up new ports and virtual hosts for your server. In this step, you will update the port values.
 - a. Open the original properties file – props1.props – with any text editor. For example, on Windows, you can use Notepad, or on Linux, you could use vi.
 - b. In the editor, find the section that contains the port information for your server by searching for “Ports Section”. You will see something similar to the following example; the port numbers will vary depending on your system. The entries highlighted in yellow are the ones you will change.

```
#
# SubSection 1.0.1 # Ports Section
#
ResourceType=EndPoint
ImplementingResourceType=Server
ResourceId=Cell={!{cellName}:Node={!{nodeName}:Server={!{serverName}
#

#
#Properties
#
SOAP_CONNECTOR_ADDRESS=8899:!!{hostName} # integer
SIP_DEFAULTHOST_SECURE=5082:!!{hostName1} # integer
SIP_DEFAULTHOST=5083:!!{hostName1} # integer
SIB_ENDPOINT_ADDRESS=7302:!!{hostName1} # integer
WC_defaulthost_secure=9463:!!{hostName1} # integer
DCS_UNICAST_ADDRESS=9370:!!{hostName1} # integer
SIB_MQ_ENDPOINT_SECURE_ADDRESS=5589:!!{hostName1} # integer
WC_adminhost_secure=9077:!!{hostName1} # integer
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=9464:!!{hostName} # integer
ORB_LISTENER_ADDRESS=9117:!!{hostName} # integer
BOOTSTRAP_ADDRESS=2823:!!{hostName} # integer
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=9465:!!{hostName} # integer
IPC_CONNECTOR_ADDRESS=9648:!!{hostName2} # integer
SIB_ENDPOINT_SECURE_ADDRESS=7303:!!{hostName1} # integer
```

```
WC_defaulthost=9091:!!{hostName1} # integer
SIB_MQ_ENDPOINT_ADDRESS=5569:!!{hostName1} # integer
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=9466:!!{hostName} # integer
WC_adminhost=9078:!!{hostName1} # integer
```

__ c. From the section of ports above, you will be modifying the entries for WC_defaulthost_secure, WC_adminhost_secure, WC_defaulthost, WC_adminhost. These are highlighted in yellow in the entry above. You will be updating the port values for these four values in the next four steps.

__ d. Locate the entry for WC_defaulthost_secure and change the port number to 21443. The new line should look like this:

```
WC_defaulthost_secure=21443:!!{hostName1} # integer
```

__ e. Locate the entry for WC_adminhost_secure and change the port number to 21043. The new line should look like this:

```
WC_adminhost_secure=21043:!!{hostName1} # integer
```

__ f. Locate the entry for WC_defaulthost and change the port number to 21080. The new line should look like this:

```
WC_defaulthost=21080:!!{hostName1} # integer
```

__ g. Locate the entry for WC_adminhost and change the port number to 21060. The new line should look like this:

```
WC_adminhost=21060:!!{hostName1} # integer
```

__ h. Now your port properties will look similar to this example:

```
#
#Properties
#
SOAP_CONNECTOR_ADDRESS=8899:!!{hostName} # integer
SIP_DEFAULTHOST_SECURE=5082:!!{hostName1} # integer
SIP_DEFAULTHOST=5083:!!{hostName1} # integer
SIB_ENDPOINT_ADDRESS=7302:!!{hostName1} # integer
WC_defaulthost_secure=21443:!!{hostName1} # integer
DCS_UNICAST_ADDRESS=9370:!!{hostName1} # integer
SIB_MQ_ENDPOINT_SECURE_ADDRESS=5589:!!{hostName1} # integer
WC_adminhost_secure=21043:!!{hostName1} # integer
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=9464:!!{hostName} # integer
ORB_LISTENER_ADDRESS=9117:!!{hostName} # integer
BOOTSTRAP_ADDRESS=2823:!!{hostName} # integer
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=9465:!!{hostName} # integer
IPC_CONNECTOR_ADDRESS=9648:!!{hostName2} # integer
SIB_ENDPOINT_SECURE_ADDRESS=7303:!!{hostName1} # integer
WC_defaulthost=21080:!!{hostName1} # integer
SIB_MQ_ENDPOINT_ADDRESS=5569:!!{hostName1} # integer
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=9466:!!{hostName} # integer
WC_adminhost=21060:!!{hostName1} # integer
```

