Script it!

Basics to automate IBM WebSphere administration

Christoph Stöttner - a stoeps

Social Connections VI June 16 & 17, 2014 · Prague

TIMETOACT GROUP
SOFTWARE & CONSULTING

software, systems and services
FRITZ & MACZIOL
group

Belsoft applications

oxyonline

media partner: IDG

SOCIAL CONNECTIONS VI

JUNE 16 & 17, 2014 · PRAGUE

About Me

IBM CHAMPION

Christoph Stöttner

- IBM Software Consultant at Fritz & Macziol
- Specialized in the IBM Connections and IBM Domino Infrastructure
- Bavarian
- Linux and Scripting Lover, Blogger
- Speaker at:

ENERGIZING LIFE'S WORK Connect 2014 January 26-30 Orlando, Florida

Social Connections

NOTES & DOMINO ADMINCAMP

ICON UK the IBM user group

DNUG The Enterprise Collaboration Professionals

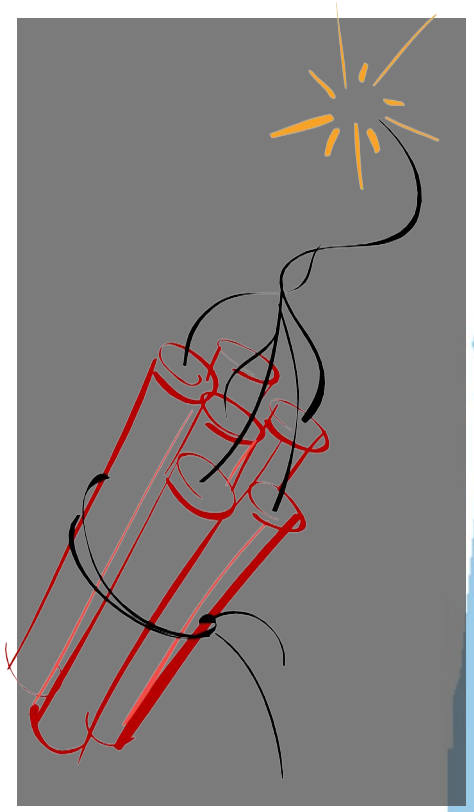
engage by blug

bccon BusinessConnect Conference

DANNOTES

Disclaimer

- With scripts
  - Shell / BASH / ZSH / KSH / SH
  - Jython / JACL
  - Powershell / Batch / VB
- You can…
  - save a lot of time!
  - **change tons of stuff in seconds!**



