Saving My Time Using Scripts

Speed up IBM Connections Administration and Configuration

DANNOTES 2013

Christoph Stöttner

IBM Software Consultant

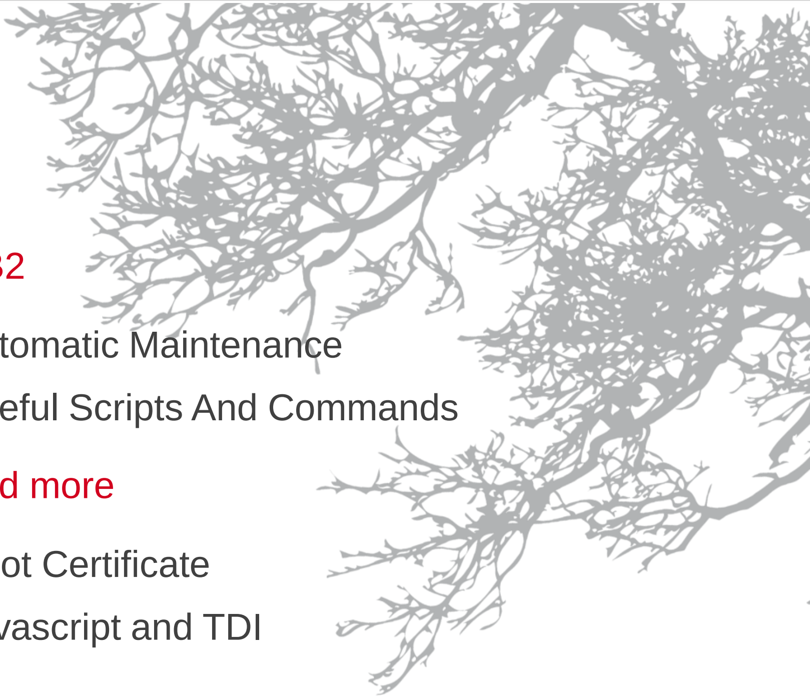
>> Christoph Stöttner

IBM Software Consultant

- IBM Connections since 2.5
- Domino / Windows / Linux Admin
- My Blog: http://www.stoeps.de
- Twitter: @stoeps
- Mail: cstoettner@fum.de
- more accounts: http://about.me/stoeps

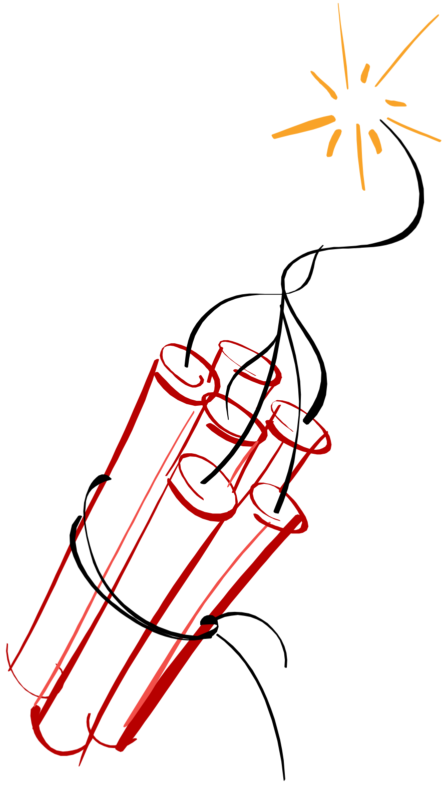
>> Agenda

1. IBM Connections Administration
   - Integrated Solution Console
   - Wsadmin
2. WebSphere Application Server
   - Jython
   - wsadmin Properties
   - Useful Scripts
3. DB2
   - Automatic Maintenance
   - Useful Scripts And Commands
4. And more
   - Root Certificate
   - Javascript and TDI



>> Caution

- With scripts
  - Shell / BASH / ZSH / KSH / SH
  - Jython / JACL
  - Powershell / Batch / VB
  - SQL
- You can...
  - Save a lot of time!
  - **change many things in seconds!**



>> Disclaimer

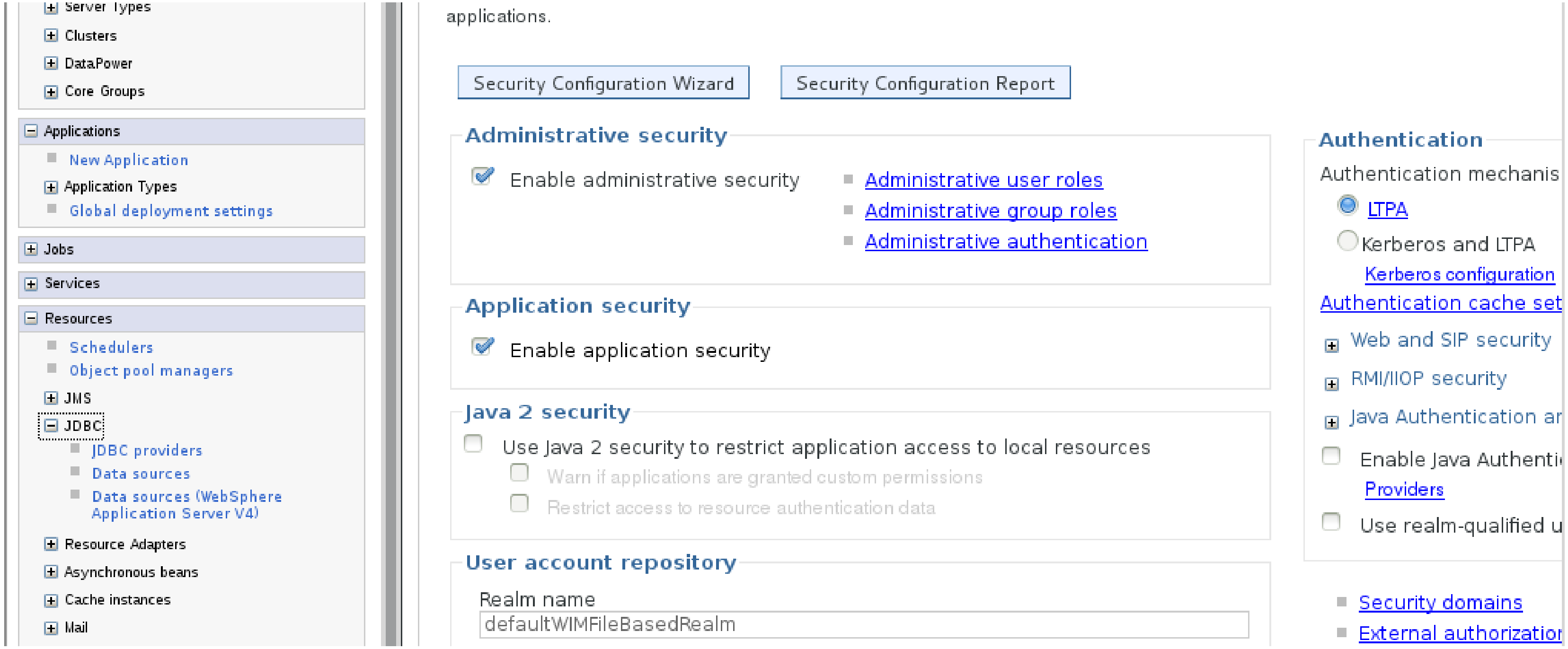
Use all scripts i show in this slides or you download from my repositories WITHOUT WARRANTY and on your own risk!

- TIPPS:
  - Be Careful! Think twice!
  - Create Backups
  - Create a Testsystem
  - Make a documentation of your changes

>> IBM Connections Administration

FRITZ & MACZIOL group

>> Integrated Solution Console



⊞ Server Types
⊞ Clusters
⊞ DataPower
⊞ Core Groups

⊟ Applications
- New Application
- ⊞ Application Types
- Global deployment settings

⊞ Jobs

⊞ Services

⊟ Resources
- Schedulers
- Object pool managers
- ⊞ JMS
- ⊟ JDBC
  - JDBC providers
  - Data sources
  - Data sources (WebSphere Application Server V4)
- ⊞ Resource Adapters
- ⊞ Asynchronous beans
- ⊞ Cache instances
- ⊞ Mail

applications.

Security Configuration Wizard | Security Configuration Report

Administrative security

[✓] Enable administrative security
- Administrative user roles
- Administrative group roles
- Administrative authentication

Application security

[✓] Enable application security

Java 2 security

[ ] Use Java 2 security to restrict application access to local resources
[ ] Warn if applications are granted custom permissions
[ ] Restrict access to resource authentication data

User account repository

Realm name
defaultWIMFileBasedRealm

Authentication

Authentication mechanis
[● LTPA]
○ Kerberos and LTPA
Kerberos configuration
Authentication cache set
⊞ Web and SIP security
⊞ RMI/IIOP security
⊞ Java Authentication ar
[ ] Enable Java Authenti
Providers
[ ] Use realm-qualified u

- Security domains
- External authorizatior

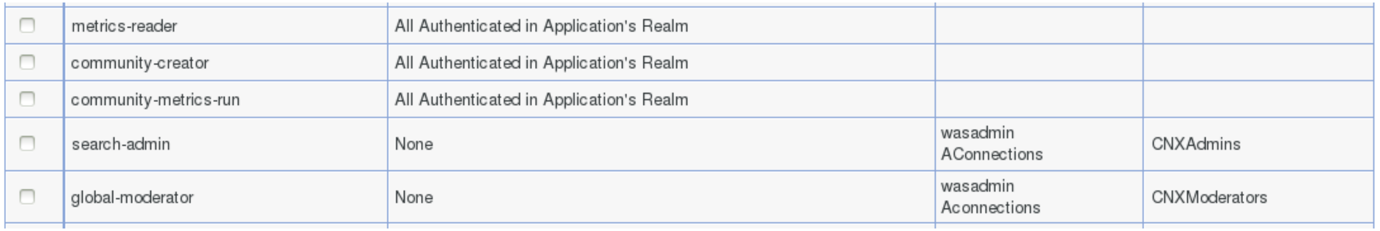
Part of Imtech

>> Save the mice!