__ i. Save the changes to the properties file, but keep it open.

___ 3. Update the virtual host properties to reflect the port value changes you made above.

__ a. In the props1.props file, search for "Virtual Hosts". There will be two sections – the first is for the default host. In this portion of the file, there will be the following sections:


1) Virtual hosts description

2) MimeTypes section

3) HostAlias section

- ___ b. The properties in the HostAlias section describe the port configuration associated with the default host. It will look similar to this:

```
#
#Properties
#
5089=*
9471=*
5088=*
9095=*
80=*
443=*
9091=*
```

 Insert lines here

- ___ c. In the previous section, you changed the port values for the default host to 21080 for normal requests and 21443 for secure requests. Configure the default host to monitor those ports by adding the following two entries to the properties list:

```
21080=*
21443=*
```

- ___ d. In the props1.props file, search for the next instance of “Virtual Hosts.” This is the virtual host configuration information for the administrative host. It includes the same sorts of information as the default host definition – a description and information about the mime types and host aliases. Scroll down through the properties file to find the “HostAlias section”. It will look similar to the following:

```
#
# SubSection 1.2 # HostAlias section
#
ResourceType=VirtualHost
ImplementingResourceType=VirtualHost
ResourceId=cells/aimcp036Node12Cell|virtualhosts.xml#VirtualHost_2
AttributeInfo=aliases(port,hostname)
#

#
#Properties
#
9078=*
9077=*
```

 Insert lines here

- ___ e. In the previous section, you changed the port values for the administrative host to 21060 for normal requests and 21043 for secure requests. Configure the administrative host to monitor those ports by adding these two entries to the properties list:

```
21060=*
21043=*
```

- ___ f. You have completed modifying the ports and virtual hosts in your properties file. Save your changes and close the file.

___ 4. In this step, you will apply the updated properties file to the Props1 profile.

- ___ a. Open a command prompt and navigate to the bin directory of the Props1 profile.

```
cd <WAS_HOME>/profiles/Props1/bin
```

- ___ b. Start wsadmin. Use the appropriate command extension (for example, .sh or .bat) for your platform.

```
wsadmin -lang jython -conntype none
```

- ___ c. Wait for wsadmin to start. You may see several messages scroll through your display about JAR file processing. When wsadmin has started, you will see a prompt like the one below:

```
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
```

- ___ d. Before applying a properties file to a server, it is a good practice to validate the file. You can do this using the AdminTask.validateConfigProperties command. Use the following command to validate your file, you should enter the command as a single line:

```
AdminTask.validateConfigProperties('-propertiesFileName
props1.props')
```

- ___ e. The validation command may take a few moments to run. When it has finished, you will see the message 'true' in your command prompt.

- ___ f. When the validation is complete, apply the new values in the properties file to your environment using the following command.

```
AdminTask.applyConfigProperties('-propertiesFileName props1.props')
```

- ___ g. When the command has completed successfully, you will see two single-quote characters in the console window, and it will return to the wsadmin prompt.

- ___ h. Now that you have updated the configuration, you need to save the configuration. Use the following command to save your changes:

```
AdminConfig.save()
```

- ___ i. Close the wsadmin prompt using the exit command.

```
exit
```

- ___ 5. Now that you have updated your server, you can start it and verify that the changes you made were applied as expected. Since you changed the administrative and default HTTP port values and the associated virtual hosts, you can verify the configuration by accessing the administrative console and the default application.

- ___ a. Start the server by opening a command prompt to the profile Props1's bin directory.

```
cd <WAS_HOME>/profiles/Props1/bin
```

- ___ b. Use the startServer command to start the server, using the appropriate command extension for your operating system.