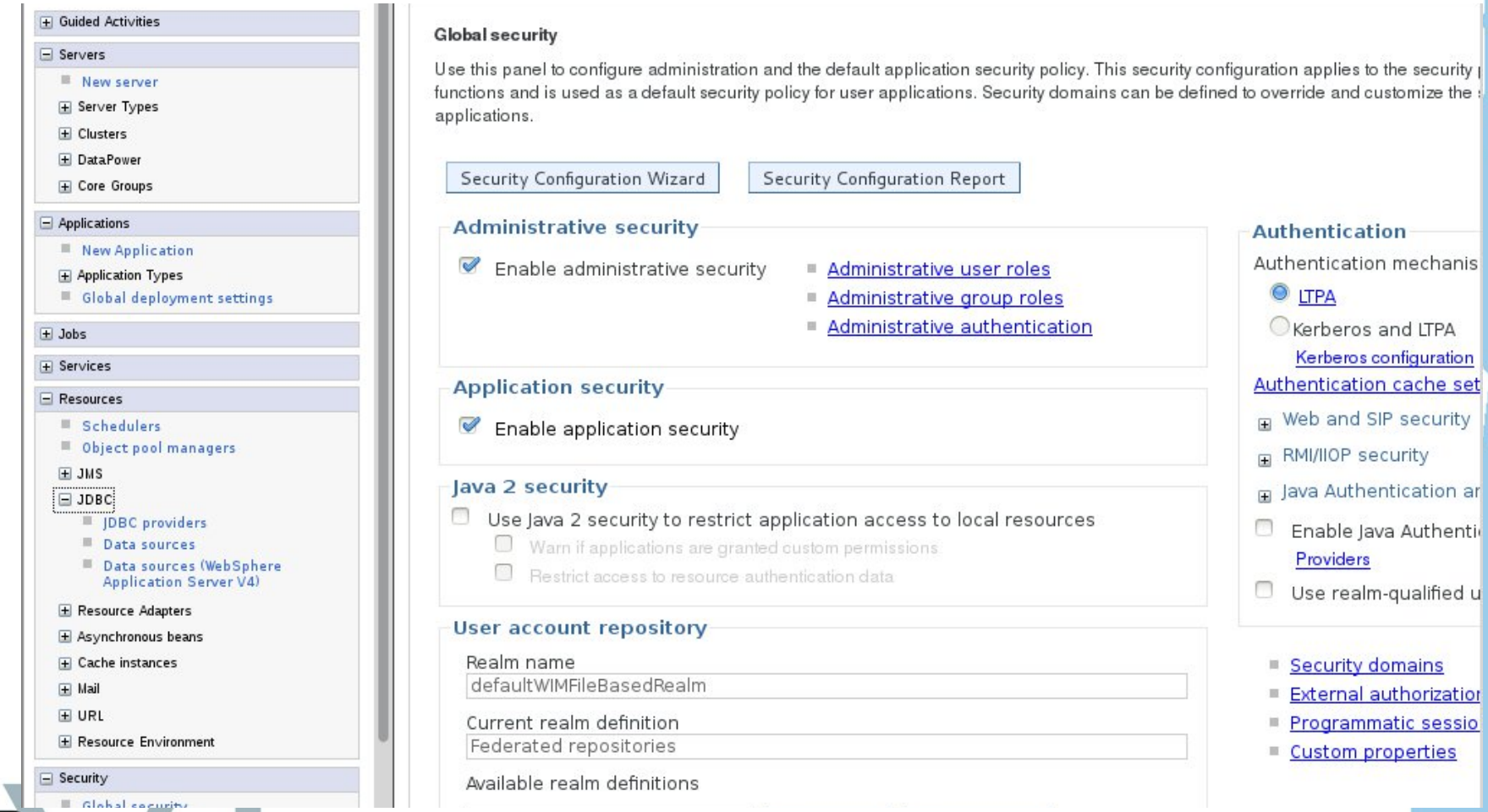
Use all scripts i show in this slides or you download from my repositories WITHOUT WARRANTY and on your own risk!

I'm Not A Developer

- but even as an Admin, i can read and write JYTHON
- Code is not beautiful, but it works
- Will provide you the basics to create scripts the next 20 minutes

WEBSPHERE BASICS

Integrated Solution Console



Integrated Solution Console (2)