I-Ta Tsai – via Flickr – CC BY-NC-SA 2.0

>> Integrated Solution Console

- Browserbased GUI for IBM WebSphere Application Server
- My Mouse pointer runs miles during a Connections Installation
  - 90% on Postinstall Tasks within ISC
- ~~Some~~ tasks are boring
  - Performance Tuning of DataSources
  - Setting Security Roles on Applications (Connections + FEB + CCM = 24 Apps)
    - Checklist needed or you miss out an application



☐	metrics-reader	All Authenticated in Application's Realm		
☐	community-creator	All Authenticated in Application's Realm		
☐	community-metrics-run	All Authenticated in Application's Realm		
☐	search-admin	None	wasadmin AConnections	CNXAdmins
☐	global-moderator	None	wasadmin Aconnections	CNXModerators

>> wsadmin: Configuration of IBM Connections

- Export and validate of Configuration Files (example: LotusConnections-config.xml)
  - Call wsadmin

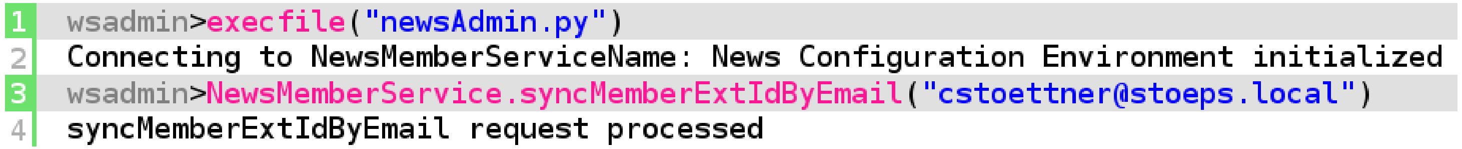
```
cd /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/bin
[root@cnxwas1 bin]$ ./wsadmin.sh -lang jython -username wasadmin -password password
```

  - Cell name, load Connections commands

```
WASX7209I: Connected to process "dmgr" on node cnxwas1CellManager01 using SOAP
connector; The type of process is: DeploymentManager
WASX7031I: For help, enter: "print Help.help()"
wsadmin>AdminControl.getCell()
'cnxwas1Cell01'
wsadmin>execfile("connectionsConfig.py")
Connections Administration initialized
wsadmin>LCConfigService.checkOutConfig('/tmp','cnxwas1Cell01')
# Edit /tmp/LotusConnections-config.xml and save your changes
wsadmin>LCConfigService.checkInConfig('/tmp','cnxwas1Cell01')
Loading schema file for validation: /tmp/LotusConnections-config.xsd
Loading schema file for validation: /tmp/service-location.xsd
/tmp/LotusConnections-config.xml is valid
Connections configuration file successfully checked in
```

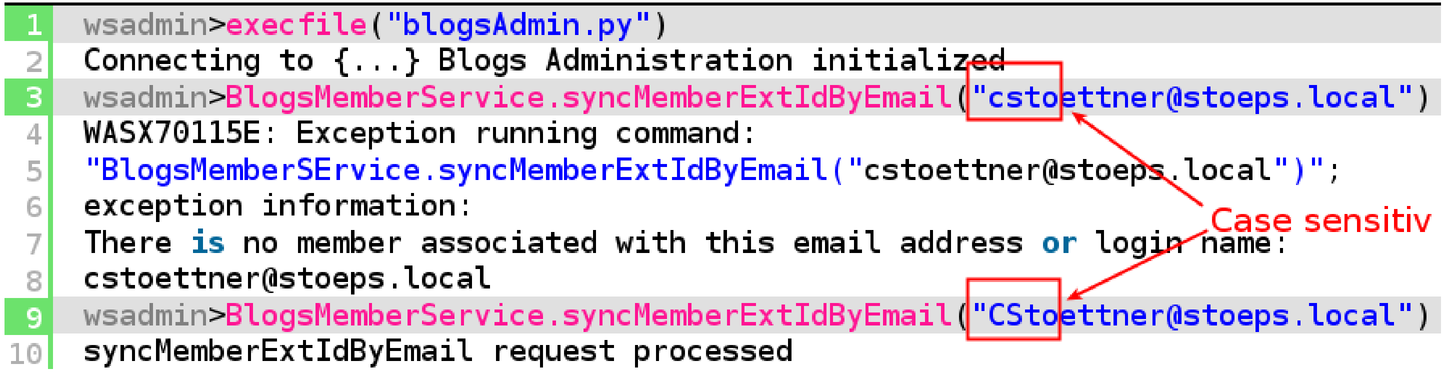
>> wsadmin: synchronize ExID with LDAP

- Synchronize User external ID against LDAP
  - News



```
wsadmin>execfile("newsAdmin.py")
Connecting to NewsMemberServiceName: News Configuration Environment initialized
wsadmin>NewsMemberService.syncMemberExtIdByEmail("cstoettner@stoeps.local")
syncMemberExtIdByEmail request processed
```

  - Blogs



```
wsadmin>execfile("blogsAdmin.py")
Connecting to {...} Blogs Administration initialized
wsadmin>BlogsMemberService.syncMemberExtIdByEmail("cstoettner@stoeps.local")
WASX70115E: Exception running command:
"BlogsMemberSErvice.syncMemberExtIdByEmail("cstoettner@stoeps.local")";
exception information:
There is no member associated with this email address or login name:
cstoettner@stoeps.local
wsadmin>BlogsMemberService.syncMemberExtIdByEmail("CStoettner@stoeps.local")
syncMemberExtIdByEmail request processed
```

>> wsadmin

- Complicated
- long commands
- case sensitiv
  - Jython / JACL commands
  - and parameters
- within Linux no history to recall commands
- Be careful with "Copy & Paste" from Websites
  - often wrong formatted quotation marks
  - Security problem: <span style="visibility:hidden">format c:</span>
- Use a Cheatsheet with
  - often used commands

>> WebSphere Application Server Scripting

Part of Imtech

>> wsadmin Properties - Command Line

- Always execute wsadmin on your Deployment Manager in the bin directory!

```
cd $WAS_HOME/profiles/Dmgr01/bin
```

- Linux | AIX

```
./wsadmin.sh -lang {jython | jacl} -username wasadmin -password password
```

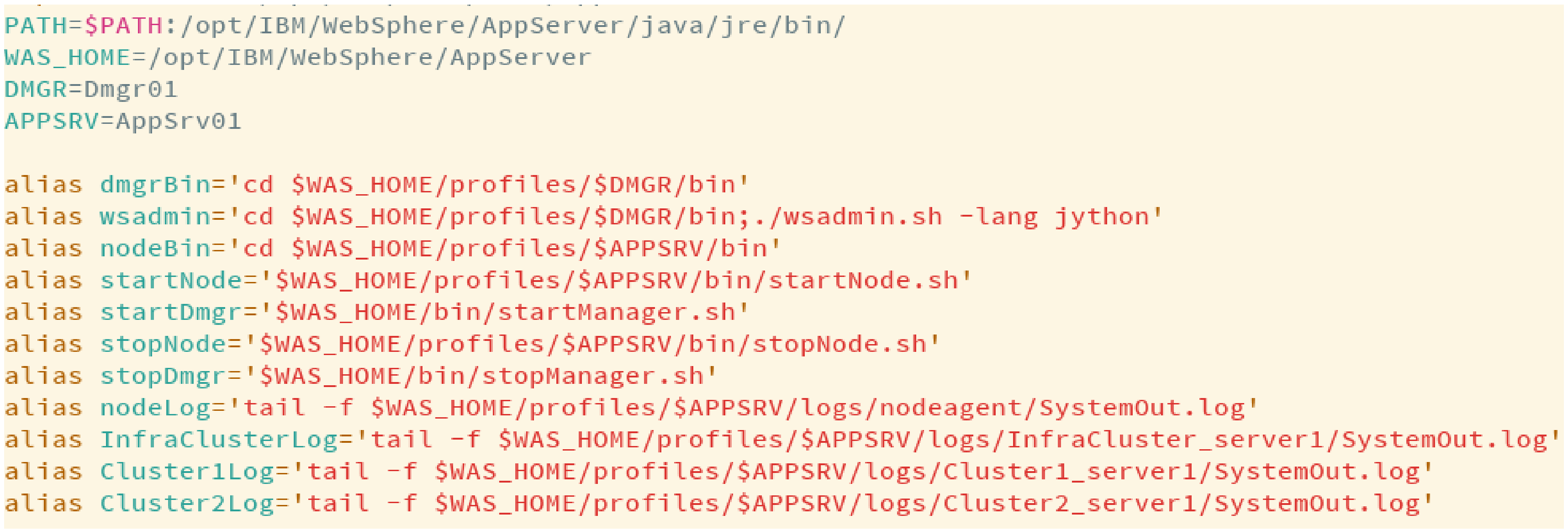
- Windows

```
wsadmin.bat -lang {jython | jacl} -username wasadmin -password password
```

- create Alias or Shell Variable
  - faster access

```
alias wsadmin='cd {WAS_HOME}/profiles/Dmgr01/bin;./wsadmin.sh -lang jython'
```

