IBMCNX Scripting Documentation

Christoph Stöttner
E-mail: christoph.stoettner@stoeps.de
Blog: http://www.stoeps.de

Abstract

This document is the main documentation of the ibmcnxscripting project. You find tips around jython and how you can create your own scripts for IBM® Connections and IBM® WebSphere® Application Server Administration.

Contents

1 Jython

Jython is the Java implementation of Python. I think it is easy to learn and very powerful!

Jython and Python are interpreting languages and you can test your code in the Jython console or wsadmin. Start Jython and type your code, easy to test and you will get fast results.

There are some good online resources and books to learn the basics of Jython or Python:

1.1 Learning Resources

1.1.1 Online materials

Online Learning (Python) You can use Python learning materials for your first steps! It is very similar to learn Jython.

- `http://www.codecademy.com/`
- `http://learnpythonthehardway.org/book/`

Reference

- `http://www.jython.org/jythonbook/en/1.0/`
- `http://www.jython.org/docs/index.html`

1.1.2 Books

WebSphere Application Server Administration Using Jython (2009)
Authors: Robert A. Gibson, Arthur Kevin McGrath and Noel J. Bergman

The Definitive Guide to Jython: Python for the Java Platform (2010)
Authors: Josh Juneau, Frank Wierzbicki, Leo Soto and Victor Ng

1.2 Language elements

Grouping in Jython is done with an indention of four spaces! Please do not use tab, because i had several issues with code errors on Windows, when i

have used tab grouping.

1.2.1 Comments

Comments in Jython start with # and all text after this sign will be ignored!

1.2.2 Variables

```
# Defining a String
x = 'Hello World'
x = "Hello World Two"

#  Defining an integer
y = 10

#  Float
z = 8.75

# Complex
i = 1 + 8.07j

# Multiple assignment
x, y, z = 1, 2, 3
```

1.2.3 Ranges

```
wsadmin>range(7)
[0, 1, 2, 3, 4, 5, 6]

# Include a step in the range
wsadmin>range(0,10,3)
[0, 3, 6, 9]

# Good base for loops
wsadmin>range(1,11)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

wsadmin>range(20,27)
[20, 21, 22, 23, 24, 25, 26]
```

1.2.4 Lists

```
#List
wsadmin>dbs = ['activities','blogs','communities','dogear']
wsadmin>dbs[1]
'blogs'
```

1.2.5 Dictionaries

```
# Dictionary with Performance Data
wsadmin>minConnections = {'activities':1,'blogs':1}
wsadmin>maxConnections = {'activities':50,'blogs':250}
wsadmin>maxConnections
{'activities': 50, 'blogs': 250}
wsadmin>maxConnections.keys()
['activities', 'blogs']
wsadmin>maxConnections.values()
[50, 250]
wsadmin>maxConnections['blogs']
250
```

1.2.6 if - elif - else

Remember: Grouping of elements is done with an indent of 4 spaces!

```
# Basic
if condition :
    # print or do something
elif other condition :
    # print or do something other
else :
    # print or do completely different

# Example
if value.find( 'CLFWY0217E' ) > -1 :
    print "\t\tuser already converted"
elif value.find( 'CLFWY0212E' ) > -1 :
    print "\t\tuser not found in database"
elif value.find( ' CLFWY0209E' ) > -1 :
    print "\t\tnew identifier '" + data[1] + "' does not exist."
else :
    print '\t\tException value: ' + value
```

1.2.7 Errorhandling

You can catch errors with try and except.

```
try:
    # perform some commands which can raise an exception
except Exception, value:
    # perform exception handling
finally:
    # perform task which must always be completed
```

2 WebSphere Application Server

After installing IBM Connections lots of configuration must be done through ISC (Integrated Solution Console). ISC is a good GUI with lots menus and sometimes long click paths. Configuration is possible through `wsadmin` too, but it is hard to find the necessary commands.

2.1 Console Preferences

2.2 wsadmin

Long casesensitiv commands with mostly casesensitiv parameters. Hard to remember and on Linux® no history to recall commands.

Always start `wsadmin` in WAS_HOME/profiles/Dmgr01/bin!

2.2.1 Start wsadmin

Windows®: wsadmin.bat −lang jython −username wasadmin −password password

Linux and AIX®: ./wsadmin.sh −lang jython −username wasadmin −password password

2.2.2 wsadmin set jython as default language

You can set the default language which is used with wsadmin, so you can save some character in typing.