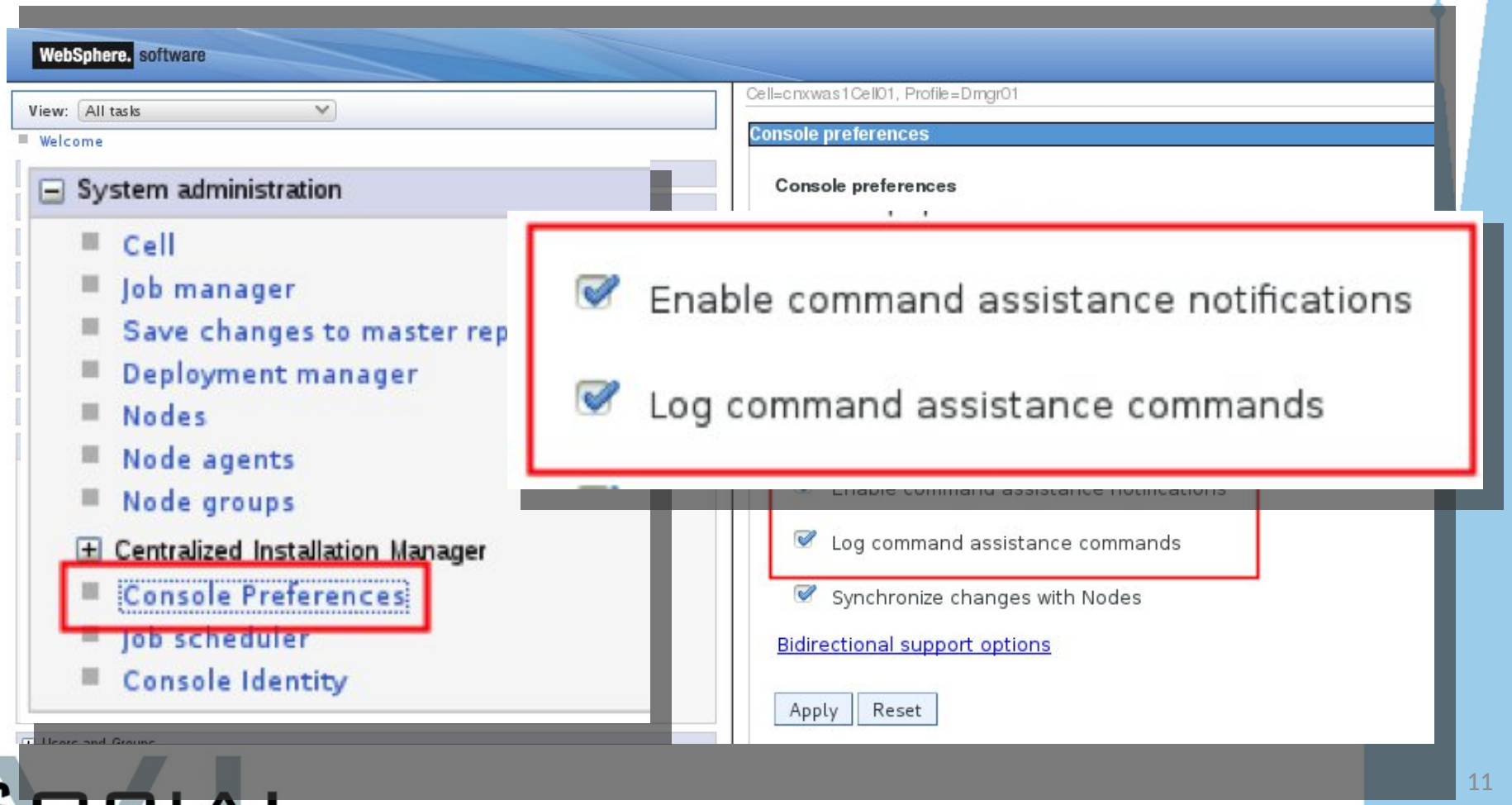
- Browserbased GUI for IBM WebSphere Application Server
- Mouse pointer runs miles during a Connections Installation
  - ~~Some~~ tasks are boring
  - Performance Tuning of DataSources
  - Setting Security Roles on Applications (Connections + FEB + CCM = 24 Apps)