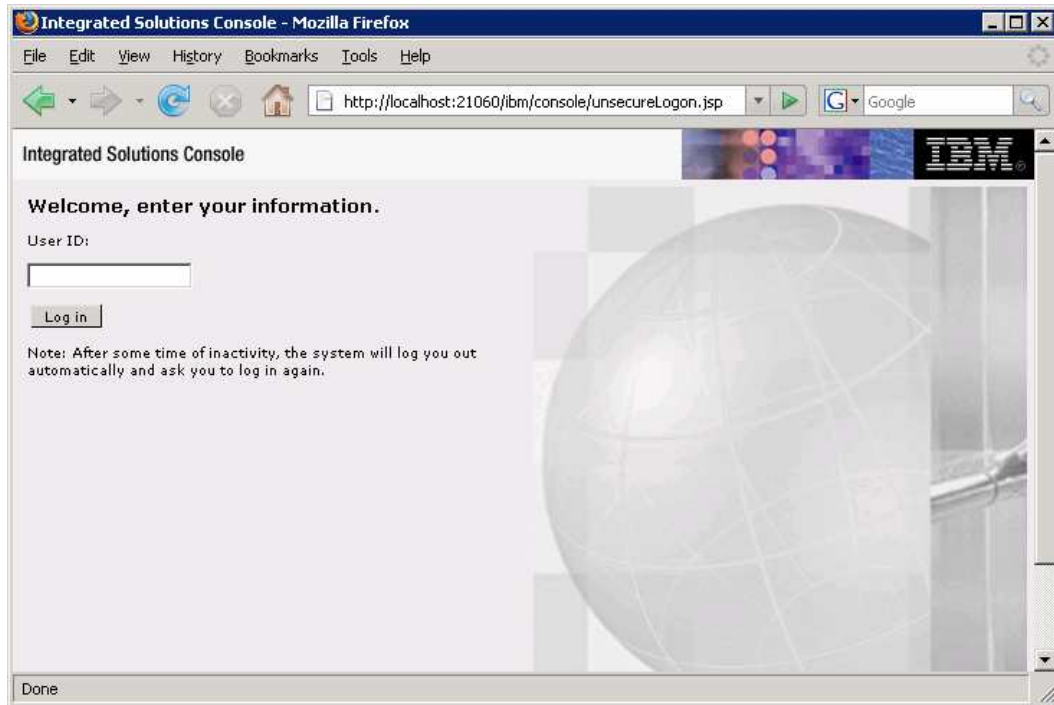
```
startServer server1
```

- ___ c. Wait for the server to start. Once it has started, you will see a message like the one below in your command prompt:

```
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 3104
```

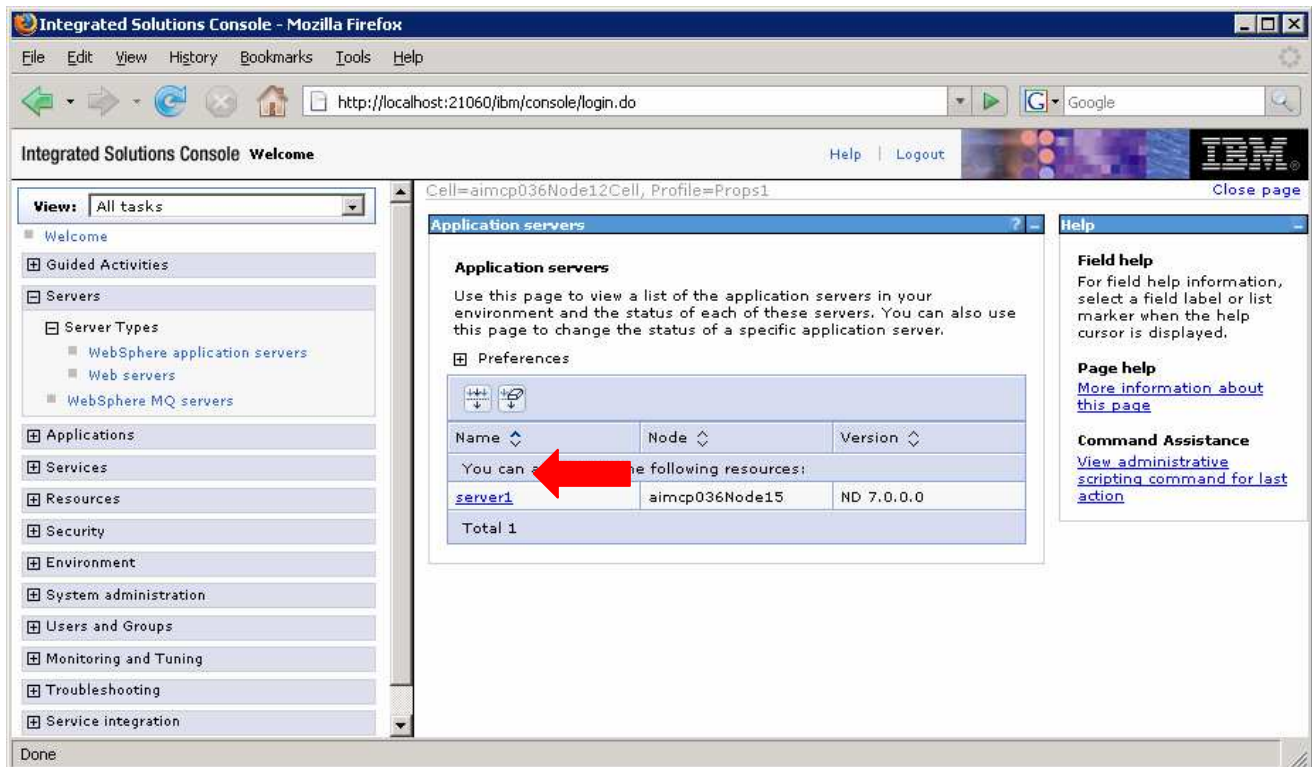
- ___ d. Open the administrative console by starting a browser on your test system and pointing it to the following URL, recall that you changed the default administrative port to 21060 in the previous sections of the exercise – <http://localhost:21060/ibm/console>

