



# Domain Creation

DOAG 2013

Andreas Chatziantoniou - Foxglove-IT BV

Thorsten Wussow – Slix GmbH



Bio –

Andreas Chatziantoniou



- Freelance Oracle Fusion Middleware Consultant
- 15 Jahre Oracle Erfahrung/25 Jahre IT (Unix/C)
- Oracle ACE
- [andreas@foxglove-it.nl](mailto:andreas@foxglove-it.nl)



# Bio – Thorsten Wussow



- Mitarbeiter Slix GmbH seit 1998
- 18 Jahre IT, davon 15 Jahre mit Oracle
- Fokus Oracle WebCenter
- [thorsten@slix.de](mailto:thorsten@slix.de)



# Agenda



- Einleitung
- Oracle Werkzeuge
- Unsere Lösung
  - Excel/python
  - Updates/Propagation
- Demo



# Einleitung



- Raise-of-hands
  - Wer baut Domains?
  - Wer baut regelmäßig (viele) Domains?
  - FMW/Fusion Apps/Custom?
  - Lifecycle?





# Voraussetzungen



- Um WLS und/oder FMW Domains zu bauen müssen eine Reihe von Voraussetzungen erfüllt sein
  - OS und System (Plattform)
  - Anlegen eines Installations Users
  - Disk Space (temp. und Image)
  - Administrator (Windows)
  - Node Manager als Windows Service

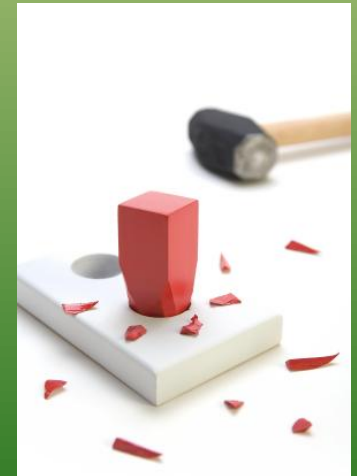




# Voraussetzungen



- Alle Voraussetzungen sollten erfüllt sein
- Wird oft durch verschiedene Teams geliefert
  - Synchronisierung führt zu Problemen (Zeit, Qualität)
- OS Image vs. selber anpassen?
  - Regelmäßig neues Image bauen
  - Rechte nicht vorhanden in großen Organisationen





# Voraussetzungen



- Software Verteilung
  - Die Software muss auf die Rechner gespielt werden
  - Verschiedene Möglichkeiten
- Entscheidend ist wie oft die Software benötigt wird, bzw. wie oft eine Domain erstellt wird



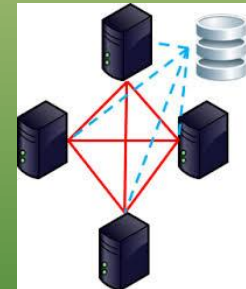
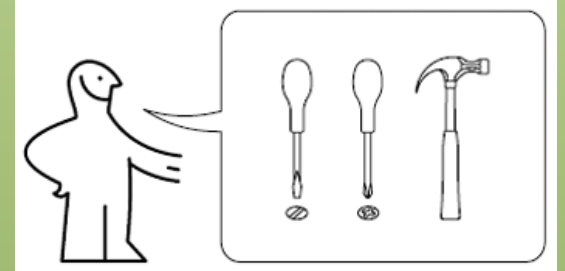




# Voraussetzungen



- Manuelle Installation
  - Einfach
  - Sehr fehleranfällig
  - Selten reproduzierbar
- Shared disks
  - Teilen von Resources kann zu Fehlern führen
  - Logfiles, Zertifikate





# Voraussetzungen



- Package/Cloning
  - Lösung mit Skripten
  - CloudControl Provisioning

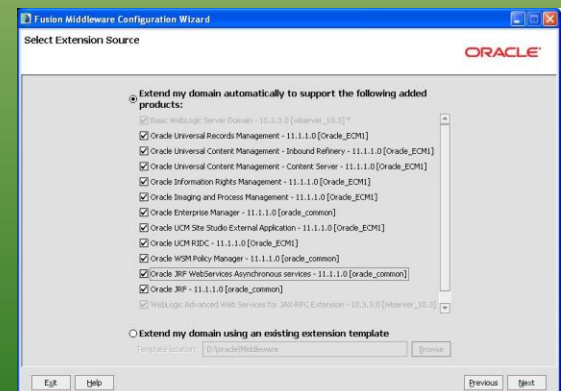




# Oracle Werkzeuge



- `config.sh / config.cmd`
  - Interaktives Aufbau von Domains
  - Für komplexe Umgebungen ggf. mehrmals nacheinander benutzen
    - Erst WLS, dann evt. FMW Komponenten
  - Cluster und Datasources sind schwierig einzugeben
    - Multi datasources
    - Active Grid Link





# config.sh



- Kernproblem des config.sh ist die Interaktivität
  - Wiederholen von Konfiguration führt schnell zu verschiedenen Endresultaten
  - Bestimmte Parameter lassen sich nicht einstellen
    - Nodemanager
    - SSL Keystores