wsadmin

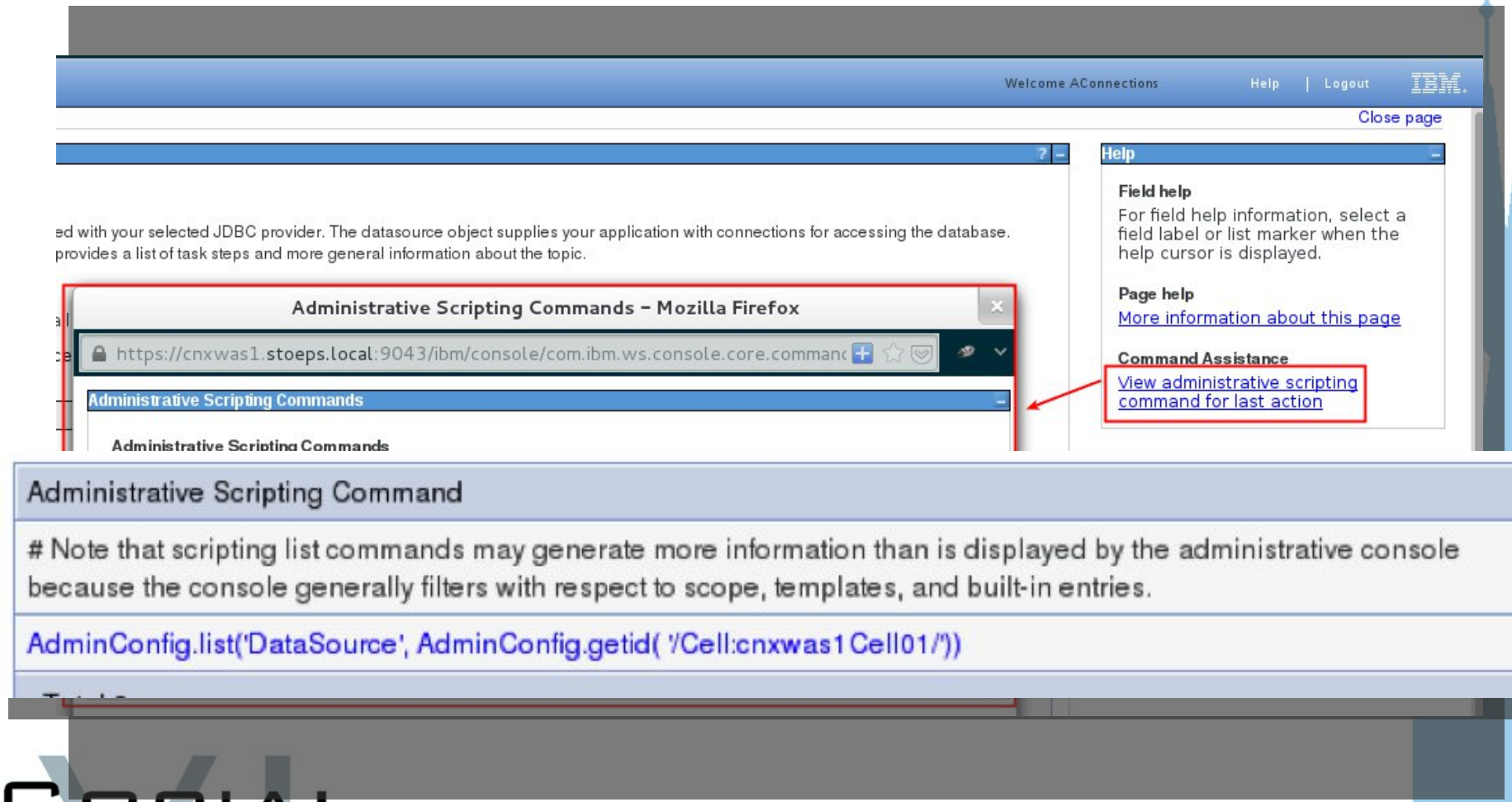
- seems often complicated at first look
- long case sensitiv commands
- within Linux no history to recall commands
- default language: JACL (depricated)
- Use a Cheatsheet with often used commands

WEBSPHERE SCRIPTING

Get Jython Commands



Command Assistance Notification



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/'))
```

Make Scripts Portable

- Command assistance contains hardcoded
  - cell
  - server
  - nodename
- Change them to port to other hosts!
- So fill these points is the biggest issue!
  - Solution: be patiented

wsadmin - Command Line

- execute wsadmin in Deployment Manager bin

```
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

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

wsadmin / jython / python

- Test your commands in the built-in shell

```
[root@cnxwas1 ~]# wsadmin
WASX7209I: Connected to process "dmgr" on node cnxwas1CellManager01 using SO
AP connector;  The type of process is: DeploymentManager
WASX7031I: For help, enter: "print Help.help()"
wsadmin>AdminControl.getCell()
'cnxwas1Cell01'
wsadmin>print 1+1
2
wsadmin>test = "My test string"
wsadmin>print test
My test string
wsadmin>
```

wsadmin – WebSphere Tasks

- AdminControl.getCell()
  - print the cellname of your environment
- AdminTask.listServers()
  - List all servers (nodeagents, dmgr, appserver, webservers)
- AdminTask.listServers('[-serverType APPLICATION_SERVER]')
  - List of all Enterprise Application Servers
- AdminApp.list()
  - List of all installed applications
- AdminConfig.getid('/DataSource: blogs/')
  - access DataSource ID BLOGS
- AdminConfig.save()
  - save configuration, without all changes are lost after closing wsadmin

wsadmin – Connections Tasks

- execfile("connectionsConfig.py")
- execfile("applicationAdmin.py")
- FilesPolicyService.add(title, size)
- CommunitiesService.moveCommunityToSubcommunity(comm_id_parent, comm_id)
- synchAllNodes()

