

Skriptbasierte Administration des WebLogic Servers mittels WLST 12.1.2

Thomas Robert
Oracle Deutschland B.V. & Co. KG
Hamburg

Schlüsselworte

WebLogic, WLST, WebLogic Scripting Tool, Administration, OHS, Coherence

Einleitung

Neben der Browser basierten Administrationskonsole stellt das WebLogic Scripting Tool (WLST) *das* zentrale Werkzeug für die Erstellung und Verwaltung von WebLogic Umgebungen dar. Im Gegensatz zur interaktiven Konsole ermöglicht WLST die Erstellung von batch-orientierten Skripten zur automatisierten Lösung nahezu aller Aufgaben, die bei der Administration einer Domäne anfallen.

Grundlagen

WLST 12.1.2 basiert auf der Skriptsprache Jython 2.2.1. Jython führt Python Sourcecode innerhalb einer JVM aus. WLST stellt eine umfangreiche Bibliothek von Kommandos bereit, um automatisiert – d.h. über entsprechende Skripte – administrative Tätigkeiten wahrzunehmen. Neben der Erstellung und Erweiterung von WebLogic Domänen, dem Starten und Stoppen von Komponenten und dem Zugriff auf alle im WebLogic Server definierten Managed Beans ist seit der Version 12 auch die Administration von Coherence Clustern und des Oracle HTTP Servers immer weiter ausgebaut worden. WLST Skripte können mittels der Jython Sprachsyntax beliebig erweitert werden, um auch komplexe Aufgaben automatisiert durchzuführen.

Überblick

WLST Befehle können auf drei Arten abgesetzt werden:

- interaktiv über die WLST Shell,
- innerhalb eines komplexen Skriptes,
- in Java Code eingebettet.

Außerdem kann WLST im „offline“ oder „online“ Modus mit seiner Fusion Middleware Umgebung interagieren. Allerdings sind nicht alle Befehle in jedem Modus erlaubt. So können im „offline“ Modus vor allem Aktivitäten des Configuration Wizards (config.sh / config.cmd) wie z.B. das Erstellen oder Modifizieren einer WebLogic Domäne automatisiert werden. Der „online“ Modus ermöglicht dagegen alle administrativen Tätigkeiten, die in der WebLogic Konsole interaktiv durchgeführt werden können. Hierfür ist es notwendig, sich zunächst mit dem Admin Server oder einem Managed Server zu verbinden:

```
wls:/offline> connect('weblogic', 'welcome1', 'localhost:7001')  
wls:/base_domain/serverConfig>
```

Abb. 1: Beispiel für eine Anmeldung an die WebLogic Domäne „base_domain“

Nach der Anmeldung stehen alle MBeans der Domäne in einer hierarchischen Navigationsstruktur zur Verfügung. Dabei organisiert der WebLogic Server seine MBeans in mehrere Bäume:

serverConfig: dieser Baum enthält Konfigurationsinformationen des Managed Servers, gegen den man aktuell verbunden ist. Dies ist der Standardbaum nach Anmeldung an einen Server;

domainConfig: enthält Konfigurationsinformationen der gesamten Domäne;

edit: ein editierbarer Baum der Domänen Konfiguration;

serverRuntime: enthält die Runtime MBeans des Servers, gegen den man angemeldet ist;

domainRuntime: enthält die Runtime-MBeans der gesamten Domäne;

jndi: der JNDI Baum des Servers, gegen den man sich verbunden hat;

custom: Baum der custom MBeans im Server, gegen den man angemeldet ist;

domainCustom: Baum der custom MBeans in der gesamten Domäne.

Der Wechsel von einem Baum zu einem Anderen erfolgt durch Eingabe des entsprechenden Namens, z.B.:

```
wls:/base_domain/serverConfig> domainConfig()
wls:/base_domain/domainConfig> serverRuntime()
wls:/base_domain/serverRuntime> jndi()
wls:/base_domain/jndi>
```

Abb.2: Wechseln der Navigationsbäume, z.B. serverConfig, domainConfig, serverRuntime und jndi

Die WLST Variable „cmo“ enthält zu jeder Zeit das aktuelle MBean-Objekt, auf dem dann entsprechende get/set/invoke Befehle ausgeführt werden können:

```
wls:/base_domain/serverConfig> domainConfig()
wls:/.../domainConfig> cd ('ServerLifeCycleRuntimes/ManagedServer1')
wls:/.../ServerLifeCycleRuntimes/ManagedServer1> cmo.getState()
'SHUTDOWN'
wls:/.../ServerLifeCycleRuntimes/ManagedServer1> ls()
...
-r--  Name                ManagedServer1
-r--  NodeManagerRestartCount  0
-r--  State                SHUTDOWN
-r--  Type                ServerLifeCycleRuntime
...
```