- ___ e. Since this server does not have administrative security enabled, you do not need to provide any authentication credentials. Click the **Log in** button to enter the console.

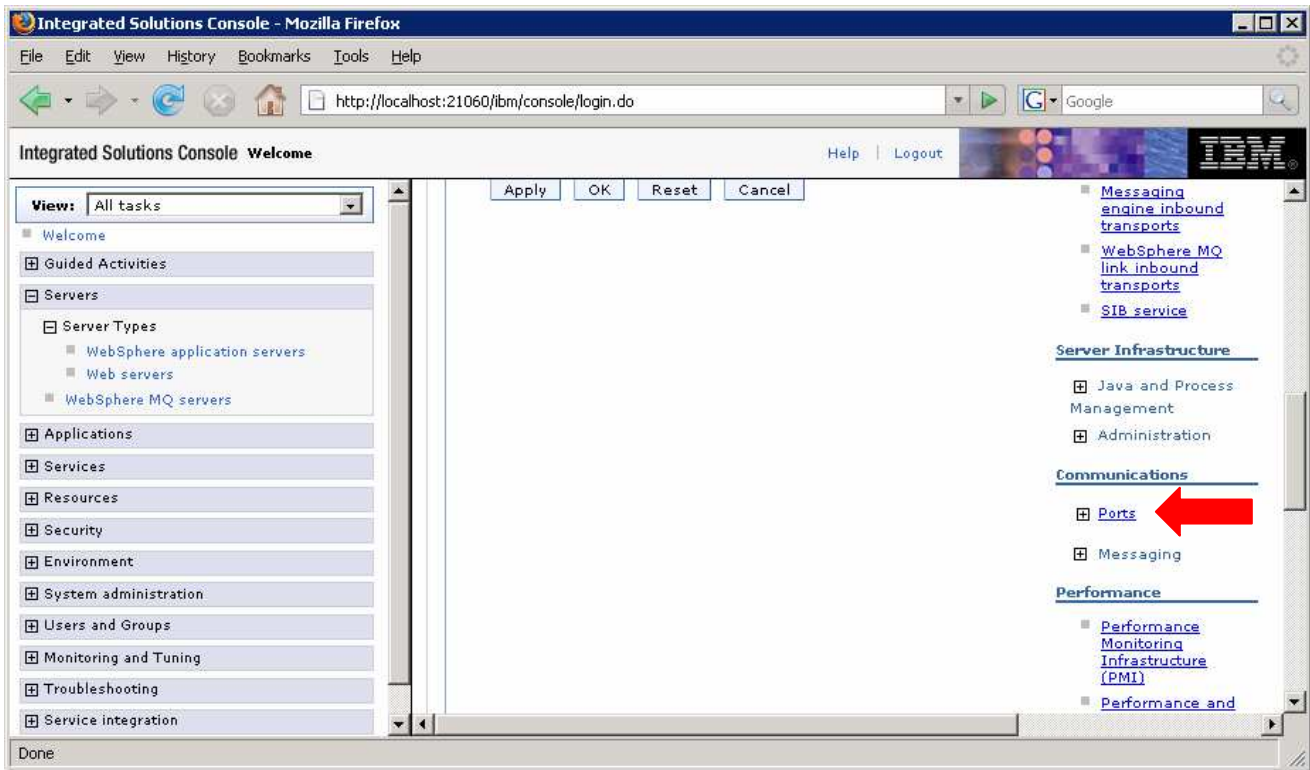


___ f. Check the port values for your server to verify that they match the values you configured above. In the left navigation menu, select **Servers > Server Types > WebSphere application servers**.

