IBM Software Group

Introduction to Properties-based configuration and Examples

Sangwon Oh (sangwon@us.ibm.com)
Yan Zhao (yanzhao@us.ibm.com)
WebSphere Application Server Support Engineer
October 1, 2013

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS™

Agenda

- *Overview*
- *Limitations*
- *PropertiesBasedConfiguration Commands*
- *Examples of User Scenario*
- *Troubleshooting and debugging*
- *Useful Links*
- Additional WebSphere Product Resources/Links
- *Questions and answer*

Overview

- PropertiesBasedConfiguration was introduced in WebSphere Application Server v7.0
- Consists of a group commands to query and apply configuration objects
- Requires no prior knowledge of scripting language
- Easy to use and modify configuration
- Helps administrators to automate tasks quickly
- Faster to implement than scripting
- Not a replacement for configuration archives

Limitations

- Cannot clone the current environment by extracting whole cell/server/cluster
- z/OS and distributed configuration are not interchangeable
- Following configurations cannot be configured using properties file.
  - SIBus
  - Business-level applications
  - Key and trust store files
  - Web services policy set files
  - File-based user registry

PropertiesBasedConfiguration command

- Some usage of PropertiesBasedConfiguration
  - You can extract the configuration attributes and values from your environment to properties files.
  - Copy configuration properties from one environment to another.
  - Troubleshoot configuration issues.
  - Apply one set of configuration properties across multiple profiles, nodes, cells, servers, or applications.
  - Change the scope of configuration objects

PropertiesBasedConfiguration Commands

- extractConfigProperties
  - extracts configuration data to a human readable text file
- validateConfigProperties
  - validates and determines whether properties file can be applied to environment or not
- applyConfigProperties
  - applies configuration from properties file to current environment
- deleteConfigProperties
  - deletes configuration objects using properties file
- createPropertiesFileTemplates
  - Creates template properties files which are used to create or delete specific object types

PropertiesBasedConfiguration Commands

- AdminTask.extractConfigProperties
  - Target object
  - Required parameter
    - -propertiesFileName (String)
  - Optional parameter
    - -configData (String)
    - -filterMechanism (String)
      - All
      - NO_SUBTYPES
      - SELECTED_SUBTYPES
    - -selectedSubTypes (String)
    - -options (Properties)

PropertiesBasedConfiguration Commands

- - extracting node configuration
  AdminTask.extractConfigProperties('[-propertiesFileName nodeConfig.props -configData Node=MyNode]')
- - extracting server configuration
  AdminTask.extractConfigProperties('[-propertiesFileName serverconfig.props -configData Server=server1]')
- - extracting WebContainer and EJBContainer configuration from server
  AdminTask.extractConfigProperties('[-propertiesFileName serverConfig.props -configData Server=server1 -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [EJBContainer WebContainer]]')
- - extracting server configuration using config ID
  serverID = AdminConfig.getid("/Node:MyNode01/Server:MyServer1/")
  'MyServer1(cells/dmgrCell/nodes/MyNode01/servers/server1|server.xml#Server_1378420059596)'
  AdminTask.extractConfigProperties(serverID, '-propertiesFileName serverConfig.props')

PropertiesBasedConfiguration Commands

- AdminTask.validateConfigProperties
  - Required parameter
    - -propertiesFileName (String)
  - Optional parameter
    - -variablesMapFileName (String)
    - -variablesMap (Properties)
    - -reportFileName (String)
    - -reportFilterMechanism (String)
    - -zipFileName (String)

PropertiesBasedConfiguration Commands

- Validate cell config
  - AdminTask.validateConfigProperties('[-propertiesFileName cellconfig.props -reportFileName report.txt -reportFilterMechanism Errors_And_Changes]')

PropertiesBasedConfiguration Commands

- AdminTask.applyConfigProperties
  - Required parameter
    - -propertiesFileName (String)
  - Optional parameter
    - -variablesMapFileName (String)
    - -variablesMap (Properties)
    - -reportFileName (String)
    - -reportFilterMechanism (String)
      - All
      - Errors
      - Errors_And_Changes
    - -validate (Boolean)
    - -zipFileName (String)