>> Example .bashrc

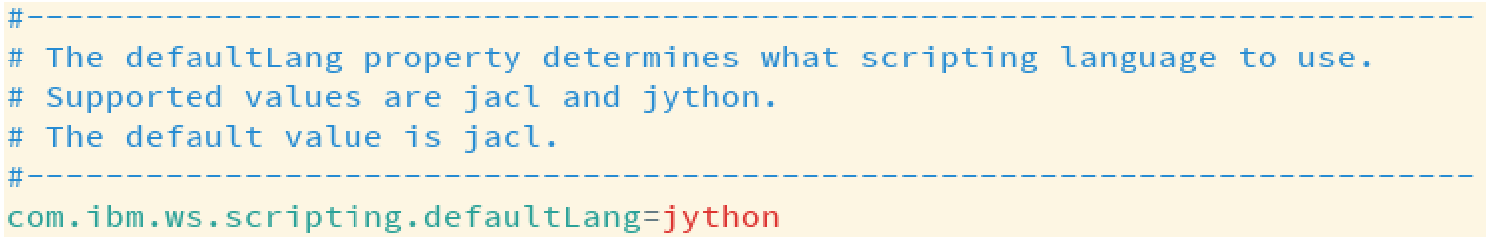


```
PATH=$PATH:/opt/IBM/WebSphere/AppServer/java/jre/bin/
WAS_HOME=/opt/IBM/WebSphere/AppServer
DMGR=Dmgr01
APPSRV=AppSrv01

alias dmgrBin='cd $WAS_HOME/profiles/$DMGR/bin'
alias wsadmin='cd $WAS_HOME/profiles/$DMGR/bin;./wsadmin.sh -lang jython'
alias nodeBin='cd $WAS_HOME/profiles/$APPSRV/bin'
alias startNode='$WAS_HOME/profiles/$APPSRV/bin/startNode.sh'
alias startDmgr='$WAS_HOME/bin/startManager.sh'
alias stopNode='$WAS_HOME/profiles/$APPSRV/bin/stopNode.sh'
alias stopDmgr='$WAS_HOME/bin/stopManager.sh'
alias nodeLog='tail -f $WAS_HOME/profiles/$APPSRV/logs/nodeagent/SystemOut.log'
alias InfraClusterLog='tail -f $WAS_HOME/profiles/$APPSRV/logs/InfraCluster_server1/SystemOut.log'
alias Cluster1Log='tail -f $WAS_HOME/profiles/$APPSRV/logs/Cluster1_server1/SystemOut.log'
alias Cluster2Log='tail -f $WAS_HOME/profiles/$APPSRV/logs/Cluster2_server1/SystemOut.log'
```

FRITZ & MACZIOL group

>> wsadmin: Change Default Language

- `edit {WAS_HOME}/profiles/Dmgr01/properties/wsadmin.properties`
- Default:
  - `com.ibm.ws.scripting.defaultLang=jacl`
- Change to:
  - `com.ibm.ws.scripting.defaultLang=jython`



```
#-------------------------------------------------------------------------
# The defaultLang property determines what scripting language to use.
# Supported values are jacl and jython.
# The default value is jacl.
#-------------------------------------------------------------------------
com.ibm.ws.scripting.defaultLang=jython
```

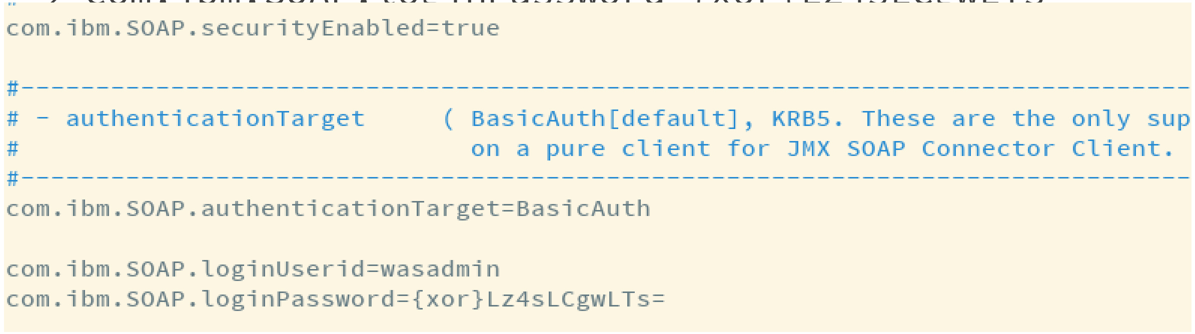
>> wsadmin – Login / Credentials

WAS_HOME}/profiles/Dmgr01/properties/soap.client.props

- Decreases Security (see next slide)
  - com.ibm.SOAP.securityEnabled=true
  - com.ibm.SOAP.loginUserid=wasadmin
  - com.ibm.SOAP.loginPassword=password

PropFilePasswordEncoder.sh soap.client.props com.ibm.SOAP.loginPassword

→ com.ibm.SOAP.loginPassword={xor}Lz4sLCgwLTs=



```
com.ibm.SOAP.securityEnabled=true

#------------------------------------------------------------------------------
# - authenticationTarget     ( BasicAuth[default], KRB5. These are the only sup
#                              on a pure client for JMX SOAP Connector Client.
#------------------------------------------------------------------------------
com.ibm.SOAP.authenticationTarget=BasicAuth

com.ibm.SOAP.loginUserid=wasadmin
com.ibm.SOAP.loginPassword={xor}Lz4sLCgwLTs=
```

Part of Imtech

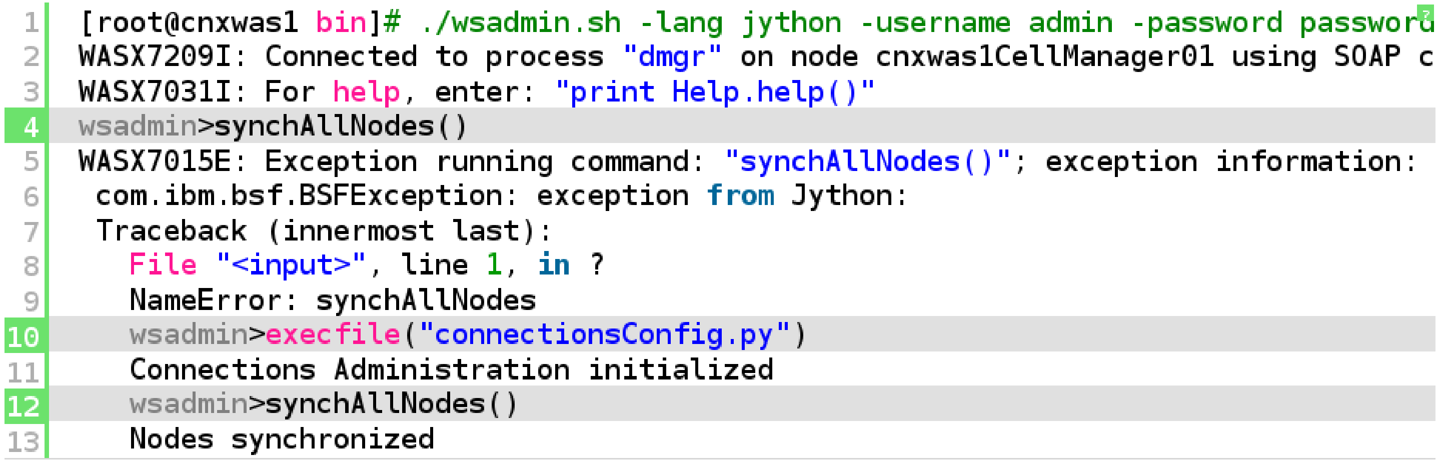
>> WebSphere Password Decoding

- Please do NOT store passwords in productive Environments!
- Passwords are simple XORed with "_" and base64 encoded
- Decryption:
  - Several Webpages: e.g. http://www.sysman.nl/wasdecoder

```
cd WAS_HOME

java/jre/bin/java \
-Djava.ext.dirs=deploytool/itp/plugins/com.ibm.websphere.v8_1.0.201.v20111031_1843/wasJars \
-cp securityimpl.jar:iwsorb.jar com.ibm.ws.security.util.PasswordDecoder \
"{xor}Lz4sLCgwLTs="

encoded password == "{xor}Lz4sLCgwLTs=", decoded password == "password"
```

>> Connections Administration with wsadmin

- Each application need it's own commands
- execfile("scriptname") loads the commands



```
[root@cnxwas1 bin]# ./wsadmin.sh -lang jython -username admin -password password
WASX7209I: Connected to process "dmgr" on node cnxwas1CellManager01 using SOAP c
WASX7031I: For help, enter: "print Help.help()"
wsadmin>synchAllNodes()
WASX7015E: Exception running command: "synchAllNodes()"; exception information:
 com.ibm.bsf.BSFException: exception from Jython:
 Traceback (innermost last):
   File "<input>", line 1, in ?
   NameError: synchAllNodes
   wsadmin>execfile("connectionsConfig.py")
   Connections Administration initialized
   wsadmin>synchAllNodes()
   Nodes synchronized
```

>> Connections Administration Commands

- create a script to load all Connections commands in one step
- call this script
  - within wsadmin:

    ```
    execfile("loadAll.py")
    ```
- call script through com.ibm.ws.scripting.profiles:

  ```
  $WAS_HOME/profiles/Dmgr01/properties/wsadmin.properties
  ```
- Call wsadmin to execute the script

  ```
  ./wsadmin.sh -lang jython -profile loadAll.py
  ```

>> Load all Connections commands

- `loadAll.py`
- save in `$WAS_HOME/profiles/Dmgr01/bin`

```
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")
```

- **Caution:**

  In multinode cluster environments you're asked on which server you want to work!

