

Welcome to wsadminlib.py

Simplifying configuration of IBM WebSphere Application Server



Configuring the IBM WebSphere Application Server

- Definition: Wide variety of 'configuration tasks' for IBM WebSphere Application Server
 - Create and delete clusters, servers, proxies
 - Install applications, map virtual host, set autostart
 - Set tracing, replication, caching, shared libraries, security
 - Start and stop processes and applications
 - etc.

- Two traditional ways to 'configure'
 - GUI: Browse to administrative console, manually view and click
 - Good for one-time tasks, but slow
 - Command-line: Run automation scripts
 - Good repeatability, accurate, and fast, but complicated syntax

Challenges and opportunities: Scripting syntax

- Opinion: AdminConfig, AdminControl, AdminApp, and AdminTask commands are complex
 - Syntax
 - Input parameters
 - Often encapsulation of JACL
 - Output results and return codes
 - Documentation
 - Error messages

- Wish-List: Simplification
 - Library with python methods to call directly
 - Simple names, simple parameters, simple return values
 - Hide AdminConfig, AdminControl, and AdminTask
 - Methods to copy/paste to your own scripts
 - Command 'reference' embodied in working examples

Solution: Introducing wsadminlib.py

- Free sample script package from IBM developerWorks
 - <http://www.ibm.com/developerworks/websphere/library/samples/SampleScripts.html>
 - Search for 'wsadminlib'
- Explicitly designed to simplify configuring the IBM WebSphere Application Server
- wsadminlib.py is one huge single python file
 - More than 500 methods
 - Intuitive method names
 - Intuitive parameter names
 - Easy-to-parse results
- Easily used:
 - Directly: call methods from your scripts
 - Sample: copy-paste methods
 - Reference: easy to search single file
- Caveat: Not supported. Sample script packages are not supported by IBM.

History of wsadminlib.py

- Originally created in 2006
 - Two IBM product developers wrote and shared a few methods
 - Goal was to share research and hide Admin syntax
- Shared and grew in popularity as IBM-internal community-contribution project
 - More than 30 contributors from diverse product component groups worldwide
 - Totally contributor-driven
- Now used in wide range of IBM-internal environments
 - Automated continuous-test frameworks
 - Individual developers running single commands
- Officially released on IBM developerWorks in April 2010
 - Free, unsupported, sample script package
- Steady growth continues

How to use wsadminlib.py 'manually'

- Connect wsadmin to your running Application Server or Deployment Manager

```
root@ding6:/opt/WAS70# bin/wsadmin.sh -lang jython -host ding4 -port 8880
```

```
WASX7209I: Connected to process "server1" on node ding4V8DefaultNode1 using SOAP connector;
```

```
The type of process is: UnManagedProcess
```

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

```
wsadmin>
```

- Access wsadminlib.py by execfile or import

```
wsadmin>execfile('/home/ding/tmp/wsadminlib.py')
```

```
$Id: test.wsadminlib.py 104 2010-02-15 19:06:18Z dingsor $
```

```
wsadmin>
```

- Call a method

```
wsadmin>getCellName()
```

```
'ding4DefaultCell1'
```

```
wsadmin>
```

More simple wsadminlib.py methods

```
wsadmin>listNodes()  
['ding4V8DefaultNode1']
```

Note: Result is returned in a python list of strings. Easy to parse and iterate.

```
wsadmin>serverStatus()
```

Server status

=====

NODE ding4CellManager01 on ding4.raleigh.ibm.com (windows) - Deployment manager <== *Connected to a dmgr*

NODE ding4Node01 on ding4.raleigh.ibm.com (windows)

APPLICATION_SERVER fritzserver running

NODE_AGENT nodeagent running

APPLICATION_SERVER server1 stopped

APPLICATION_SERVER sippbx1 stopped

APPLICATIONS:

commsvc.pbx

fritzlet_ear

pxyapps

Note: serverStatus() is one of few commands intended for human consumption. Great when starting manual operations.

How to use wsadminlib.py from a script - using execfile

```
#-----  
# enableTracing.py  
# Sets the trace specification on all existing application servers.  
#-----  
execfile('wsadminlib.py')  
  
# Enable verbose messages from wsadminlib.py  
enableDebugMessages()  
  
# Define the desired trace specification  
sipTraceSpec = "*=info:com.ibm.ws.sip.*=all"  
  
# Set the trace spec on each application server.  
appServerList = listAllAppServers()    <== Returns list of lists: [ ['node1','server1'], ['node2','server2'], etc]  
for appServer in appServerList:  
    nodename = appServer[0]  
    servername = appServer[1]  
    setServerTrace( nodename, servername, traceSpec = sipTraceSpec )    <== Plus many optional parameters  
  
# Save and sync  
save()
```


How to use wsadminlib.py from a script - using import

```
#-----  
# enableTracing.import.py  
# Sets the trace specification on all existing application servers.  
#-----  
import wsadminlib  
  
# Enable verbose messages from wsadminlib.py  
wsadminlib.enableDebugMessages()  
  
# Define the desired trace specification  
sipTraceSpec = "*=info:com.ibm.ws.sip.*=all"  
  
# Set the trace spec on each application server.  
appServerList = wsadminlib.listAllAppServers()  
for appServer in appServerList:  
    nodename = appServer[0]  
    servername = appServer[1]  
    wsadminlib.setServerTrace( nodename, servername, traceSpec = sipTraceSpec )  
  
# Save and sync  
wsadminlib.save()
```

Dozens of powerhouse methods in wsadminlib.py

- Set custom properties
- Define core groups and bridges
- Create SIBuses
- Configure shared libraries
- Define virtual host aliases
- Make an application start automatically. Or not.
- Configure security
- Configure BLA
- Create/start/stop/delete clusters and servers
- etc.
- etc.

Ramping-up with wsadminlib.py

- Configuration size limits: None
- Documentation: Method prologues (pydoc) and self-documenting parameters
- Searching for functions:
 - Grep for likely method names
 - Grep for underlying Admin command syntax
- Save and Sync: Always syncs
- Version Compatibility:
 - WAS V8 and V7 well-proven. WAS V6 used less.
 - Also works with Extreme Scale (nee XD), Process Server, Cloud Editions
- Modifications:
 - You may modify wsadminlib freely
 - Balance changes against diffing future updates
- Language: English only
- Support: None. Unsupported sample script.

Conclusions and recommendations for wsadminlib.py

- Read an 'unbiased' review
 - https://www.ibm.com/developerworks/mydeveloperworks/blogs/cloudview/entry/tools_from_the_experts85
- Subscribe to wsadminlib blog.
 - <http://wsadminlib.blogspot.com/>
- Get wsadminlib.py
 - <http://www.ibm.com/developerworks/websphere/library/samples/SampleScripts.html>
 - Search for 'wsadminlib'
- Try it
 - Vast and incredible resource

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2011. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.