

# Anwendungsmigration von OC4J nach WebLogic

**Andreas Wussler**  
PROMATIS software GmbH  
Ettlingen

**Marc Brenkmann**  
SüdLeasing GmbH  
Stuttgart

**Dr. Joachim Reising**  
PROMATIS software GmbH  
Ettlingen

## Schlüsselworte

Oracle Fusion Middleware 11g, WebLogic Server

## Einleitung

Bereits seit Ende 2011 steht der WebLogic-Server in Version 12c als Nachfolger des nun schon in die Jahre gekommenen OC4J-Containers zur Verfügung. Für Entwickler und Middleware-Administratoren stellt sich nun die Frage, was alles beachtet werden muss, wenn bestehende OC4J-Anwendungen zukünftig auf dem WebLogic-Server betrieben werden sollen? Klar ist, dass die meist veralteten Komponenten und Bibliotheken nicht mehr unterstützt werden und somit die "ear"-, "war"- bzw. "jar"-Dateien nicht einfach im WebLogic-Server deployt werden können. Sehr wichtig ist in diesem Zusammenhang aber auch der Dialog zwischen den Entwicklern und Middleware-Administratoren, denn auch für diese ist die neue Umgebung eine Herausforderung. Vor allem was die Konfiguration des WebLogic-Servers in einer hochverfügbaren Umgebung und das Vorgehen beim Deployment betrifft, müssen die von der Entwicklung benötigten Vorgaben und Einstellungen in der der Middle-Tier-Umgebung zusammen erarbeitet und umgesetzt werden – auch hinsichtlich Sicherheitsaspekten, also Authentifizierung, Autorisierung sowie die möglichen Zugriffswege.

## Aufbau der Infrastruktur

Durch das Ende des Oracle Application Servers mit Version 10g galt bzw. gilt es, beim langjährigen Oracle Kunden SüdLeasing eine ganze Reihe von 10er-Komponenten auf Fusion Middleware zu migrieren. Nachdem die neue Infrastruktur (OID, OAM, Reports, Discoverer, WebServer und WebCache) schon nahezu vollständig nach 11g umgestellt ist, müssen nun auch die Java-Anwendungen vom OC4J-Container nach WebLogic umgezogen werden. Als Basis für eine der zentralen Applikationen wurde hierfür die Version 12c des WebLogic-Servers gewählt. Neben der Installation der WebLogic-Server-Software selbst, benötigt man zusätzlich einen passenden WebServer (Fusion Middleware Web-tier 11g) sowie eine zugehörige WebGate-Installation. Die notwendige Konfiguration des WebLogic-„Security Realms“ im Hinblick auf die Single Sign-On-Authentifizierung und Autorisierung mit dem Oracle Access Manager erfordert zunächst das Einrichten zusätzlicher „Provider“ und „Identity Asserter“. Diese sind nötig, um generell die Anmeldung mittels Single Sign-On an dieser WebLogic-Instanz und den darin bereitgestellten Anwendungen zu ermöglichen. Für die anstehenden Deployments muss dann mit den Entwicklern abgeklärt werden, welche Sicherheitseinstellungen, sowohl auf der Anwendungs- als auch auf der Server-Seite zu wählen sind.

Nach der Konfiguration der Domäne selbst werden für die verschiedenen Anwendungen noch die jeweiligen „Managed Server“ erstellt und konfiguriert. In unserem Fall erstreckt sich die Domäne über zwei physikalische Rechner. Um Hochverfügbarkeit zu gewährleisten, wird zunächst jeweils ein „Managed Server“ pro Rechner in einem Anwendungscluster zur Verfügung gestellt. Sollten weitere Ressourcen notwendig sein, kann dieses Cluster dann um zusätzliche „Managed Server“ erweitert werden. Außerdem werden die benötigten Datenquellen nach Vorgabe der Entwickler erstellt und den zugehörigen Clustern zugewiesen.

All diese Konfigurationsschritte können wahlweise über eines der mitgelieferten Administrations-Tools durchgeführt werden: über die beiden grafischen Oberflächen Administrations-Konsole des WebLogic-Servers bzw. den Enterprise Manager (Fusion Middleware Control) oder aber über das WebLogic-Scripting-Tool (WLST). WLST ist eine Scripting-Umgebung (Kommandozeilen). Es basiert auf dem Java Scripting-Interpreter Jython und wird zum Verwalten und Überwachen von Weblogic Server Domains verwendet. WLST ist besonders gut für die automatisierte Konfiguration des Servers geeignet. Beispielsweise lassen sich Datasources oder auch Authentifizierungsprovider automatisiert auf den einzelnen Umgebungen einrichten.