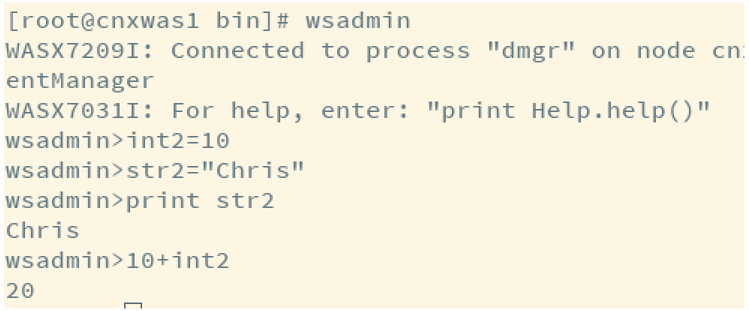
>> Learning Jython

- Easy to learn ~~but~~ and powerful
- Python for the Java Platform
  - http://www.jython.org/jythonbook/en/1.0/
  - http://www.jython.org/docs/index.html
- 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
- Learn Python (similar to Jython)
  - Great online courses on http://www.codecademy.com/ (Python, API, JavaScript)
  - http://learnpythonthehardway.org/book/

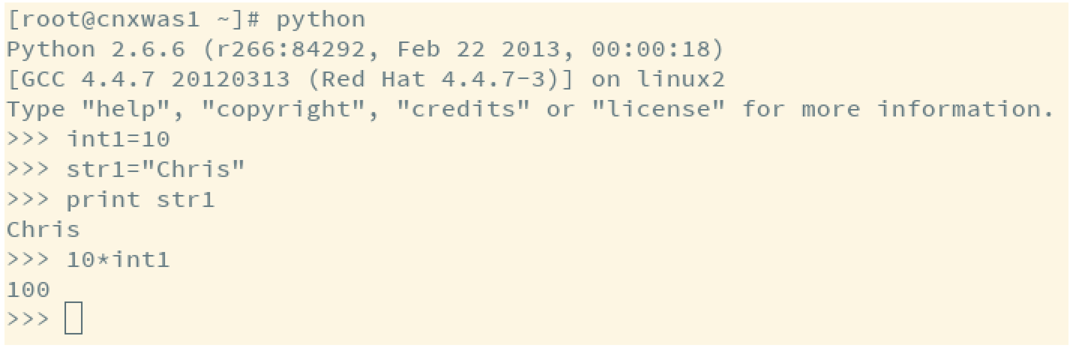
>> Jython Basics

>> Jython Basics

- Readable Code
- Shell and Command Line Interpreter
- Shell good for Testing



```
[root@cnxwas1 bin]# wsadmin
WASX7209I: Connected to process "dmgr" on node cn
entManager
WASX7031I: For help, enter: "print Help.help()"
wsadmin>int2=10
wsadmin>str2="Chris"
wsadmin>print str2
Chris
wsadmin>10+int2
20
```



```
[root@cnxwas1 ~]# python
Python 2.6.6 (r266:84292, Feb 22 2013, 00:00:18)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-3)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> int1=10
>>> str1="Chris"
>>> print str1
Chris
>>> 10*int1
100
>>>
```

Part of Imtech

>> Jython Basics: Variables

- You haven't to declare a type
- String with " or '
- Integer: a number
- Float: a number with .
- # begin a comment

```
# 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
```

>> Jython Basics: Ranges

- Very useful within loops
- returns a list starting with 0
  - range(n) → [0, 1, 2, 3, ..., n-1]
- except you call range with a start parameter
  - range(10,13) → [10, 11, 12]
- third parameter for steps
  - range(10,20,5) → [10, 15]
  - range(10,21,5) → [10, 15, 20]

```
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]
```

>> Jython Basics: Lists and Dictionaries

- List

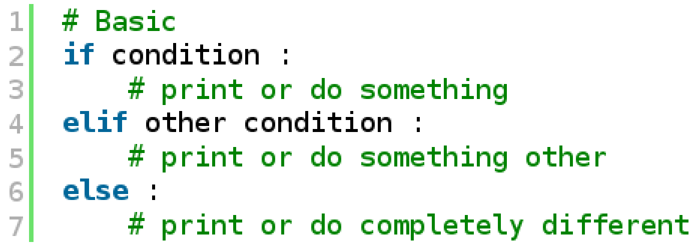
```
#List
wsadmin>dbs = ['activities','blogs','communities','dogear','files','forum']
wsadmin>dbs[1]
'blogs'
```

- Dictionary

```
# Dictionary with Performance Data
wsadmin>minConnections = {'activities':1,'blogs':1,'communities':10,'dogear':1}
wsadmin>maxConnections = {'activities':50,'blogs':250,'communities':200}
wsadmin>maxConnections
{'communities': 200, 'activities': 50, 'blogs': 250}
wsadmin>maxConnections.keys()
['communities', 'activities', 'blogs']
wsadmin>maxConnections.values()
[200, 50, 250]
wsadmin>maxConnections['blogs']
250
```

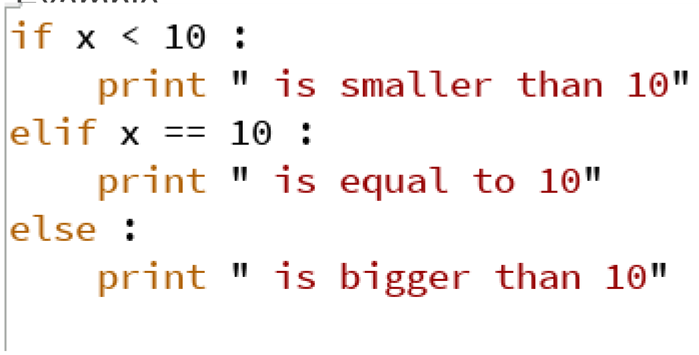
>> Jython Basics: if – elif - else

- Group your code with four spaces



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

- Example



```
if x < 10 :
    print " is smaller than 10"
elif x == 10 :
    print " is equal to 10"
else :
    print " is bigger than 10"
```

>> Jython Basics: Loops

- For Loop

```
# For Loops
dbs = ['activities','blogs','communities','dogear','files','forum','homepage']
for db in dbs: #loop through databases
    print "Database %s" % db
```

- While

```
# While
x = 0
y = 3
while x <= y :
    print 'Value of x is: %d' %(x)
    x += 1
else:
    print 'Completed!'

Value of x is: 0
Value of x is: 1
Value of x is: 2
Value of x is: 3
Completed!
```

>> Jython Basics: Exception Handling

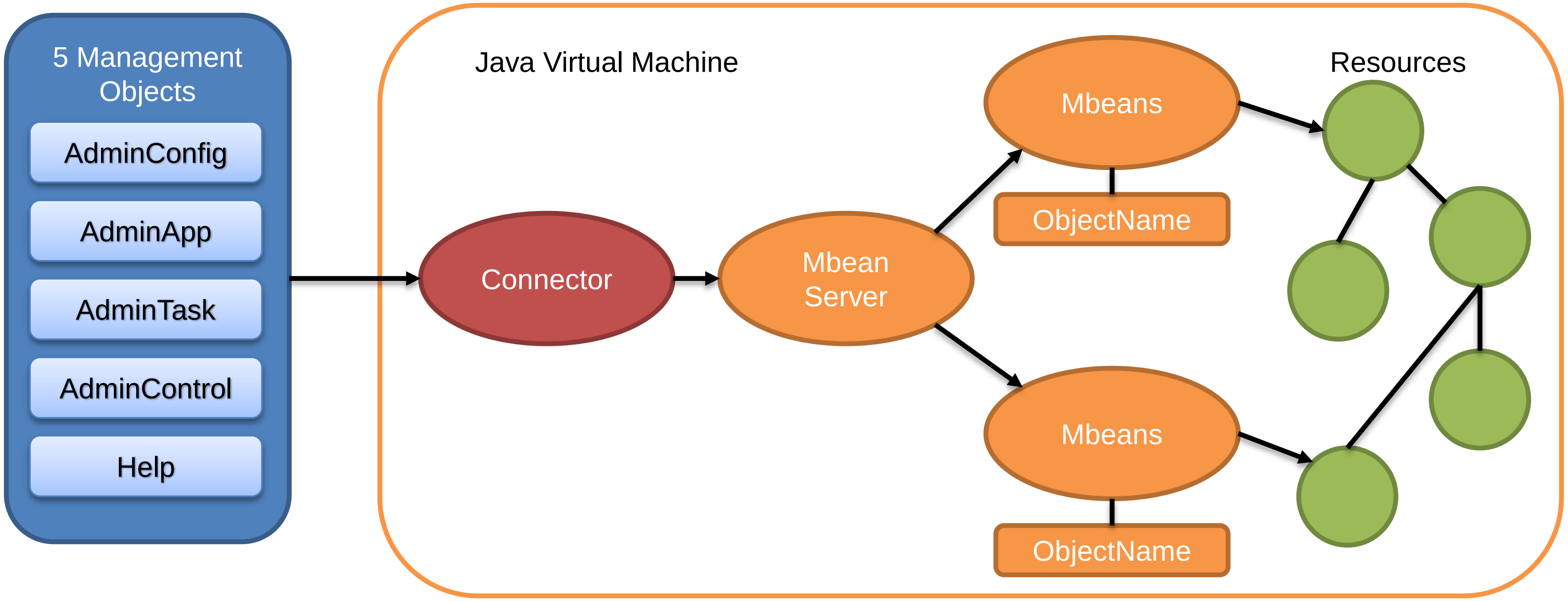
- Scripts will abort, when Exception are raised
- Catch them!

```
try:
    # perform some task that may raise an exception
except Exception, value:
    # perform some exception handling
finally:
    # perform tasks that must always be completed
```