PropertiesBasedConfiguration Commands

- Apply server config
  - AdminTask.applyConfigProperties('[-propertiesFileName serverConfig.props]')

PropertiesBasedConfiguration Commands

- AdminTask.deleteConfigProperties
  - Required parameter
    - -propertiesFileName (String)
  - Optional parameter
    - -variablesMapFileName (String)
    - -variablesMap (Properties)
    - -reportFileName (String)
    - -reportFilterMechanism (String)
      - All
      - Errors
      - Errors_And_Changes
    - -validate (Boolean)

PropertiesBasedConfiguration Commands

- AdminTask.deleteConfigProperties(['-propertiesFileName', '*myPropFile.props*'])

PropertiesBasedConfiguration Commands

- AdminTask.createPropertiesFileTemplates
  - Required parameter
    - -propertiesFileName (String)
  - Optional parameter
    - -configType (String)
      - Server
      - ServerCluster
      - Application
      - AuthorizationGroup

PropertiesBasedConfiguration Commands

- AdminTask.createPropertiesFileTemplates(['-propertiesFileName', '*serverTemplate.props*', '-configType', '*Server*'])

Examples of User Scenario

- Scenario 1
  - Description: WAS admin were asked to add custom property to dmgr to enable HTTPOnly due to security vulnerability scanner indicating adminconsole cookies are missing HTTPOnly flags. Since WebContainer component for dmgr is not manageable from admminconsole, the change needs to be made by scripting.
  - Tasks:
    1. wsadmin>AdminTask.extractConfigProperties('[-propertiesFileName dmgrProps.props -configData Server=dmgr -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [WebContainer]]')

Examples of User Scenario

- Scenario 1 (cont'n)

2. Open dmgrProps.props using editor, then look for "WebContainer properties" subsection.

```
/============================================
#
# SubSection 1.0.8 # WebContainer properties
#
ResourceType=WebContainer
ImplementingResourceType=WebContainer
ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!{serverName}:ApplicationServ
er=ID#ApplicationServer_1:WebContainer=ID#WebContainer_1
AttributeInfo=properties(name,value)
#

#
#Properties
#
============================================/
```

Examples of User Scenario

- Scenario 1 (cont'n)

3. Add "com.ibm.ws.webcontainer.httpOnlyCookies=*" custom property at the end of section, and save.

```
/============================================
...
#
#Properties
#
com.ibm.ws.webcontainer.httpOnlyCookies=*
============================================/
```

4. Apply modified dmgrProps.props file to environment and save.

```
wsadmin>AdminTask.applyConfigProperties('[-propertiesFileName dmgrProps.props -validate -
reportFileName report.txt]')
wsadmin>AdminConfig.save()
```

Examples of User Scenario

- Scenario 2
  - Description: WAS Admin were asked to move all 30 datasources over 3 JDBCProviders that has been configured on cell scope to a cluster scope. Changing scope of resources are not facilitated by adminconsole.
  - Tasks:
    1. extract three JDBCProviders using following wsadmin commands

```
jdbc1 = AdminConfig.getid("/JDBCProvider:DB2 Universal JDBC Driver Provider/")
jdbc2 = AdminConfig.getid("/JDBCProvider:Microsoft SQL Server JDBC Driver/")
jdbc3 = AdminConfig.getid("/JDBCProvider:Oracle JDBC Driver/")
AdminTask.extractConfigProperties(jdbc1, '[-propertiesFileName jdbc1.props -options
[[PortablePropertiesFile true]] ]')
AdminTask.extractConfigProperties(jdbc2, '[-propertiesFileName jdbc2.props -options
[[PortablePropertiesFile true]] ]')
AdminTask.extractConfigProperties(jdbc3, '[-propertiesFileName jdbc3.props -options
[[PortablePropertiesFile true]] ]')
```

Examples of User Scenario

- Scenario 2 (cont'n)

2. Manually modify jdbcx.props files using editor, then rename the file to be different. For example, jdbc1.props_modded

   a) Search cell scope and replace it with cluster scope
   - from "Cell=!{cellName}:JDBCProvider="
   - to "Cell=!{cellName}:ServerCluster=!{clusterName}:JDBCProvider="
   b) At the end of the line of each props files, add cluster name
   clusterName=[your_target_cluster_name_to_copy_resources_over]