___ g. On the Application Servers screen, click the hyperlinked text **server1**.



- ___ h. On the next screen, there will be an expandable **Ports** menu on the right side under **Communications**. You may need to scroll down to see this section of the page. Click the hyperlinked text **Ports**.



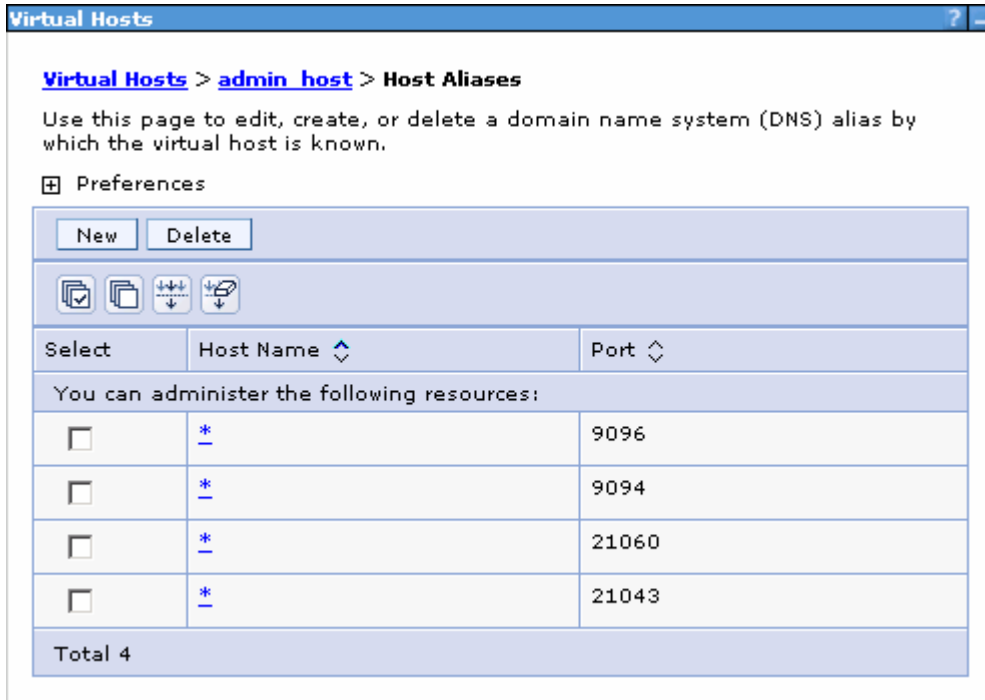
- ___ i. On the ports page, you can verify that the port configuration matches what you specified in the properties file. These are the values that you set in the file:

Description	Port
WC_adminhost	21060
WC_adminhost_secure	21043
WC_defaulthost	21080
WC_defaulthost_secure	21443

___ j. This screen is an example of what you will see on the Ports page. In this example, the last four rows contain the relevant port information.