Abb.3: Beispiel für „cmo“ = Current Management Object

Aufruf von WLST

Der Aufruf der WLST Shell, um interaktive Befehle abzusetzen oder eine Batch-Datei zu starten, erfolgt am Besten über das Command-File wlst.sh (Unix / Linux) bzw. wlst.cmd (Windows), da dort alle notwendigen (Klassen-)Pfade korrekt gesetzt werden. In einer Fusion Middleware Umgebung, in der neben dem WebLogic Server noch andere Produkte installiert sind, gibt es wlst Command Files in unterschiedlichen Unterverzeichnissen des Oracle_Homes:

Im \$ORACLE_HOME/wlserver/common/bin befindet sich das „basis“-Command-File zum Starten von WLST. Es enthält alle notwendigen Pfade, um mit den WebLogic Servern zu interagieren.

Im \$ORACLE_HOME/oracle_common/common/bin befindet sich das Command-File, in dem zusätzliche Klassenpfade z.B. für die SOA- oder WebCenter Suite oder andere Komponenten wie JRF gesetzt werden. Hier werden auch weitere, spezialisierte WLST Funktionen definiert. In der Regel sollte die WLST Shell aus diesem Verzeichnis heraus gestartet werden.

Im `$ORACLE_HOME/ohs/common/bin` befindet sich ein Command File, das Erweiterungen für die Verwaltung des Oracle HTTP Servers beinhaltet.

Anmeldung an eine Domäne

Um WLST im „online“ Mode zu benutzen, muss man sich mit Username und Passwort eines Benutzers, der in der Security Realm der Domäne definiert ist, anmelden. Weiter muss der Hostname und Port des Servers angegeben werden. Man kann sich auch an einen Managed Server anmelden. Dann stehen jedoch nur die MBeans des jeweiligen Managed Servers zur Verfügung. Meldet man sich dagegen beim Admin Server an, hat man zusätzlich Zugriff auf alle MBeans der Domäne, d.h. auch auf die MBeans sämtlicher in der Domäne konfigurierten Managed Server.

Mit dem Befehl `storeUserConfig()` werden die Anmeldeinformationen nach erfolgter Anmeldung (verschlüsselt) in einer Properties-Datei abgelegt. So muss man bei zukünftigen Anmeldungen die Credentials nicht mehr im Klartext angeben.

WLST Online

Im „online“ Mode ermöglicht WLST innerhalb einer Domäne u.a.¹:

- die Überwachung von Managed Servern
- die Änderung von Konfigurationen, wie z.B. Erstellen oder Löschen von JMS Diensten, Anpassung von Datasource Konfigurationen, etc.
- Starten, Stoppen, Erstellen, Löschen, Clonen von Clustern und Managed Servern
- Starten, Stoppen, Erstellen, Löschen, Clonen von Coherence Clustern und Coherence Servern

```
wls:/base_domain/serverConfig> edit()
wls:/base_domain/edit !> startEdit()
wls:/base_domain/edit !> cmo.createServer('CoherenceServer1')
wls:/base_domain/edit !> cd('/Servers/CoherenceServer1')
wls:/base_domain/edit/Servers/CoherenceServer1 !> cmo.setListenPort(9001)
wls:/base_domain/edit/Servers/CoherenceServer1 !> cd('/')
wls:/.../edit !> cmo.createCoherenceClusterSystemResource('MyCoherenceCluster')
wls:/.../edit !> cd('/Servers/CoherenceServer1')
wls:/.../edit/Servers/CoherenceServer1 !>
cmo.setCoherenceClusterSystemResource(getMBean('/CoherenceClusterSystemResources/My
CoherenceCluster'))
wls:/base_domain/edit/Servers/CoherenceServer1 !>
cd('/CoherenceClusterSystemResources/MyCoherenceCluster')
wls:/base_domain/edit/CoherenceClusterSystemResources/MyCoherenceCluster !>
cmo.addTarget(getMBean('/Servers/CoherenceServer1'))
wls:/.../edit/CoherenceClusterSystemResources/MyCoherenceCluster !> activate()
```

Abb.4: Beispiel für das Anlegen eines Coherence Servers

WLST Offline

Der „offline“ Modus² ermöglicht die Anpassung und Erweiterung von Domänenkonfigurationen durch Templates. Außerdem kann eine Domäne z.B. in einer neuen Umgebung komplett, mit allen Artefakten, von Grund auf neu erstellt werden. Der „offline“ Modus darf nur gegen eine nicht gestartete Domäne verwendet werden, da Konfigurationsänderungen ansonsten von der laufenden Admin Server Instanz wieder überschrieben werden würden. Der Vorteil des „offline“ Modus ist, dass

¹ Siehe [WebLogic Scripting Tool Command Reference, ch. 2.2](#)

² Siehe [WebLogic Scripting Tool Command Reference, ch. 2.3](#)

er Änderungen an einer gestoppten Domäne erlaubt. Somit ist sichergestellt, dass während dieser Konfigurationsänderungen keine anderen Aktivitäten passieren.