   Example, clusterName=test_cluster1

3. Apply updated props file to dmgr config by running following commands:

   AdminTask.applyConfigProperties('-propertiesFileName jdbc1.props_modded -validate true')
   AdminTask.applyConfigProperties('-propertiesFileName jdbc2.props_modded -validate true')
   AdminTask.applyConfigProperties('-propertiesFileName jdbc3.props_modded -validate true')
   AdminConfig.save()

Examples of User Scenario

- Scenario 3
  - Description: WAS Administrator was asked to modify HTTP endpoint URL info for deployed application using jython script, however due to WAS limitation, there's no AdminTask command defined for modifying HTTP endpoint URL, it can only be done by adminconsole manually.
  - Tasks:
    1. Extract the properties for the application, MyWebServicesApp
       AdminTask.extractConfigProperties(['-propertiesFileName', 'MyWebServicesApp.props', '-configData', 'Deployment=MyWebServicesApp'])

Examples of User Scenario

- Scenario 3 (cont'n)

2. Look for "URLPrefixMap Section" for desired webservices module and make proper modification

```
/============================================
#
# CWSAD0103I: URLPrefixMap Section: module=MyWebServices-war.war
#
ResourceType=Application
ImplementingResourceType=Application
ResourceId=Cell=!{cellName}:Deployment=!{applicationName}
ExtensionId=ApplicationWebServicesExtension
#

#
#Properties
#
http=http://localhost:9080/MyWebServices
module=MyWebServices-war.war
============================================/
```

Examples of User Scenario

- Scenario 3 (cont'n)

3. Modify "http=http://localhost:9080/MyWebServices" to desired URL and save the props file.

4. Apply modified dmgrProps.props file to environment and save.

```
wsadmin>AdminTask.applyConfigProperties('[-propertiesFileName MyWebServicesApp.props -
validate -reportFileName report.txt]')
wsadmin>AdminConfig.save()
```

Examples of User Scenario

- Scenario 4
  - Description: WAS Adminitrator was asked to replicate MQQueueConnectionFactory MyMQQCF defined for ClusterA to a different environment, but there are others defined with exact same name on different clusters. Since "-configData MQQueueConnectionFactory=MyMQQCF" will return multiple results, she need to figure out how to extract properties from exact cluster scope.
  - Tasks:

    1. Extract MQQueueConnectionFactory from ClusterA

       AdminTask.extractConfigProperties('[-propertiesFileName MyMQQCF.props -configData "ServerCluster=ClusterA:JMSProvider=WebSphere MQ JMS Provider:MQQueueConnectionFactory=MyMQQCF" -options [[PortablePropertiesFile true]]]')

Examples of User Scenario

2. Modify "Environment Variables" section to match new environment

```
/============================================
EnvironmentVariablesSection
#
#
#Environment Variables
clusterName=ClusterA
cellName=dmgrCell
============================================/
```

3. Apply updated props file to dmgr config by running following commands.

```
AdminTask.applyConfigProperties('-propertiesFileName jdbc1.props_modded -
    validate true')
```

Examples of User Scenario

- Scenario 5
  - Description: WAS Adminitrator is trying to update application to map webserver1 to web module using ProperyFileBasedConfig feature.
  - Tasks:

    1. Extract the properties for the application, MyApplication

```
AdminTask.extractConfigProperties(['-propertiesFileName',
    'MyApplication.props', '-configData', 'Deployment=MyApplication' ])
```

Examples of User Scenario

2. Open MyApplication.props file using editor and look for "MapModulesToServers" subsection. For example.

```
/============================================
# SubSection 1.0.2 # MapModulesToServers Section. taskName and row0 should not be
      edited. row0 contains column names for the task.
#
ResourceType=Application
ImplementingResourceType=Application
ResourceId=Cell=!{cellName}:Deployment=!{applicationName}
#
#Properties
#
taskName=MapModulesToServers
row1={"MyApplicationWeb" MyApplicationWeb.war,WEB-INF/web.xml
WebSphere:cell=!{cellName},cluster=!{clusterName} 22 moduletype.web "Web Module"}
...
============================================/
```