>> WebSphere Jython Commands

Part of Imtech

>> wsadmin commands



Part of Imtech

>> Five Management Objects: AdminApp

- 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'])`

```
wsadmin>print AdminApp.list()

Activities
```

```
wsadmin>list=AdminApp.list().split('\n')

wsadmin>print list

['Activities', 'Blogs', 'Common',
'Communities', 'Dogear', 'FNCS',
'FileNetEngine', 'Files', 'Forms Experience
Builder', 'Forums', 'Help', 'Homepage',
'Metrics', 'Mobile', 'Mobile
Administration', 'Moderation', 'News',
'Profiles', 'Search', 'ViewerApp',
'WebSphereOauth20SP', 'WidgetContainer',
'Wikis', 'connectionsProxy']
```

```
Moderation
News
Profiles
Search
ViewerApp
WebSphereOauth20SP
WidgetContainer
Wikis
connectionsProxy
```

>> Five Management Objects: 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/')
```

>> Five Management Objects: AdminConfig (2)

```
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()
''
```

>> Five Management Objects: AdminControl

- invoke operational commands that manage objects for the application server
- Examples:
  - `AdminControl.getCell()`
  - `AdminControl.queryNames('type=Server,*')`

```
wsadmin>print AdminControl.queryNames('type=Server,*')
WebSphere:name=Cluster1_server1,process=Cluster1_server1,platform=proxy,node=cnxwas1Node01,j2e
eType=J2EEServer,version=8.0.0.5,type=Server,mbeanIdentifier=cells/cnxwas1Cell01/nodes/cnxwas1
Node01/servers/Cluster1_server1/server.xml#Server_1371479529024,cell=cnxwas1Cell01,spec=1.0,pr
ocessType=ManagedProcess
WebSphere:name=Cluster2_server1,process=Cluster2_server1,platform=proxy,node=cnxwas1Node01,j2e
eType=J2EEServer,version=8.0.0.5,type=Server,mbeanIdentifier=cells/cnxwas1Cell01/nodes/cnxwas1
Node01/servers/Cluster2_server1/server.xml#Server_1371479841514,cell=cnxwas1Cell01,spec=1.0,pr
ocessType=ManagedProcess
WebSphere:name=InfraCluster_server1,process=InfraCluster_server1,platform=proxy,node=cnxwas1No
de01,j2eeType=J2EEServer,version=8.0.0.5,type=Server,mbeanIdentifier=cells/cnxwas1Cell01/nodes
/cnxwas1Node01/servers/InfraCluster_server1/server.xml#Server_1371479008625,cell=cnxwas1Cell01
,spec=1.0,processType=ManagedProcess
...
```

>> Five Management Objects: AdminTask

- run administrative commands

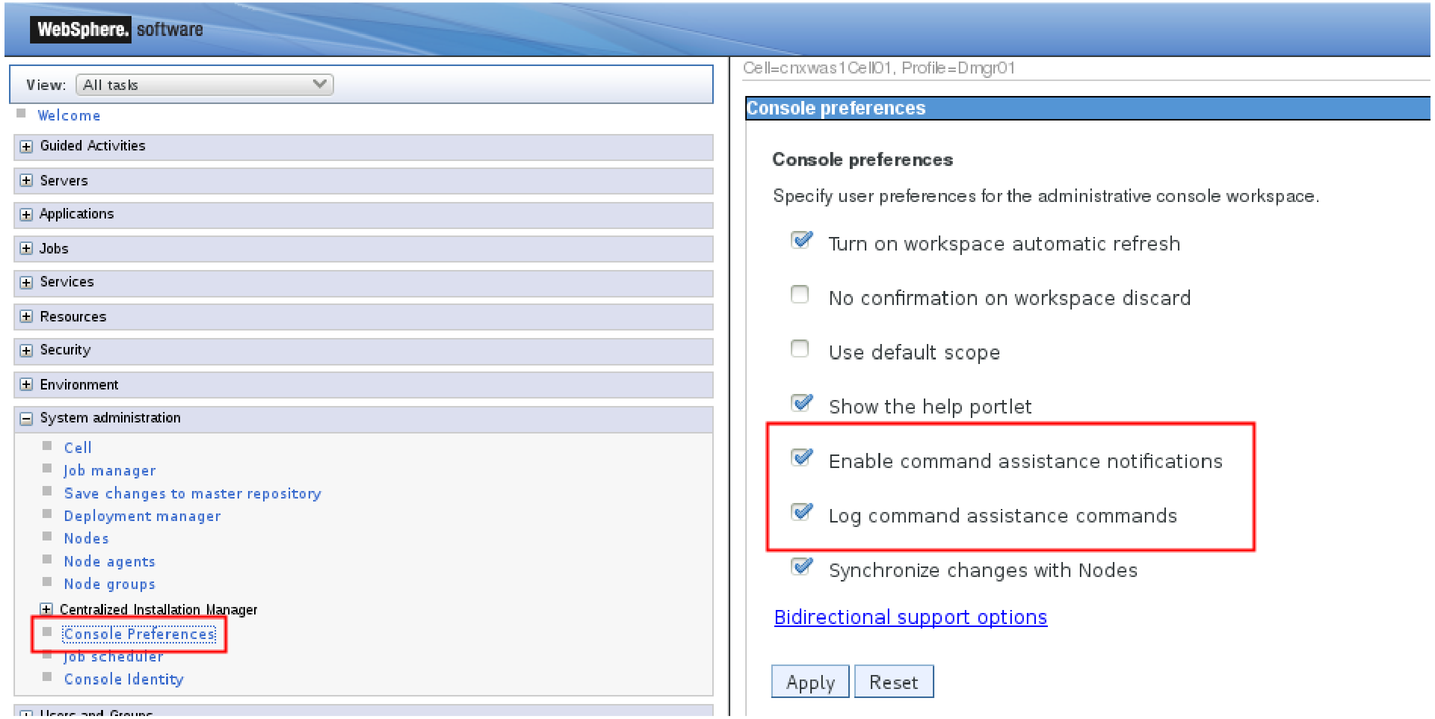
```
wsadmin>print AdminTask.listServers( '[-serverType APPLICATION_SERVER]' )

FEB_server1(cells/cnxwas1Cell01/nodes/cnxwas1Node01/servers/FEB_server1|server.xml)
Cluster1_server1(cells/cnxwas1Cell01/nodes/cnxwas1Node01/servers/Cluster1_server1|server.xml)
Cluster2_server1(cells/cnxwas1Cell01/nodes/cnxwas1Node01/servers/Cluster2_server1|server.xml)
InfraCluster_server1(cells/cnxwas1Cell01/nodes/cnxwas1Node01/servers/InfraCluster_server1|server.xml)
ConversionMember1(cells/cnxwas1Cell01/nodes/cnxdocsNode01/servers/ConversionMember1|server.xml)
ViewerMember1(cells/cnxwas1Cell01/nodes/cnxdocsNode01/servers/ViewerMember1|server.xml)
DocsMember1(cells/cnxwas1Cell01/nodes/cnxdocsNode01/servers/DocsMember1|server.xml)
```

>> Five Management Objects: Help

- Online Help for Jython and JACL Scripting
- Example:
  - print Help.AdminApp()
  - print Help.AdminConfig()