JYTHON BASICS

Jython

- Version 2.5.2 included in wsadmin
  - Functions of Python 2.7 or 3.x will not work!
- Grouping of code (functions, if ...) without brackets
  - indent of 4 spaces
  - problems with tabs on Windows
    - replace tabs with 4 spaces (editor settings)

Jython

- Variables can be declared without type
  - String
    - variable1 = "This is a string"
  - Integer
    - variable2 = 1
  - Float
    - variable3 = 1.2
- Comments with #

Jython 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
```

Some more

- if

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

- for

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

PROGRAMMING EXAMPLES

Combine Commands

- Let's create the first script
- Update VersionStamp in lotusconnections-config.xml
- What would you do within wsadmin:
  - execfile("connectionsConfig.py")
  - LCConfigService.checkOutConfig('**D:\\temp**', '**cnxwas1Cell01**')
  - LCConfigService.updateConfig('versionStamp',"")
  - LCConfigService.checkInConfig('**D:\\temp**', '**cnxwas1Cell01**')
  - synchAllNodes()

Combine Commands (2)

- We have to use 2 variables for path and cellname

```
cellname = AdminControl.getCell()
path = raw_input('Please type in a temporary directoy path: ')

execfile('connectionsConfig.py')
LCConfigService.checkOutConfig( path, cellname )
LCConfigService.updateConfig( 'versionStamp', "" )
LCConfigService.checkInConfig( path, cellname )
synchAllNodes()
```

J2EE Backup

```
Role:  search-admin
Everyone?:  No
All authenticated?:  No
Mapped users:  AConnections|wasadmin
Mapped groups:  CNXAdmins
All authenticated in trusted realms?:  No
Mapped users access ids:  user:defaultWIMFileBasedRealm/CN=Admin Connect
ions,OU=users,O=stoeps|user:defaultWIMFileBasedRealm/uid=wasadmin,o=defa
ultWIMFileBasedRealm
Mapped groups access ids:  group:defaultWIMFileBasedRealm/CN=CNXAdmins
```

Create a backup of J2EE Roles

- Get a list of all installed applications
  - AdminApp.list()
    - String with \n after each app
    - split to get a jython list with splitline()
- Print information of J2EE Roles
  - AdminApp.view('BLOGS', "-MapRolesToUsers")
- create a file, open for writing
  - my_file = open(/tmp/doc1.txt, *'w'*)
- write information to file
  - myfile.write(' Text to write here')

Create a backup of J2EE Roles (2)

- Put it together

```
path = raw_input( "Please provide a path for your backup files: " )
appsList = AdminApp.list().splitlines()

for app in appsList:
    filename = path + "/" + app + ".txt"
    print "Backup of " + app + " security roles saved to " + filename
    my_file = open( filename, 'w' )
    my_file.write ( AdminApp.view( app, "-MapRolesToUsers" ) )
    my_file.flush
    my_file.close()
```

- More details? Have a look at …

COMMUNITY SCRIPTS

Community Scripts

- Download links in the end of this presentation
- Useful Scripts
- New version (github.com/stoeps13/ibmcnx2)
  - all scripts moved to a subfolder
  - using a properties-file to get save time
- Integrated menu

Installation

- Extract zip to DMGR/bin
- all files are in subfolder ibmcnx
  - rename ibmcnx_sample.properties to ibmcnx.properties
  - edit ibmcnx.properties
    - check j2ee.* (used for setting security roles)
    - db* (DB Host, User and Password)
    - Values within [Tuning] are copied from IBM Connections Performance Tuning Guide

Load DB2 JDBC Driver

- Linux
  - create wasuserscript.sh
    - export WAS_EXT_DIRS=$WAS_EXT_DIRS:/opt/IBM/JDBC
  - export WAS_USER_SCRIPT=wasuserscript.sh
    - before starting wsadmin!
- Windows
  - edit setupCmdLine.bat (dmgr\bin)
  - add jdbc path at WAS_EXT_DIRS

ibmcnx.properties