Examples of User Scenario

3. Modify target server to add webserver using variable such as the following:

```
/=============================================

row1={MyApplicationWeb MyApplicationWeb.war,WEB-INF/web.xml
    WebSphere:cell=!{cellName},cluster=!{clusterName}+WebSphere:cell=!{cellName},node=!{node
    Name},server=!{serverName} 22 moduletype.web "Web Module"}

=============================================/
```

4. Add following environment variables

```
/=============================================

EnvironmentVariablesSection

#

#Environment Variables

applicationName=MyApplication

clusterName=ClusterA

cellName=dmgrCell

serverName=webserver1

nodeName=Node01

=============================================/
```

Examples of User Scenario

6. Apply modified dmgrProps.props file to environment and save.

```
AdminTask.applyConfigProperties('[-propertiesFileName MyApplication.props -
    validate true -reportFileName report.txt]')
AdminConfig.save()
```

Examples of User Scenario

- Scenario 6
  - Description: WAS Administrator was asked to copy over all the JAAS - J2C authentication data from one environment to another environment using ProperyFileBasedConfig feature.
  - Tasks:
    1. Extract the properties for the security config data using JAASAuthData as subtype

```
AdminTask.extractConfigProperties('[-propertiesFileName JaasAuthData.props -
    configData Security= -filterMechanism SELECTED_SUBTYPES -selectedSubTypes
    [JAASAuthData] -options [[PortablePropertiesFile true]] ]')
```

Examples of User Scenario

2. Modify "Environment Variables" section to match new environment:

```
/============================================
EnvironmentVariablesSection
#
#
#Environment Variables
cellName=dmgrCell
============================================/
```

3. Apply updated props file to dmgr config by running following commands.

```
AdminTask.applyConfigProperties('-propertiesFileName JaasAuthData.props -validate true')
AdminConfig.save()
```

Examples of User Scenario

- Scenario 7
  - Description: WAS Administrator was asked to copy over all the Shared libraries from one environment to another environment using PropertiesFileBasedConfig feature.
  - Tasks:

    1. Extract the properties for the security config data using Library as subtype

       AdminTask.extractConfigProperties('[-propertiesFileName ShareLibs.props -configData Cell=dmgrCell -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [Library] -options [[PortablePropertiesFile true]]]')

       Note: for specific shared library for specific scope, following command can be used

       AdminTask.extractConfigProperties('-propertiesFileName MyShareLib.props -configData Cell=dmgrCell:Library=MySharedLib -options [[PortablePropertiesFile true]]')

Examples of User Scenario

2. Modify "Environment Variables" section to match new environment

```
/============================================
EnvironmentVariablesSection
#
#Environment Variables
cellName=dmgrCell
============================================/
```

3. Apply updated props file to dmgr config by running following commands.

```
AdminTask.applyConfigProperties('-propertiesFileName ShareLibs.props -validate true')
AdminConfig.save()
```

Examples of User Scenario

- Scenario 8
  - Description: WAS Administrator wants to copy all of the current JVM custom properties from one server to multiple servers.
  - Tasks:

    1. Extract the properties

       AdminTask.extractConfigProperties('[-propertiesFileName JVM.props -configData Server=some_server -filterMechanism SELECTED_SUBTYPES_AND_EXTENSIONS -selectedSubTypes [JavaVirtualMachine] -options [[PortablePropertiesFile true]]]')

Examples of User Scenario

2. Modify JVM.props, look for "System properties" subsection

```
/=======================================
#
# SubSection 1.0.12.0.4.1 # System properties
#
ResourceType=JavaVirtualMachine
ImplementingResourceType=Server
ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!{serverName}:JavaProcessDef=p
rocessType#null:JavaVirtualMachine=
AttributeInfo=systemProperties(name,value)
#

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

Examples of User Scenario

3. Add custom properties in "Properties" section and save, for example:

```
/=======================================
#
#Properties
#
com.ibm.security.jgss.debug=off
com.ibm.security.krb5.Krb5Debug=off
com.ibm.cacheLocalHost=true

java.net.preferIPv4Stack=true
=======================================/
```

4. Modify "EnvironmentVariablesSection" section prior to apply to other environments