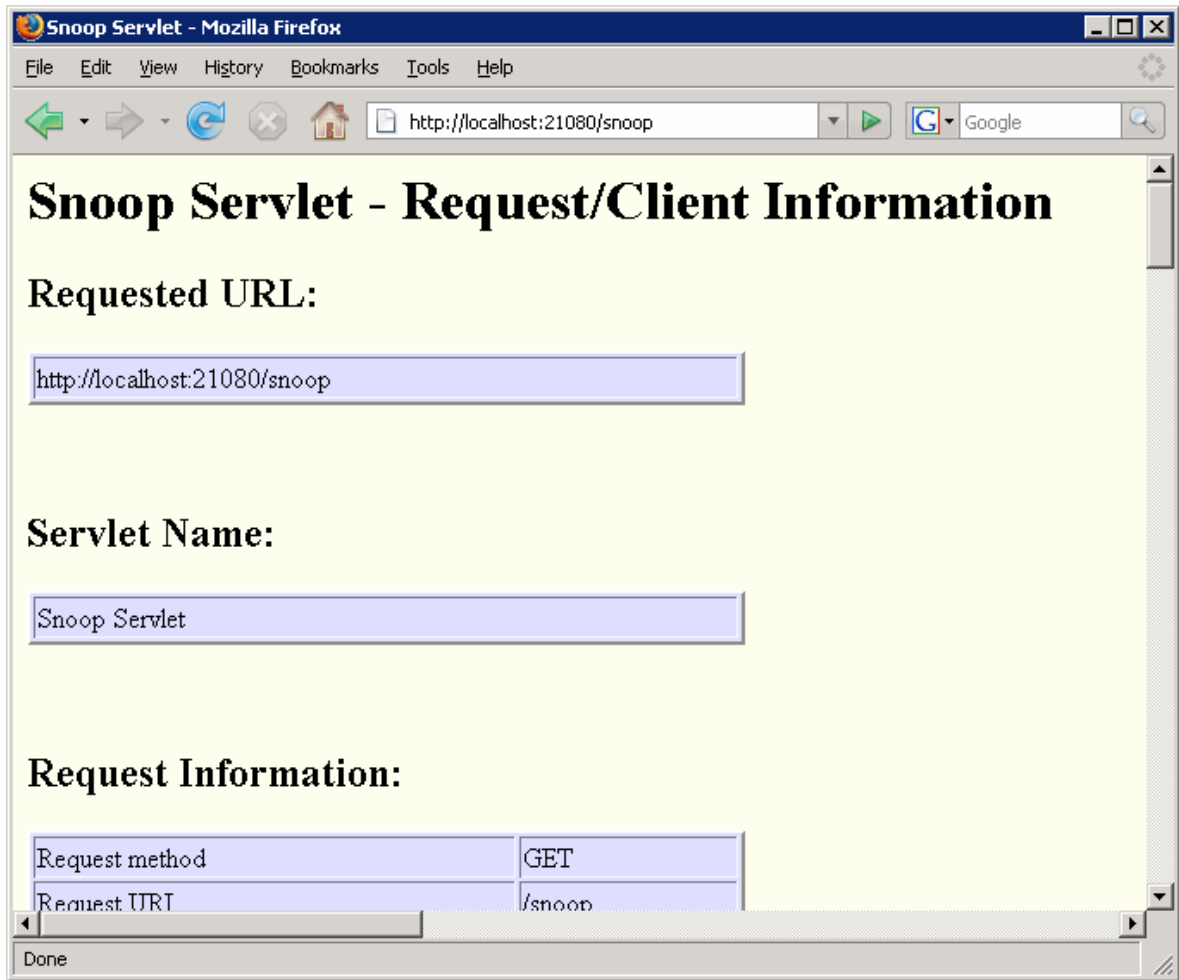
Select	Port Name	Host	Port	Transport Details
You can administer the following resources:				
<input type="checkbox"/>	BOOTSTRAP ADDRESS	aimcp036.austin.ibm.com	2826	No associated transports
<input type="checkbox"/>	CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	aimcp036.austin.ibm.com	9475	No associated transports
<input type="checkbox"/>	CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	aimcp036.austin.ibm.com	9476	No associated transports
<input type="checkbox"/>	DCS_UNICAST_ADDRESS	*	9373	View associated transports
<input type="checkbox"/>	IPC_CONNECTOR_ADDRESS	localhost	9651	No associated transports
<input type="checkbox"/>	ORB_LISTENER_ADDRESS	aimcp036.austin.ibm.com	9120	No associated transports
<input type="checkbox"/>	SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	aimcp036.austin.ibm.com	9477	No associated transports
<input type="checkbox"/>	SIB_ENDPOINT_ADDRESS	*	7308	View associated transports
<input type="checkbox"/>	SIB_ENDPOINT_SECURE_ADDRESS	*	7309	View associated transports
<input type="checkbox"/>	SIB_MQ_ENDPOINT_ADDRESS	*	5572	View associated transports
<input type="checkbox"/>	SIB_MQ_ENDPOINT_SECURE_ADDRESS	*	5592	View associated transports
<input type="checkbox"/>	SIP_DEFAULTHOST	*	5089	View associated transports
<input type="checkbox"/>	SIP_DEFAULTHOST_SECURE	*	5088	View associated transports
<input type="checkbox"/>	SOAP_CONNECTOR_ADDRESS	aimcp036.austin.ibm.com	8902	No associated transports
<input type="checkbox"/>	WC_adminhost	*	21060	View associated transports
<input type="checkbox"/>	WC_adminhost_secure	*	21043	View associated transports
<input type="checkbox"/>	WC_defaulthost	*	21080	View associated transports
<input type="checkbox"/>	WC_defaulthost_secure	*	21443	View associated transports
Total 18				

- ___ k. From the administrative console, you can also access the virtual host configuration to verify that the virtual hosts were updated correctly, based on the properties file. From the left navigation menu, expand **Environment** and select **Virtual hosts**. On the virtual host page, select **admin_host**. On the admin_host page, under **Additional Properties** on the right of the page, select **Host Aliases**. Entries for 21060 and 21043 should be in the table for the configuration of the admin_host.