edit WAS_HOME/profiles/Dmgr01/properties/wsadmin.properties

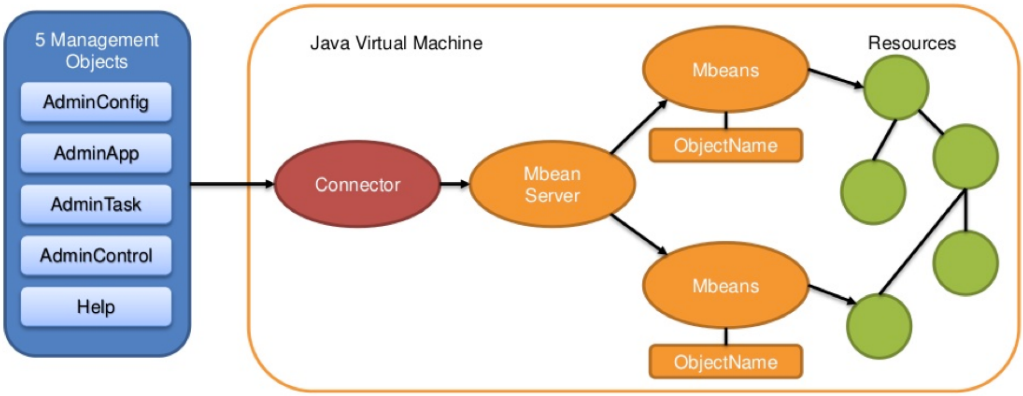
Default: `com.ibm.ws.scripting.defaultLang=JACL`

Set Jyton: `com.ibm.ws.scripting.defaultLang=jython`

2.3 Wsadmin commands

There are five management objects within wsadmin. These objects group the commands to the different MBean Servers.

- AdminConfig
- AdminApp
- AdminTask
- AdminControl
- Help



2.3.1 AdminApp Object

Use the AdminApp object to

- Installing and uninstalling applications
- Listing applications
- Editing applications or modules

Examples

- List of all applications
  - **print** AdminApp.list()
  - AdminApp.list()
  - list =AdminApp.list().split('\n')
- Change options of applications
  - AdminApp.edit('appname',['options'])

2.3.2 AdminConfig

- manage the configuration information that is stored in the repository
- Example change min- and maxConnections of the DataSource Blogs

```
wsadmin>AdminConfig.getid('/DataSource:blogs/')
'blogs(cells/cnxwas1Cell01|resources.xml#DataSource_1371479885975)'
wsadmin>dataSource1=AdminConfig.getid('/DataSource:blogs/')
wsadmin>print AdminConfig.show(dataSource1)
[authDataAlias blogsJAASAuth]
[authMechanismPreference BASIC_PASSWORD]
[connectionPool (cells/cnxwas1Cell01|resources.xml#ConnectionPool_1384252180672)]
    [datasourceHelperClassname com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper]
    [description "Blogs DB2 DataSource"]
[...]
[jndiName jdbc/rollerdb]
[name blogs]
[...]
[provider blogsJDBC(cells/cnxwas1Cell01|resources.xml#JDBCProvider_1371479882710)]
    [providerType "DB2 Universal JDBC Driver Provider"]
[statementCacheSize 100]
wsadmin>AdminConfig.modify( dataSource1, '[[statementCacheSize 50]]')
''
wsadmin>AdminConfig.modify( dataSource1, '[[connectionPool [[minConnections 10]
    [maxConnections 100]]]]' )
''
wsadmin>AdminConfig.save()
''
```

3 IBM Connections wsadmin commands

Each application need its own commands:

```
execfile("connectionsConfig.py")
execfile("activitiesAdmin.py")
...
```

3.1 Load all administrative commands on startup

Please be careful, when you use this tip, because in multicluster environments, each execfile("appnameAdmin.py") will ask you to select a node, where it will run.

You can create a script with all commands for preloading:

`loadAll.py:`

```
execfile('connectionsConfig.py')
execfile("activitiesAdmin.py")
execfile("blogsAdmin.py")
execfile("communitiesAdmin.py")
execfile("dogearAdmin.py")
execfile("filesAdmin.py")
execfile("forumsAdmin.py")
execfile("homepageAdmin.py")
execfile("newsAdmin.py")
execfile("profilesAdmin.py")
execfile("wikisAdmin.py")
```

Load this script at wsadmin startup:
./wsadmin.sh −lang jython −profile loadAll.py

A Linux Bash Goodies

A.1 .bashrc

```
# Insert ulimit for file limit
ulimit -n 64000
# set WAS_HOME
export WAS_HOME=/opt/IBM/WebSphere/AppServer
# set profileNames
dmgrProfile=Dmgr01
appSrvProfile=AppSrv01
alias dmgrBin='cd $WAS_HOME/profiles/$dmgrProfile/bin'
alias wsadmin='cd $WAS_HOME/profiles/$dmgrProfile/bin;./wsadmin.sh -lang jython'
alias nodeBin='cd $WAS_HOME/profiles/$appSrvProfile/bin'
alias startNode='$WAS_HOME/profiles/$appSrvProfile/bin/startNode.sh'
alias startDmgr='$WAS_HOME/bin/startManager.sh'
alias stopNode='$WAS_HOME/profiles/$appSrvProfile/bin/stopNode.sh'
alias stopDmgr='$WAS_HOME/bin/stopManager.sh'
alias nodeLog='tail -f $WAS_HOME/profiles/$appSrvProfile/logs/nodeagent/SystemOut.log'
alias InfraLog='tail -f $WAS_HOME/profiles/$appSrvProfile/logs/InfraCluster_server1/SystemOut.log'
alias Cluster1Log='tail -f $WAS_HOME/profiles/$appSrvProfile/logs/Cluster1_server1/SystemOut.log'
alias Cluster2Log='tail -f $WAS_HOME/profiles/$appSrvProfile/logs/Cluster2_server1/SystemOut.log'
```