# Template Builder



- Der Template Builder ist ein geeignetes Werkzeug um komplexere Domains aufzubauen
  - Vorteile
    - Wiederholbarkeit, geeignet für komplexe Domains
  - Nachteil:
    - Lifecycle eines Templates (Patches, Parameteränderungen)
    - Unterschiedliche Umgebungen z.B. DEV 2 Node Cluster, PRD 4 Node Cluster



# WebLogic Scripting Tool



- WLST bietet eine maximale Flexibilität
  - Konfiguration auf dem Level von MBeans
  - „Programmiersprache“
  - Sehr komplexe Domains können gebaut werden
  - Auslesen von existierenden Werten

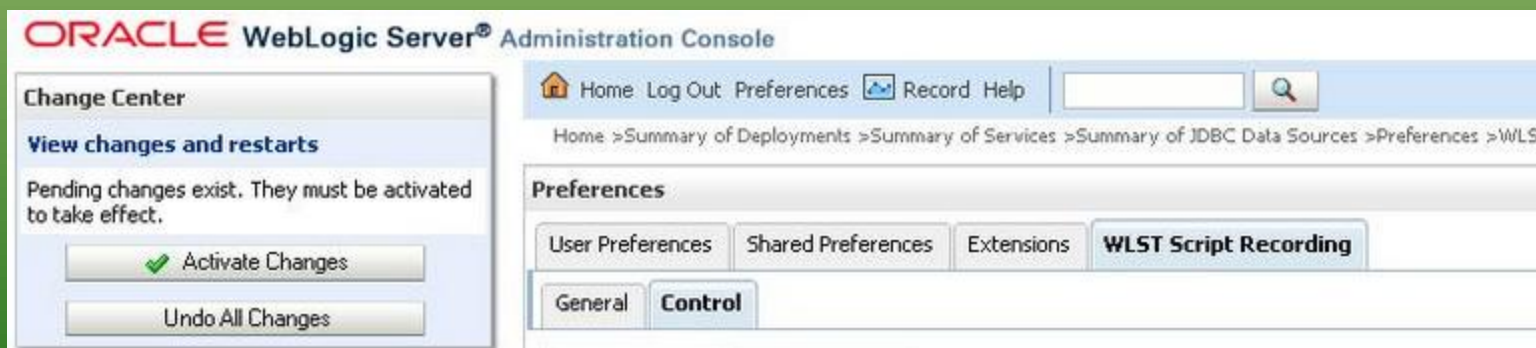
```
if serverState != "SHUTDOWN":
    cd("/ServerRuntimes/" + name.getName() + "/ThreadPoolRuntime/ThreadPoolRuntime")
    tput=cmo.getThroughput()
    cd("/ServerRuntimes/" + name.getName() + "/JVMRuntime/" + name.getName())
    heapSize=cmo.getHeapSizeCurrent()
    print name.getName()
    print " Throughput:",tput
    print " HeapSize:",heapSize
    print "\n"
```



# WebLogic Scripting Tool



- WLST hat auch einige Nachteile
  - Hoher Lernaufwand
  - Trial-and-error/keine IDE
  - „Aufnehmen“ in der Admin Console ist möglich
    - „Abspielen“ hat Quirks, ab und zu fehlen Werte





# Unsere Lösung



- Create Domain2 – CreDo2
  - Entstanden auf der Basis von verschiedenen Vorläufern
- Excel – mit Properties der Domain
  - Übersichtlich, einfach um komplexe Multi-Node Domains aufzubauen
  - Erzeugt Properties Datei
- Python Skript um WLST zu erzeugen
  - Wird auf dem Admin Server Node ausgeführt
  - Kopieren oder nmEnroll auf Managed Servers





# CreDo2



	A	B	C
1	<b>Environment</b>		
2	<b>Description</b>	<b>Property</b>	<b>Values</b>
3	PropertiesFilePath	NA	C:\
4	Createdate	NA	15-Nov-13
5	Author	NA	Andreas Chatziantoniou
6	PropertiesFilename	NA	wls
7	Version	domain_version	0.0.1
8	Domain Name	domain_name	doagdomain
9	Description	domain_description	Testdomain DOAG
10	Java Home	java_home	/extra/oracle/product/jrockit
11	Java Vendor	java_vendor	Oracle
12	Middleware Home	mwHome	/opt/app/oracle/ofm
13	Domain Home	domain_base	/opt/app/oracle/domains
14	Domain Logbase	domain_logbase	/var/log/
15	Domain Admin	domain_admin	weblogic
16	Password	domain_password	doag123
17	Create the Domain	createdomain	true
18			
19			
20			
21			
22			
23			
24			