Klonen von Domänen

Es gibt verschiedene Möglichkeiten, eine bereits vorhandene Domäne in ein neues Domänenverzeichnis oder auf einen anderen Rechner zu kopieren. Voraussetzung ist, dass die WebLogic Software in einer kompatiblen Version auf dem Zielsystem bereits installiert ist:

Das „pack“ und „unpack“ Command File packt eine Domänenkonfiguration auf einer Umgebung mit allen Artefakten ein und auf dem Zielsystem wieder in die neue Umgebung aus. Hierbei handelt es sich nicht um WLST Kommandos.

Mit dem WLST Befehl writeTemplate() wird aus der Domäne, gegen die man verbunden ist, ein Template File erzeugt. Dieses Template kann dann benutzt werden, um eine Domäne in einer neuen Umgebung identisch zu erzeugen.

Als dritte Variante bietet WLST den Befehl configToScript(). Dieser erzeugt ein weiteres WLST Skript sowie ein Properties File, dessen Parameter für die neue Umgebung angepasst werden können. Diese Variante stellt die flexibelste Form dar, um eine bestehende Domäne zu kopieren.

WLST für nicht-WebLogic Komponenten

Mit der Integration verschiedenster Oracle Fusion Middleware Produkte, wie z.B. SOA Suite, WebCenter Suite, Identity Management Suite, Oracle HTTP Server in den WebLogic Server, werden auch die verfügbaren WLST Befehle entsprechend erweitert. So können dann nicht nur Standard WebLogic Anwendungen und Libraries, sondern auch SOA Composites via WLST deployt werden und WebCenter Artefakte und SSO- oder Authentication Provider des Oracle Access Managers via WLST verwaltet werden.

Darüber hinaus stellt WLST in dieser Version eine Reihe neuer Befehle für die Administration eines Oracle HTTP Servers zur Verfügung, mit denen sich z.B. OHS Instanzen anlegen oder löschen, starten, restarten oder stoppen lassen. Welche Befehle es gibt, hängt davon ab, ob der OHS in einem gemeinsamen Verzeichnis mit dem WLS, oder als sog. „Standalone“ Domäne installiert ist. (Der OHS lässt sich zusammen mit dem WLS nur in ein gemeinsames Oracle_Home der sog. „Fusion Middleware 12c Infrastructure“ installieren. Diese muss von der Oracle eDelivery Webseite heruntergeladen werden. In ein aus dem wls_121200.jar erzeugten Oracle_Home kann der OHS nicht installiert werden.)

Wird der OHS als standalone Domäne installiert, dann muss die OHS Instanz über den Konfigurations-Wizard (config.sh) angelegt werden. Hierfür gibt es dann keine WLST Unterstützung. In dieser Umgebung stehen nur WLST Befehle zur Verfügung, die über den NodeManager an den OHS geschickt werden können (z.B. nmStart, nmStop).

```
wls:/base_domain/serverConfig> nmStart(serverName='ohs1', serverType='OHS')
wls:/base_domain/serverConfig> nmSoftRestart(serverName='ohs1', serverType='OHS')
wls:/base_domain/serverConfig> nmKill(serverName='ohs1', serverType='OHS')
```

Abb.5: Starten, Stoppen bzw. Restart der OHS via Nodemanager

Ist der OHS in einem WLS Oracle_Home installiert, müssen die WLS Domänen zunächst um das HTTP Server Template erweitert werden, damit die entsprechenden MBeans zur Verfügung stehen.