```
wsadmin>print Help.AdminApp()
WASX7095I: The AdminApp object allows application objects to be manipulated
           -- this includes installing, uninstalling, editing, and listing. Most
           of the commands supported by AdminApp operate in two modes: the default
           mode is one in which AdminApp communicates with the WebSphere server to
           accomplish its tasks.  A local mode is also possible, in which no
           server communication takes place.  The local mode operation is invoked
           by bringing up the scripting client with no server connected using the
           command line "-conntype NONE" option tor setting the
           "com.ibm.ws.scripting.connectionType=NONE" property in the
           wsadmin.properties.
```

>> Find Jython commands



WebSphere. software

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
- Jobs
- Services
- Resources
- Security
- Environment
- System administration
  - Cell
  - Job manager
  - Save changes to master repository
  - Deployment manager
  - Nodes
  - Node agents
  - Node groups
  - Centralized Installation Manager
  - Console Preferences
  - Job scheduler
  - Console Identity

Cell=cnxwas1Cell01, Profile=Dmgr01

Console preferences

Console preferences

Specify user preferences for the administrative console workspace.

- [✓] Turn on workspace automatic refresh
- [ ] No confirmation on workspace discard
- [ ] Use default scope
- [✓] Show the help portlet
- [✓] Enable command assistance notifications
- [✓] Log command assistance commands
- [✓] Synchronize changes with Nodes

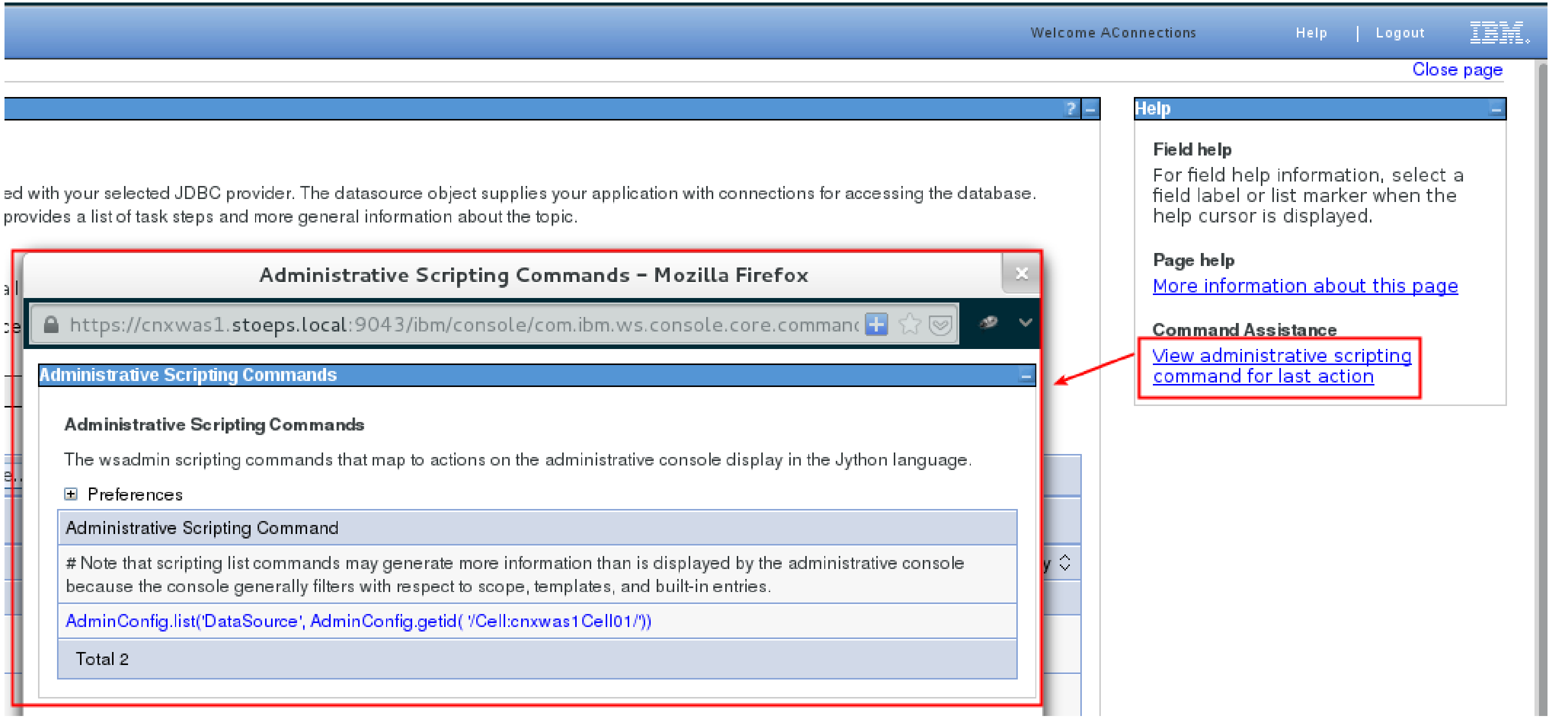
Bidirectional support options

Apply Reset

Part of Imtech

FRITZ & MACZIOL group

>> Enable command assistance notifications



Welcome AConnections Help | Logout

Close page

ed with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database.
provides a list of task steps and more general information about the topic.

Administrative Scripting Commands – Mozilla Firefox

https://cnxwas1.stoeps.local:9043/ibm/console/com.ibm.ws.console.core.comman

Administrative Scripting Commands

Administrative Scripting Commands

The wsadmin scripting commands that map to actions on the administrative console display in the Jython language.

⊞ Preferences

Administrative Scripting Command
Note that scripting list commands may generate more information than is displayed by the administrative console because the console generally filters with respect to scope, templates, and built-in entries.
AdminConfig.list('DataSource', AdminConfig.getid('/Cell:cnxwas1Cell01/'))
Total 2

Help

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

Part of Imtech

>> Log Command Assistance Commands

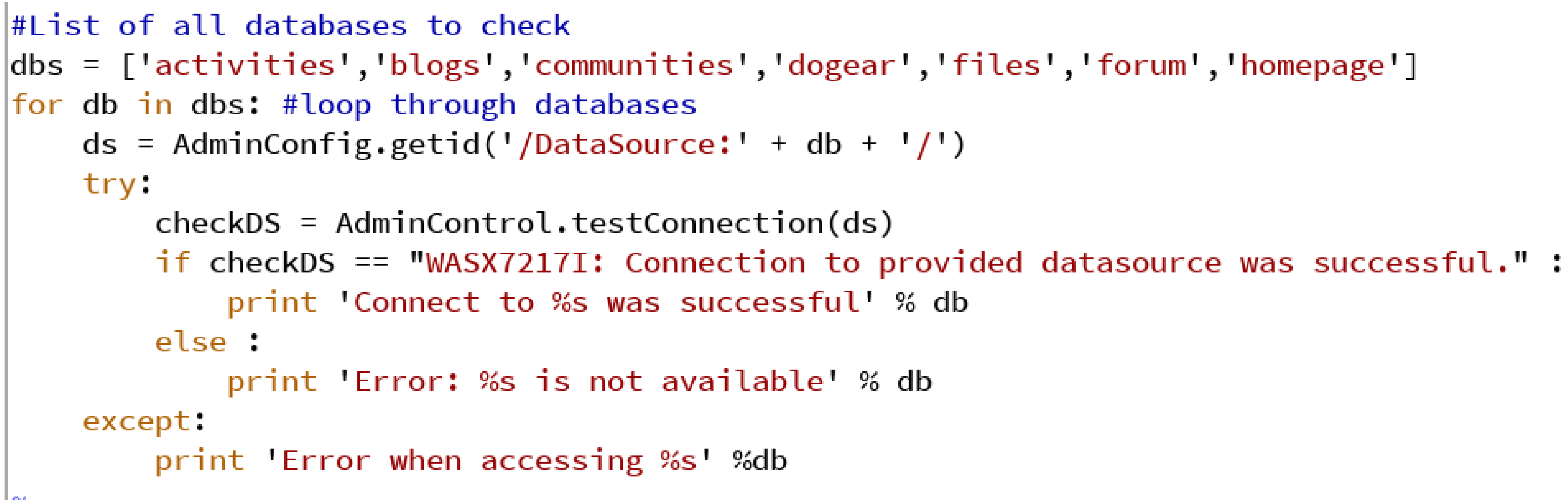
- $WAS_HOME/profiles/Dmgr01/logs/dmgr/commandAssistanceJythonCommands_*username*.log

```
AdminConfig.list('ServerCluster', AdminConfig.getid( '/Cell:cnxwas1Cell01/'))

# [9/20/13 12:45:36:204 CEST] WebSphere application server clusters
AdminControl.invoke('WebSphere:name=InfraCluster,process=dmgr,
platform=common,node=cnxwas1CellManager01,version=8.0.0.5,type=Cluster,
mbeanIdentifier=InfraCluster,cell=cnxwas1Cell01,spec=1.0', 'rippleStart')

# Note that scripting list commands may generate more information than is
# displayed by the administrative console because the console generally filters
# with respect to scope, templates, and built-in entries.

# [9/22/13 19:09:43:718 CEST] DataSource
AdminConfig.list('DataSource', AdminConfig.getid( '/Cell:cnxwas1Cell01/'))
```

>> Example Scripts

>> Test Database Connections

- checkDataSource.py



```
#List of all databases to check
dbs = ['activities','blogs','communities','dogear','files','forum','homepage']
for db in dbs: #loop through databases
    ds = AdminConfig.getid('/DataSource:' + db + '/')
    try:
        checkDS = AdminControl.testConnection(ds)
        if checkDS == "WASX7217I: Connection to provided datasource was successful." :
            print 'Connect to %s was successful' % db
        else :
            print 'Error: %s is not available' % db
    except:
        print 'Error when accessing %s' %db
```

>> Change DataSource Parameters

- Setting Default Parameters as mentioned in IBM Connections 4.0 Performance Tuning Guide
- Dictionary with database names and parameters

```
perf = {'activities':{'minConnections':1,'maxConnections':50},
        'blogs':{'minConnections':1,'maxConnections':250},
        'communities':{'minConnections':10,'maxConnections':200},
        'dogear':{'minConnections':1,'maxConnections':150},
        'files':{'minConnections':10,'maxConnections':100},
        'forum':{'minConnections':50,'maxConnections':100},
        'homepage':{'minConnections':20,'maxConnections':100},
        ...
        'wikis':{'minConnections':1,'maxConnections':100}}
```

- statementCacheSize=100 (für DB2) bzw. 50 (für Oracle)

>> cfgDataSource.py

```
perf = {'activities':{'minConnections':1,'maxConnections':50},
        'blogs':{'minConnections':1,'maxConnections':250},
        'communities':{'minConnections':10,'maxConnections':200},
        'dogear':{'minConnections':1,'maxConnections':150},
        'files':{'minConnections':10,'maxConnections':100},
        'forum':{'minConnections':50,'maxConnections':100},
        'homepage':{'minConnections':20,'maxConnections':100},
        'metrics':{'minConnections':1,'maxConnections':75},
        'mobile':{'minConnections':1,'maxConnections':100},
        'news':{'minConnections':50,'maxConnections':75},
        'profiles':{'minConnections':1,'maxConnections':100},
        'search':{'minConnections':50,'maxConnections':75},
        'wikis':{'minConnections':1,'maxConnections':100}}

statementCacheSize = 100 #change to 50 for oracle

for db in perf.keys(): # Looping through databases
    print 'Change DataSource parameters for: %s' % db.upper()

    t1=AdminConfig.getid('/DataSource:' + db + '/')
    print '  statementCacheSize: ' + str(statementCacheSize)
    print '  minConnections: ' + str(perf[db]['minConnections'])
    print '  maxConnections: ' + str(perf[db]['maxConnections'])
    AdminConfig.modify(t1,'[[statementCacheSize "' + str(statementCacheSize) + '"]]')
    AdminConfig.modify(t1,'[[connectionPool [[minConnections "' + str(perf[db]['minConnections']) + '"]
[maxConnections "' + str(perf[db]['maxConnections']) + '"]]]]')
    AdminConfig.save()
```

>> call MemberService for all applications

- memberSyncByEmail.py
- ./wsadmin.sh –lang jython –f "memberSyncByEmail.py" cstoettner@stoeps.local
  - Mail Address as a parameter