```
15  [Generic]
16  # Variables for Usermapping, add multiple Users/Groups with |
17  j2ee.cnxwasadmin = wasadmin
18  j2ee.cnxadmin = AConnections
19  j2ee.cnxmoderators = AConnections
20  j2ee.cnxmetrics = AConnections
21  j2eeconnmobile = Aconnections
22  # Variables for Groupmapping
23  j2ee.cnxadmingroup = cnxAdmins
24  j2ee.cnxmoderatorgroup = cnxAdmins
25  j2ee.cnxmetricsgroup = cnxAdmins
26  j2ee.cnxmobilegroup = cnxAdmins
27
28  [Database]
29  dbUser = lcuser
30  dbPassword = password
31  dbHost = cnxwin.stoeps.local
32  dbPort = 50000
33  dbName = PEOPLEDB
```

Access Properties

- Getting a value from properties

```
import ConfigParser
configParser = ConfigParser.ConfigParser()
configFilePath = r'ibmcnx/ibmcnx.properties'        # File with properties
configParser.read( configFilePath )                 # Open properties file
variable = configParser.get('Database','dbUser')    # get value of dbUser in part [Database]
```

Get A List Of All Servers

- Class to get a list of servers

```
import ibmcnx.appServer

WS1 = ibmcnx.appServer.WasServers()

for count in range(WS1.serverNum):
    jvm = WS1.jvm[count]
    cell = WS1.cell[count]
    node = WS1.node[count]
    servername = WS1.serverName[count]
    print cell + " " + node + " " + servername
```

```
cnxwinCell01 cnxwinCogNode03 nodeagent
cnxwinCell01 cnxwinCogNode03 cognos_server1
cnxwinCell01 cnxwinNode01 nodeagent
cnxwinCell01 cnxwinNode01 Cluster1_server1
cnxwinCell01 cnxwinNode01 Cluster2_server1
cnxwinCell01 cnxwinNode01 InfraCluster_server1
cnxwinCell01 cnxwinNode02 ConversionCluster_server1
cnxwinCell01 cnxwinNode02 nodeagent
cnxwinCell01 cnxwinNode02 ViewerCluster_server1
cnxwinCell01 cnxwinNode02 DocsCluster_server1
cnxwinCell01 cnxwinCellManager01 dmgr
```

ibmcnx/functions.py

- Collections of functions used within the scripts
- import ibmcnx.functions
- ibmcnx.functions.checkBackupPath(path)
  - checks if path available or create it
- ibmcnx.functions.getDSId(dbname)
  - returns DataSource ID of dbname
  - required to change DataSource Settings

DEMO

RESSOURCES

Documentation

- Community Scripts
  - http://scripting101.org
- Blogs
  - http://www.stoeps.de
  - http://kbild.ch
  - http://www.socialshazza.com

Links

- Learn Jython / Python
  - 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/

Download Scripts

- http://github.com/stoeps13/ibmcnx2
- http://github.com/stoeps13/ibmcnxscripting
- http://openntf.org/main.nsf/project.xsp?r=project/Administration%20Scripts%20for%20WebSphere

THANK YOU

christophstoettner
y www.stoeps.de
y scripting101.org
github.com/stoeps13
christoph.stoettner@stoeps.de
ab twitter.com/stoeps
c8j facebook.com/christoph.stoettner
l www.stoeps.de/+
slideshare.net/ChristophStoettner
linkedin.com/pub/christoph-stoettner/13/30a/2b3/
xing.com/profile/Christoph_Stoettner
about.me/stoeps

Christoph Stöttner
IBM Software Consultant
Fritz & Macziol GmbH
www.fum.de
cstoettner@fum.de