- ___ l. Repeat the previous step, accessing the virtual host entry for default_host. That table should contain entries for 21080 and 21443.

- m. Finally, you can access the default application, snoop, using the default host that you have configured. Open a browser and point it to <http://localhost:21080/snoop>



Part 3: Use filters to extract properties

In the previous sections, you used a large properties file that contained all of the configuration information for an entire cell. It is also possible to use filters when extracting a properties file. In this section of the exercise, you will extract a properties file that contains information about the Java Virtual Machine configuration for your server, modify the properties, and apply the changes to the server.

___ 1. Extract the JVM properties of your server.

___ a. Open a command prompt and go to the Props1 profile's bin directory:

```
cd <WAS_HOME>/profiles/Props1/bin
```

___ b. Run this command to stop the server, using the appropriate extension for your operating system:

```
stopServer server1
```

___ c. After the server has stopped, use the same command prompt to start wsadmin:

```
wsadmin -lang jython -conntype none
```

___ d. There are a variety of filter types available for limiting the set of properties that you extract into a properties file using the `extractConfigProperties` command. Read through the categories in the following table to understand the different types of filters that are available:

AdminService	NodeGroup
Application	ObjectPoolProvider
ApplicationServer	ObjectRequestBroker
AuthorizationGroup	PMEServerExtension
AuthorizationTableExt	PMIModule
Cell	PMIService
CoreGroup	PortletContainer
CoreGroupBridgeService	SIPContainer
DynamicCache	SchedulerProvider
EJBContainer	Security
EventInfrastructureProvider	Server
EventInfrastructureService	ServerCluster
HAManagerService	TPVService
J2CResourceAdapter	TimerManagerProvider
JDBCProvider	TransactionService
JMSProvider	URLProvider
JavaVirtualMachine	VariableMap
Library	VirtualHost
MailProvider	WebContainer
NameServer	WebserverPluginSettings
Node	WorkManagerProvider

___ e. Use the `JavaVirtualMachine` filter to create a properties file that contains JVM-related configuration information for your server. The following command should be typed as a single line, with no line breaks.

```
AdminTask.extractConfigProperties('-propertiesFileName jvm.props
-configData Server=server1 -filterMechanism SELECTED_SUBTYPES
-selectedSubTypes [JavaVirtualMachine]')
```

- ___ f. When the command above has finished, open the `jvm.props` file with a text editor; the file will be located in the `bin` directory where you have been working. Notice that this properties file is much smaller than the file that you generated in the first exercise. In this case, the file only contains information related to Java Virtual Machine properties. Take a moment to browse through the file and look at the properties. Below is an example of what the full contents of the file will look like.

```
#
# Configuration properties file for
cells/aimcp036Node09Cell/nodes/aimcp036Node12/servers/server1|server.xml#
# Extracted on Mon May 19 13:18:33 CDT 2008
#

#
# Section 1.0 ##
Cell={!{cellName}:Node={!{nodeName}:Server={!{serverName}:JavaProcessDef=ID#JavaProc
essDef_1183122130078:JavaVirtualMachine=ID#JavaVirtualMachine_1183122130078
#

#
# SubSection 1.0.2.0.3 # JVM Section
#
ResourceType=JavaVirtualMachine
ImplementingResourceType=Server
ResourceId=Cell={!{cellName}:Node={!{nodeName}:Server={!{serverName}:JavaProcessDef=
ID#JavaProcessDef_1183122130078:JavaVirtualMachine=ID#JavaVirtualMachine_11831221
30078
AttributeInfo=jvmEntries
#

#
#Properties
#
internalClassAccessMode=ALLOW #ENUM(ALLOW|RESTRICT)
JavaHome="C:\Program Files\IBM\WebSphere\AppServer\V7\20080513/java" #readonly
debugArgs="-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7811"
classpath={}
initialHeapSize=0 #integer
runHProf=false #boolean
genericJvmArguments=
hprofArguments=
osName=null
bootClasspath={}
verboseModeJNI=false #boolean
maximumHeapSize=0 #integer
disableJIT=false #boolean
verboseModeGarbageCollection=false #boolean
executableJarFileName=null
verboseModeClass=false #boolean
debugMode=false #boolean
#
# SubSection 1.0.2.0.3.1 # System properties
#
ResourceType=JavaVirtualMachine
ImplementingResourceType=Server
ResourceId=Cell={!{cellName}:Node={!{nodeName}:Server={!{serverName}:JavaProcessDef=
ID#JavaProcessDef_1183122130078:JavaVirtualMachine=ID#JavaVirtualMachine_11831221
30078
AttributeInfo=systemProperties(name,value)
#

#
#Properties
```

```
#
com.ibm.security.krb5.Krb5Debug=off
com.ibm.security.jgss.debug=off

#
#
EnvironmentVariablesSection
#
#
#Environment Variables
#Mon May 19 13:18:38 CDT 2008
hostName2=localhost
hostName1=*
cellName=aimcp036Node09Cell
nodeName=aimcp036Node12
hostName=aimcp036.austin.ibm.com
serverName=server1
```