Do It



# CreDo2



	A	B	C	D	E
1	Description	Property	Values	Example	Range
2	Name	name	DOAG_TEST_ADMIN		
3	Machine	machine	testmachine		
4	Listener Address	listenaddress	localhost		
5	Listen Port	listenport	7011		
6	Tunneling Enabled	tunneling	true		
7					
8	SSL Listen Port Enabled		TRUE_FALSE	false	
9	SSL Listen Port		PORT_OF_THE_LISTENING_ADDRESS	7009	
10	Client Cert Proxy Enabled		TRUE_FALSE	false	
11	Java Compiler		JAVA_COMPILER	javac	
12	Diagnostic Volume		LEVEL	low	off,low,medium,high
13	Virtual Machine Name		NAME	MS_SRV01	
14	WebLogic Plug-in Enabled		TRUE_FALSE	false	
15	Prepend to Classpath		PATH		
16	Append to Classpath		PATH		
17	Extra RMI Compiler Options		OPTIONS		
18	Extra EJB Compiler Options		OPTIONS		
19	External Listen Address		DNS_NAME_OF_THE_EXTERNAL_LISTENING_ADDRESS	localhost	
20	Local Administration Port Override		PORT_OF_THE_LISTENING_ADDRESS	9002	
21	Startup Mode		MODE	Running	Running,Administration,Standby
22	JDBC LLR Table Name		LOGGING_LAST_RESOURCE_DB_TABLE		
23	RMI JDBC Security		SECURITY_PROTOCOL	Compatibility	Compatibility,Secure
24					



# CreDo2



```
37 # Set the path separator
38 operating_system = 'Linux';
39 pathSeparator = '/';
40
41 if operating_system == 'Linux':
42     pathSeparator = '/';
43 elif operating_system == 'Windows':
44     pathSeparator = '\\';
45 else:
46     pathSeparator = '/';
47
48 # Set the path in which the domain should be created
49 domainLocation = '';
50
51 # Set the path where the JVM is located
52 jvmLocation = '';
53
54 # Name the template for the domain
55 domainTemplate = '';
56
57 # The initialize function will set the basic settings or the domain creation
58 def initialize():
59     global domainLocation;
60     global jvmLocation;
61     global domainTemplate;
62     .....
63     print '-----';
64     print '--- initialize                ---';
65     print '-----';
66
67     # Load the properties
68     # The properties are in the file specified when calling this python script
69     loadProperties("/jvm/wlscreate.properties");
70     .....
71     # Inside the properties file the properties domainHome and domainName are present
72     # These will be used to create the domainLocation
73     print 'Prepare the domainLocation ...'
74     domainLocation = domainHome + pathSeparator + domainName;
75     print 'The domainLocation is: ' + domainLocation;
76
```



# CreDo2



- Metadaten
  - Wann wurde ein Properties File erzeugt?
  - Wer hat das Properties File erzeugt?
  - Version
  - Änderungen



# CreDo2



- Updates
  - Flag zeigt an ob eine Domain erzeugt wird oder ein update stattfindet
- Propagation
  - Anpassen der Properties, z.B. Prefix (TST,PRD)
  - Ports und Pfade sollten nicht verändert werden



# Demo



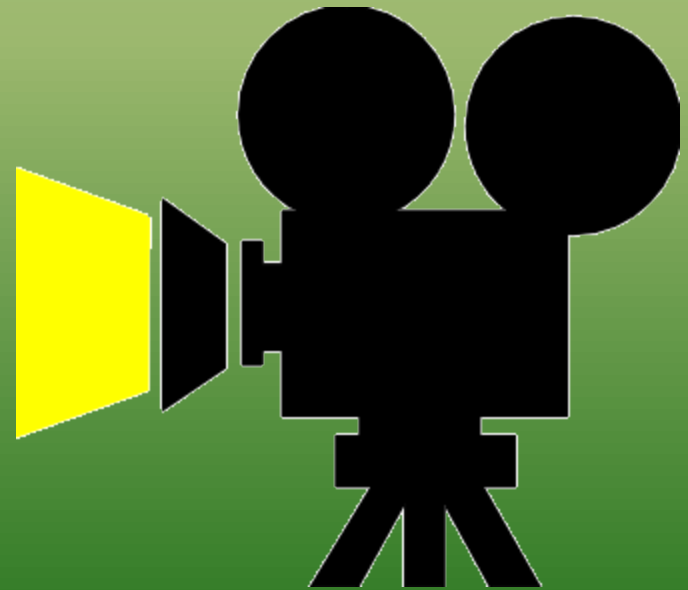
- Demo Teil 1
  - Erstellen einer WLS Domain
  - Properties
  - Ausführen
- Demo Teil 2
  - Erstellen einer WebCenter Domain
  - Cluster
  - Datasources für Metadata Repository



# Demo



- Demo 1: <http://youtu.be/TRtsO26XVds>
- Demo 2: <http://youtu.be/...>





# CreDo2 bei Ihnen



- Wollen Sie CreDo2 einsetzen?
- Nehmen Sie einfach Kontakt mit uns auf!

[thorsten@slix.de](mailto:thorsten@slix.de)

[andreas@foxglove-it.nl](mailto:andreas@foxglove-it.nl)