```
MAILADDRESS = sys.argv[0]
print "Syncing MemberService for " + MAILADDRESS
execfile("/opt/install/scripts/loadAll.py")
ActivitiesMemberService.syncMemberExtIdByEmail(MAILADDRESS)
BlogsMemberService.syncMemberExtIdByEmail(MAILADDRESS)
CommunitiesMemberService.syncMemberExtIdByEmail(MAILADDRESS)
DogearMemberService.syncMemberExtIdByEmail(MAILADDRESS)
FilesMemberService.syncMemberExtIdByEmail(MAILADDRESS)
ForumsMemberService.syncMemberExtIdByEmail(MAILADDRESS)
NewsMemberService.syncMemberExtIdByEmail(MAILADDRESS)
WikisMemberService.syncMemberExtIdByEmail(MAILADDRESS)
```

- better:

```
try:
   print "Sync Dogear"
   DogearMemberService.syncMemberExtIdByEmail(MAILADDRESS)
except:
   print 'No user with Email ' + MAILADDRESS +' found'
```

>> Backup Security/J2EE Roles

- Script creates text-files with a backup of the security roles of all applications
  - `./wsadmin.sh -lang jython -f "{path}/securityrolebackup.py" 'backuppath'`

```
apps = AdminApp.list()
appsList = apps.split(lineSeparator) #List of all applications
path = '/opt/install/backup' # must exist!
for app in appsList:
    filename = app + ".txt"
    my_file = open(path + '/' + filename,'w')
    my_file.write (AdminApp.view(app,"-MapRolesToUsers"))
    my_file.flush
    my_file.close()
```

FRITZ & MACZIOL group

>> Restore Security/J2EE Roles

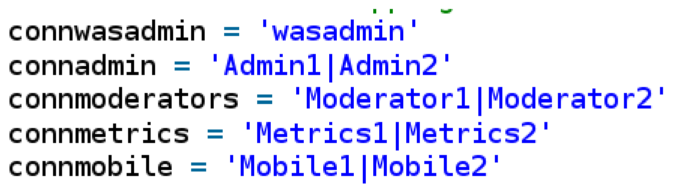
- Script read text-files of Backup
- Good to use before applying Fixes and CR
- Files are converted to dictionaries

```
def setSecurityRoles(dictionary,appName):
    strRoleChange = '['
    for role in dictionary.keys():
        # Loop through Roles
        strRoleChange += '[\"' + role + '\" '
        strRoleChange += dictionary[role]['Everyone?'] + ' '
        strRoleChange += dictionary[role]['All authenticated?'] + ' '
        strRoleChange += '\"' + dictionary[role]['Mapped users'] + '\" '
        strRoleChange += '\"' + dictionary[role]['Mapped groups'] + '\"] '
    strRoleChange += ']]'
    AdminApp.edit(appName, '[-MapRolesToUsers' + strRoleChange +']')
    print "Setting Roles and Users for %s" % appName
    AdminConfig.save()
```

Part of Imtech

>> Alternative to set J2EE Roles

- Script from http://kbild.ch (Blog of Klaus Bild)
- Set Default Security Roles and Restrict Roles to Administrators
- very useful to set roles initially
- i extended the script with groups



```
connwasadmin = 'wasadmin'
connadmin = 'Admin1|Admin2'
connmoderators = 'Moderator1|Moderator2'
connmetrics = 'Metrics1|Metrics2'
connmobile = 'Mobile1|Mobile2'
```

>> Set J2EE Roles consistent on all applications

```
# Variables for Groupmapping
connadmingroup = 'CNXAdmins'
connmoderatorgroup = 'CNXModerators'
connmetricsgroup = 'CNXMetricsAdmins'
connmobilegroup = 'CNXMobileAdmins'

appName = 'Activities'
# "role" Yes No = everyone
# "role" No Yes = All Authenticated
# "role" No No  = None
AdminApp.edit( appName, '[-MapRolesToUsers [["person" No Yes "" ""]
  ["everyone" Yes No "" ""] ["reader" Yes No "" ""]
  ["metrics-reader" No Yes "" ""]
  ["search-admin" No No "' + connwasadmin + '|' + connadmin + '" "' + connadmingroup + '"]
  ["widget-admin" No No "' + connwasadmin + '|' + connadmin + '" "' + connadmingroup + '"]
  ["admin" No No "' + connwasadmin + '|' + connadmin + '" "' + connadmingroup + '"]
  ["bss-provisioning-admin" No No "" ""] ]]' )
print "Setting Roles and Users for Activities"
AdminConfig.save()
...
```

>> IBM DB2 / SQL

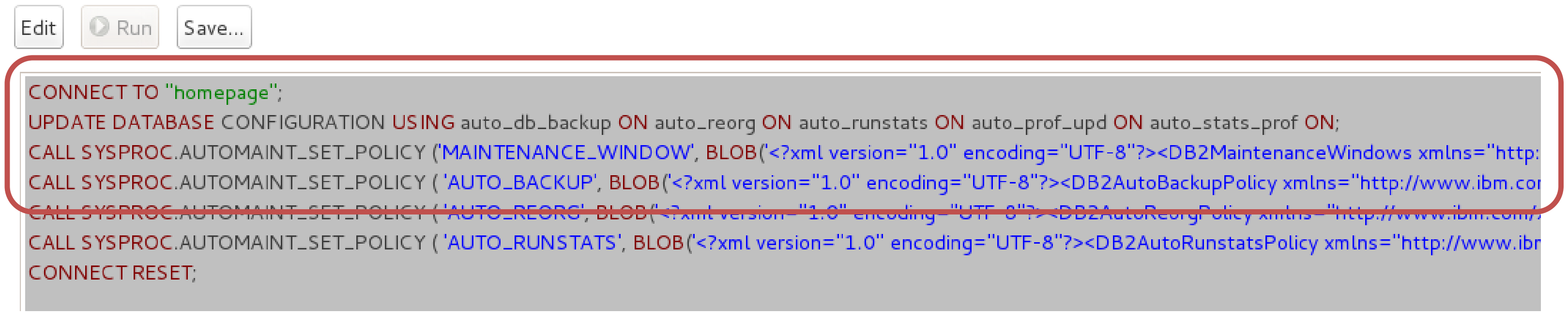
>> DB2 useful commands

- Get a list of all databases of an db2 instance
  - Linux
    - `db2 list database directory | grep alias | awk '{print $4}' | sort`
  - Windows (Powershell)
    - `db2cmd -i -c -w "db2 list database directory | where {$_ -match "alias"} | %{ $_.Split('=')[1]; }"`
- Show active databases
  - `db2 list active databases`

FRITZ & MACZIOL group

>> Automatic Maintenance

- DB2 9.7 with export of Policy Files and reimport to other databases: [Skripting DB2 Automatic Maintenance | Stoeps]
- IBM Data Studio
  - Configure Automatic maintenance and backup on one database (e.g. homepage)
  - Save commands to a sql script



```
CONNECT TO "homepage";
UPDATE DATABASE CONFIGURATION USING auto_db_backup ON auto_reorg ON auto_runstats ON auto_prof_upd ON auto_stats_prof ON;
CALL SYSPROC.AUTOMAINT_SET_POLICY ('MAINTENANCE_WINDOW', BLOB('<?xml version="1.0" encoding="UTF-8"?><DB2MaintenanceWindows xmlns="http:
CALL SYSPROC.AUTOMAINT_SET_POLICY ( 'AUTO_BACKUP', BLOB('<?xml version="1.0" encoding="UTF-8"?><DB2AutoBackupPolicy xmlns="http://www.ibm.cor
CALL SYSPROC.AUTOMAINT_SET_POLICY ( 'AUTO_REORG', BLOB('<?xml version="1.0" encoding="UTF-8"?><DB2AutoReorgPolicy xmlns="http://www.ibm.com/
CALL SYSPROC.AUTOMAINT_SET_POLICY ( 'AUTO_RUNSTATS', BLOB('<?xml version="1.0" encoding="UTF-8"?><DB2AutoRunstatsPolicy xmlns="http://www.ibr
CONNECT RESET;
```

Part of Imtech

>> Automatic Maintenance (2)

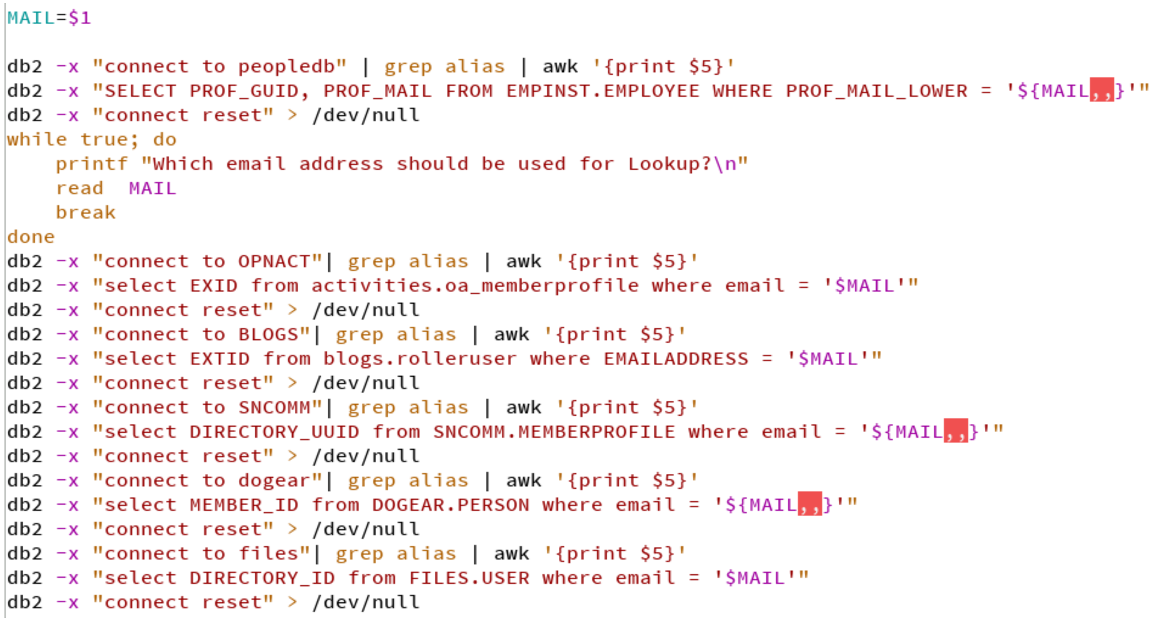
- automaint.sql
  - copy the update line and the 4 call statements
- setmaintenance.sh