___ 2. Modify the properties file and apply the changes to the server.

___ a. Find the line in the jvm.props file for the verboseModeGarbageCollection property, which is highlighted in yellow in the example above. Change this property to 'true' to indicate that you want to enable verbose GC for the server.

```
verboseModeGarbageCollection=true #boolean
```

___ b. Before applying your changes, validate the updated file using the validateConfigProperties command.

```
AdminTask.validateConfigProperties('-propertiesFileName jvm.props')
```

___ c. Wait for the validation to complete and verify that the command returned 'true.'

___ d. Apply the changes to the server.

```
AdminTask.applyConfigProperties('-propertiesFileName jvm.props')
```

___ e. Save your configuration changes.

```
AdminConfig.save()
```

___ f. Exit wsadmin.

```
exit
```

___ 3. Start the server and verify the configuration changes.

___ a. From the <WAS_HOME>/profiles/Props1/bin directory, run the following command to start the server, using the appropriate command extension for your environment

```
startServer server1
```

___ b. Wait for the server to start. When the server has started, you will see a message similar to the following in your command prompt:

___ c. Verbose GC logging is sent to the native_stderr.log file, located here:

<WAS_HOME>/profiles/Props1/logs/server1/native_stderr.log. Open the file with a text editor and check that verbose GC messages are being recorded. They will be logged as XML-formatted entries in the file, similar to the following:

```
<af type="tenured" id="1" timestamp="May 22 14:03:00 2008" intervalms="0.000">
  <minimum requested_bytes="23848" />
  <time exclusiveaccessms="0.009" meanexclusiveaccessms="0.009" threads="0"
  lastthreadtid="0x143B8300" />
```



```

<refs soft="38" weak="9366" phantom="1" dynamicSoftReferenceThreshold="32"
maxSoftReferenceThreshold="32" />
<tenured freebytes="2621440" totalbytes="52428800" percent="5" >
  <soa freebytes="0" totalbytes="49807360" percent="0" />
  <loa freebytes="2621440" totalbytes="2621440" percent="100" />
</tenured>
<gc type="global" id="1" totalid="1" intervalms="0.000">
  <classunloading classloaders="0" classes="0" timevmquiescems="0.000"
timetakenms="0.239" />
  <finalization objectsqueued="90" />
  <timesms mark="12.220" sweep="0.934" compact="0.000" total="13.464" />
  <tenured freebytes="45886376" totalbytes="52428800" percent="87" >
    <soa freebytes="43264936" totalbytes="49807360" percent="86" />
    <loa freebytes="2621440" totalbytes="2621440" percent="100" />
  </tenured>
</gc>
<tenured freebytes="45862528" totalbytes="52428800" percent="87" >
  <soa freebytes="43241088" totalbytes="49807360" percent="86" />
  <loa freebytes="2621440" totalbytes="2621440" percent="100" />
</tenured>
<refs soft="36" weak="9346" phantom="1" dynamicSoftReferenceThreshold="27"
maxSoftReferenceThreshold="32" />
<time totalms="13.555" />
</af>

```

___ d. This concludes the properties file configuration exercises. You can now stop the server:

```
stopServer server1
```

What you did in this exercise

In this lab, you worked through some samples based on the properties file configuration utilities that were introduced in WebSphere Application Server V7. You learned how to extract a properties file, based on a server's configuration – both at the cell level, and using filters. You also practiced using commands to validate and apply configuration properties from a properties file.