```
/=======================================
#Environment Variables
hostName1=*
coreGroup=DefaultCoreGroup
cellName=MyHostNode01Cell
nodeName=MyHostNode01
hostName=MyHost
serverName=MyServer
=======================================/
```

Examples of User Scenario

5. Apply modified properties file to current or other environment and save

```
AdminTask.applyConfigProperties('[-propertiesFileName somename.props -validate true -
reportFileName myReport.txt]')
AdminConfig.save()
```

Troubleshooting

- Typical exceptions:
  - WASX7015E: Exception running command: "AdminTask.extractConfigProperties('[-propertiesFileName roles.props -configData AuthorizationTableExt= -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [AuthorizationTableExt] -options [[PortablePropertiesFile true]]]')"; exception information:

    com.ibm.websphere.management.cmdframework.CommandValidationException: ADMG0657E: Configuration data AuthorizationTableExt= specified for command extractConfigProperties is invalid
  - WASX7015E: Exception running command: "AdminTask.validateConfigProperties('[-propertiesFileName server.props -reportFileName rep.txt]')"; exception information: com.ibm.ws.management.wasresource.common.WASResourceException: com.ibm.ws.management.wasresource.common.WASResourceException: java.lang.NullPointerException

- Debugging
  - com.ibm.ws.scripting.traceString=com.ibm.ws.scripting.*=all=enabled:com.ibm.ws.management.commands.properties.*=all=enabled:com.ibm.ws.management.wasresource.*=all=enabled
  - wsadmin -conntype none

- MustGather: PropertiesBasedConfiguration problems in WebSphere Application Server http://www-01.ibm.com/support/docview.wss?uid=swg21431591

APARs

- PM40943: PFBC EXTRACT DOES NOT INCLUDE J2CCONNECTIONFACTORIES
  - fixed in 7.0.0.21, 8.0.0.2
- PM55601: ADMINTASK.EXTRACTCONFIGPROPERTIES() TASK WITH JAVAVIRTUALMACHINE CONFIGDATA RESULT IN WASX7015E WITH COMMANDVALIDATIONEXCEPTION
  - fixed in 8.0.0.4
- PM69268: ADMINTASK.EXTRACTCONFIGPROPERTIES NOT ABLE TO EXTRACT THE CONFIGURATION OF JVM PMI CUSTOM MONITORING LEVEL
  - fixed in 8.0.0.6, 8.5.0.2
- PM70282: INSTALL VIA PROPERTIES FILE DOES NOT HANDLE DEFAULT BINDING OPTION PROPERLY
  - fixed in 8.0.0.6
- PM81061: Properties File Based Configuration (PFBC) MAPMODULESTOSERVERS TASK FAILS WHEN VALIDATE=TRUE SET
  - fixed in 8.0.0.7

Useful Links

- **Properties file syntax**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=rxml_prop_file_syntax

- **Managing specific configuration objects using properties files**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=txml_config_prop

- **References for supported WCCM config types and subtypes**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=rxml_7propbasedconfig

- **System administration for WebSphere Application Server V7: Part 4: Properties-based configuration**

  http://www.ibm.com/developerworks/websphere/techjournal/0904_chang/0904_chang.html

- **Extracting properties files using wsadmin scripting**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=txml_7extractprops

- **Validating properties files using wsadmin scripting**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=txml_7validateprops

Useful Links

- **Applying properties files using wsadmin scripting**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=txml_7modifyprops

- **Creating server, cluster, application, or authorization group objects using properties files and wsadmin scripting**

  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=matt&product=was-nd-dist&topic=txml_7propobjects

- **Installing enterprise application files by adding properties files to a monitored directory**

  http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.base.doc/info/aes/ae/trun_app_install_dragdrop_prop.html

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
  http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
  http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community:
  http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM® Education Assistant:
  http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
  http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails:
  http://www.ibm.com/software/support/einfo.html

Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe
2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com
3. **Be connected!**
   Connect with us on Facebook
   Connect with us on Twitter

Questions and Answers