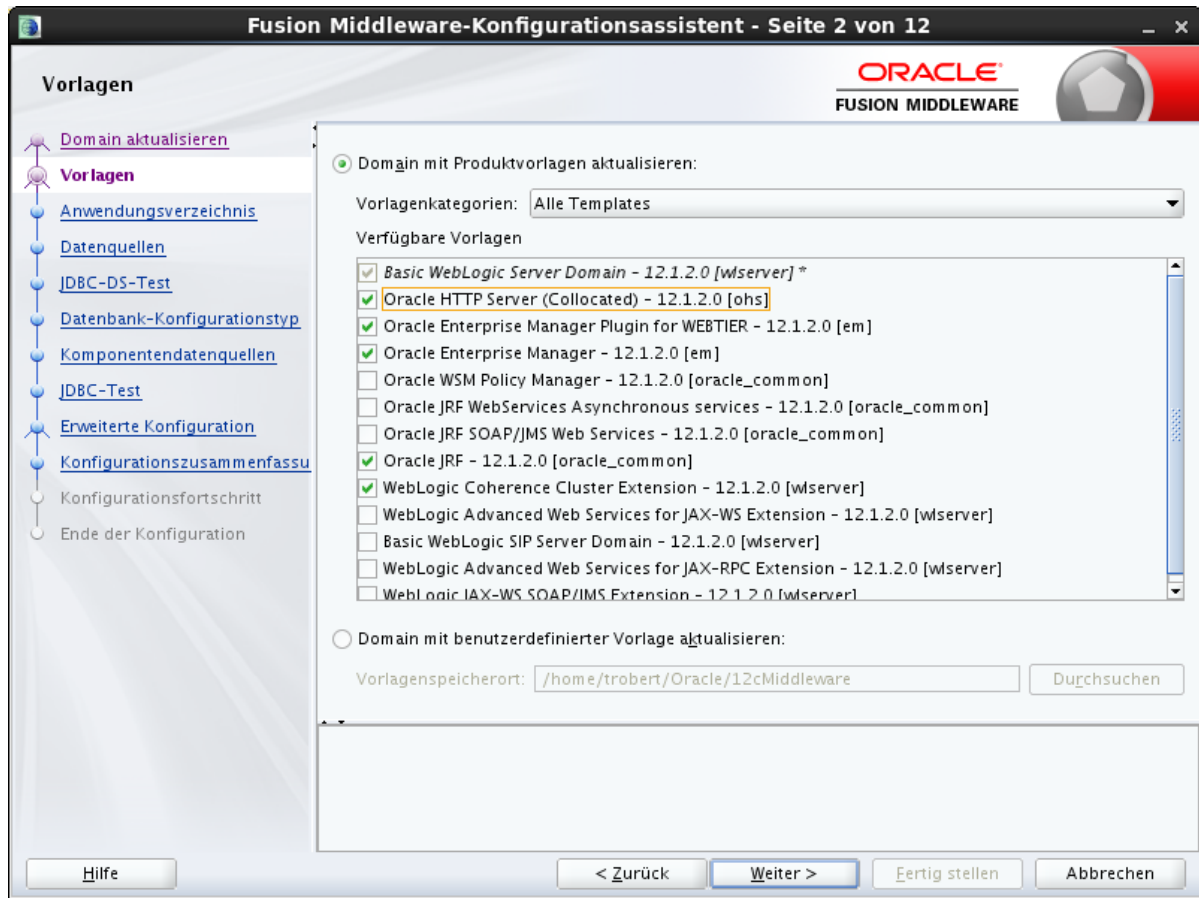


Abb.6:Erweitern der Domäne um Oracle HTTP Server Templates

Das HTTP Server Template setzt voraus, dass die benötigten Datenbankobjekte mit RCU angelegt wurden.

Dann gibt es zum Beispiel den Befehl `createOHSInstance()`, mit dem eine neue OHS Instanz erzeugt werden kann. Dieser Befehl steht allerdings nur zur Verfügung, wenn das `wlst` Command-File aus dem `ORACLE_HOME/ohs/common/bin` Verzeichnis aufgerufen wurde. In dieser Konfiguration kann der OHS auch direkt über den AdminServer gestartet oder gestoppt werden:

```
wls:/base_domain/serverConfig> start('ohs1')
wls:/base_domain/serverConfig> softRestart('ohs1')
wls:/base_domain/serverConfig> shutdown('ohs1')
```

Abb.7:Starten, Stoppen bzw. Restart der OHS

Außerdem stehen eine Reihe weiterer Befehle zur Verfügung, um die SSL Konfiguration des OHS via WLST durchzuführen.

Zusammenfassung

Mit dem WebLogic Scripting Tool bietet Oracle eine mächtige Skriptsprache um nahezu die komplette Fusion Middleware Umgebung zu administrieren. Neben dem Anlegen und Kopieren von Domänen lassen sich alle administrativen Tätigkeiten, wie das Erzeugen und Umkonfigurieren von Artefakten, das Deployen von Anwendungen oder ihrer Komponenten, sowie das Starten, Stoppen und die Überwachung aller Fusion Middleware Produkte automatisieren und nachvollziehbar durchführen.

Kontaktadresse:

Thomas Robert
Oracle Deutschland B.V. & Co. KG
Kühnehöfe 5
D-22761 Hamburg
Telefon: +49 (0) 40-890 91 188
Fax: +49 (0) 40-890 91 188
E-Mail Thomas.Robert@oracle.com
Internet: www.oracle.com/de

Michael Bräuer
Oracle Deutschland B.V. & Co. KG
Schiffbauergasse 14
D-14467 Potsdam
Telefon: +49 (0) 331-2007 306
Fax: +49 (0) 331-2007 561
E-Mail Michael.Braeuer@oracle.com
Internet: www.oracle.com/de