```bash
#!/bin/bash

databases=$(db2 list database directory | grep alias | awk '{print $4}' | sort)
for database in ${databases[@]}
do
 echo $database
 db2 "connect to $database"
 db2 -tvf automaint.sql
 db2 "connect reset"
done
```

FRITZ & MACZIOL group

>> Check ExId in all Connections Apps

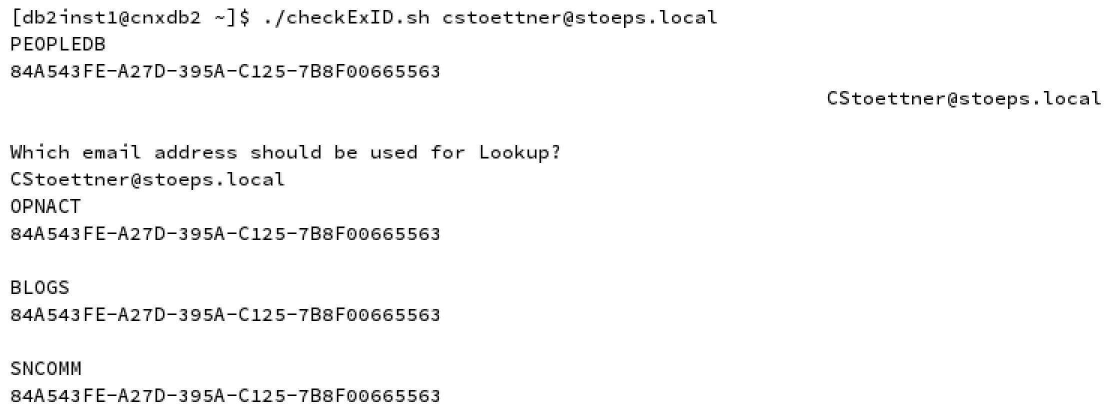
- Select User by Email from empinst.employee
- Search this UserID in all Applications



```
MAIL=$1

db2 -x "connect to peopledb" | grep alias | awk '{print $5}'
db2 -x "SELECT PROF_GUID, PROF_MAIL FROM EMPINST.EMPLOYEE WHERE PROF_MAIL_LOWER = '${MAIL,,}'"
db2 -x "connect reset" > /dev/null
while true; do
    printf "Which email address should be used for Lookup?\n"
    read  MAIL
    break
done
db2 -x "connect to OPNACT"| grep alias | awk '{print $5}'
db2 -x "select EXID from activities.oa_memberprofile where email = '$MAIL'"
db2 -x "connect reset" > /dev/null
db2 -x "connect to BLOGS"| grep alias | awk '{print $5}'
db2 -x "select EXTID from blogs.rolleruser where EMAILADDRESS = '$MAIL'"
db2 -x "connect reset" > /dev/null
db2 -x "connect to SNCOMM"| grep alias | awk '{print $5}'
db2 -x "select DIRECTORY_UUID from SNCOMM.MEMBERPROFILE where email = '${MAIL,,}'"
db2 -x "connect reset" > /dev/null
db2 -x "connect to dogear"| grep alias | awk '{print $5}'
db2 -x "select MEMBER_ID from DOGEAR.PERSON where email = '${MAIL,,}'"
db2 -x "connect reset" > /dev/null
db2 -x "connect to files"| grep alias | awk '{print $5}'
db2 -x "select DIRECTORY_ID from FILES.USER where email = '$MAIL'"
db2 -x "connect reset" > /dev/null
```

>> Check ExId in all Connections Apps

- Select User by Email from empinst.employee
- Search this UserID in all Applications



```
[db2inst1@cnxdb2 ~]$ ./checkExID.sh cstoettner@stoeps.local
PEOPLEDB
84A543FE-A27D-395A-C125-7B8F00665563
                                                                    CStoettner@stoeps.local


Which email address should be used for Lookup?
CStoettner@stoeps.local
OPNACT
84A543FE-A27D-395A-C125-7B8F00665563

BLOGS
84A543FE-A27D-395A-C125-7B8F00665563

SNCOMM
84A543FE-A27D-395A-C125-7B8F00665563
```

Und noch mehr

>> And some more

>> Get SSL Root Certificate

- Export Root certificates of selfsigned certs often complicated
- Useful in e.g.:
  - TDI and LDAPS
  - Domino and Embedded Experience Config
- Prerequist: openssl, java (keytool)
- `./create_cacerts.sh -h hostname -p port -f path/filename`
- TDIPopulation/solution.properties:
  - javax.net.ssl.trustStore=/opt/install/keystore

>> create_cacerts.sh

```
TMP1=`mktemp -d`
trap "rm -rf $TMP1" EXIT

while getopts h:p:f:?: option
do
    case "${option}"
        in
        h) SERVERNAME=${OPTARG};;
        p) SERVERPORT=${OPTARG};;
        f) STORECACERTS=${OPTARG};;
        ?) echo "USAGE: `basename $0` -h hostname -p port -f Certfile\n"
        echo "You have to type a password for keyfile twice and set\n"
        echo "the key to trusted!"
        exit
        ;;
    esac
done

if [ -z "$SERVERNAME" ] | [ -z "$SERVERPORT" ] | [ -z "$STORECACERTS" ] ; then
    echo "USAGE: `basename $0` -h hostname -p port -f Certfile"
openssl s_client -showcerts -connect $SERVERNAME:$SERVERPORT < /dev/null > $TMP1/cst-key.out
openssl x509 -outform DER < $TMP1/cst-key.out > $TMP1/cst-key.der
openssl x509 -inform der -in $TMP1/cst-key.der -out $TMP1/cst-key.pem
keytool -import -alias Selfsigned -keystore $STORECACERTS -file $TMP1/cst-key.pem
```

>> TDI and Javascript

- Combine LDAP Attributes to one value (Fullname)
- Set Database Fields to null
- set timezone
- Add Functions to profiles_functions.js
- Syntax in map_dbrepos_from_source.properties
  - z.B. displayname={functionsname}

>> Fullname / Displayname not set in LDAP

- AD:
  - only givenName and surname
  - often set to "surname, givenname"

```
function func_compute_CN(fieldname) {
    var givenName = work.getAttribute("givenName");
    var sn = work.getAttribute("sn");

    if(sn != null) {
        var result = givenName + ' ' + sn;
    }
    return result;
}
```

- Syntax in map_dbrepos_from_source.properties
  - displayName={func_compute_CN}

>> Set Timezone

- Timezone often not set in LDAP Systems
- when User edit their profile, the timezone is often set to "-12" (Default)
- profiles_functions.js:

```
function function_settimezone(fieldname){
    var timeZone = 'Europe/Amsterdam';
    result = timeZone;
    return result;
}
```

- Syntax in map_dbrepos_from_source.properties
  - timezone={function_settimezone}

>> reset Fields in empinst.employee

- Fields in POC often set all data (mobile, phone)
- reset field in profiles_functions.js

```
function function_setnull(fieldname){
    var result = '';
    return result;
}
```

- Syntax in map_dbrepos_from_source.properties
  - `mobile={function_setnull}`
- Caution: All values in this field will be deleted

>> Resources

>> Download the shown scripts

- You can download all scripts (and some more) WITHOUT WARRENTY and on your own risk:

  https://github.com/stoeps13/ibmcnxscripting
- OpenNTF Project since 21st november 2014
  - Administration Scripts for WebSphere
- There is much more:
  - memberSyncAllByEXID.py, inactivate Users
  - Create a printable version of Connections documentation
  - Database backup, Lastlogon
  - Change Monitoring Policy
  - and so on

OpenNTF.Org

>> Future

- create Powershell or Windows Batches
- scripts for basic troubleshooting
- add more error handling
- more documentation

You're invited to work with these scripts, or upload your own

- Discuss with me through
  - Skype: christophstoettner
  - Twitter: @stoeps
  - G+, Facebook, LinkedIn ...

Thanks for listening

Part of Imtech