### **Anwendungen deployen**

Mit dem Weblogic-Server eröffnen sich neue Möglichkeiten das Deployment von Anwendungen zu gestalten. Nach wie vor kann auch beim Weblogic-Server eine Anwendung mittels der Web-Administrationsoberfläche deployt werden. Eine weitere Möglichkeit die Applikation zu deployen besteht in der Nutzung von WLST. Seit dem Weblogic-Server 11g stellt Oracle seinen Kunden auch ein Maven Plug-in zur Verfügung. Dieses Plug-in ist besonders für die Softwareentwickler interessant. Es ermöglicht das Deployen von Anwendungen auf verschiedenen Umgebungen aus einem Maven build heraus. Dies ist vor allem nützlich, wenn ein Continuous Integration System, wie beispielsweise Jenkins verwendet wird. Nightly Builds auf einem Entwicklungsserver lassen sich so sehr einfach konfigurieren.

### **Entwicklungsumgebung**

Nachdem die Infrastruktur aufgesetzt und konfiguriert ist sowie Anwendungen deployt und zur Verfügung gestellt werden können, müssen die alten 10g-Anwendungen auf den neuen Weblogic-Server 12c umgestellt werden. Der OC4J 10g ist ein vollzertifizierter JEE 1.4 Applikationsserver, der bereits Teile der Spezifikation von JEE 1.5 unterstützt, wohingegen der Weblogic-Server 12c ein JEE 1.6 Server ist.

Bevor die Anwendungen migriert werden, muss eine Entwicklungsumgebung für die Migration und Weiterentwicklung festgelegt werden. Diese Entwicklungsumgebung muss den JEE 1.6 Standard unterstützen. Die vier meistgenutzten IDEs im Oracle Enterprise Umfeld sind Eclipse, IntelliJ, Netbeans und JDeveloper. All diese Entwicklungsumgebungen unterstützen den Java EE 1.6 Standard und bieten Konnektoren für die Verwendung des Weblogic-Server 12c an. Lediglich IntelliJ ist keine vollständige Freeware und muss für die Entwicklung von Java EE Applikationen lizenziert werden. Bei dem Oracle Kunden SüdLeasing kommt Eclipse Kepler in Kombination mit dem Oracle Enterprise Pack und den JBoss Tools zum Einsatz.

### **Migration der Modellschicht**

Nachdem die Entwicklungsumgebung definiert wurde, kann die Migration der Applikationen beginnen. Bei SüdLeasing wurde hierfür der Bottom-up-Ansatz gewählt. Das heißt es wird zuerst die Modellschicht, dann die Controllerschicht und schliesslich die Präsentationsschicht migriert. Bei der Migration der Modellschicht wird meist von JPA 1.0 Toplink zu JPA 2.0 EclipseLink migriert. Dazu müssen die neuen EclipseLink Libraries eingebunden, die alten Toplink Libraries entfernt und die persistence.xml-Datei auf den neuen 2.0 Standard umgestellt werden. Gegebenenfalls sind noch geringfügige Änderungen an den Entitäten nötig.

### **Migration der Controllerschicht**

Bei der Migration der Controllerschicht wird entweder vom EJB 3.0 oder vom EJB 2.1 Standard auf den EJB 3.1 Standard migriert. Bei SüdLeasing stellt die Ausgangsbasis ein EJB 3.0 Modul dar. Daher ist der Migrationsaufwand bei der Umstellung gering.

Jetzt müssen noch die alten Deployment-Deskriptoren auf die neuen Weblogic-Deskriptoren umgestellt werden. Für die Controllerschicht muss bzw. kann für die Konfiguration der EJBs eine weblogic-ejb-jar.xml-Datei erstellt werden. In diesem Deskriptor können unter anderem die Transaction Timeout-Zeiten sowie Security- und Initialisierungsregeln konfiguriert werden. Falls die Controllerschicht in einer EAR-Datei ausgeliefert wird, kann die Applikation noch mit Hilfe der weblogic-application.xml-Datei konfiguriert werden. In diesem Deskriptor können beispielsweise bevorzugte Packages oder Shared Libraries spezifiziert werden.

### **Migration der Präsentationsschicht**

Der Weblogic-Server 12c unterstützt im JEE 1.6 Standard erstmals JSF 2.0 und JSF 2.1. Da die alten Applikationen im OC4J noch mit JSF 1.1 laufen, ist es ratsam, diese auf den neuen JSF Standard zu migrieren und dabei gleichzeitig auf das neue „xhtml“-Format umzustellen. Oracle ADF wurde von Oracle an Apache gespendet. Falls ADF verwendet wurde, muss ADF zu Apache MyFaces Trinidad migriert werden.

Diese Umstellung stellt bei SüdLeasing den aufwändigsten Teil der gesamten Migration dar. Ein Teil dieser Web-Migration wurde mit dem integrierten Migrationswerkzeug des JDevelopers 11g durchgeführt. Da das Migrationswerkzeug allerdings nur einen kleinen Teil der Migration übernimmt, muss in jedem Fall noch manuell nachgebessert werden. Sobald die JSF-Seiten umgestellt sind, werden auch die JSF Beans, welche in der faces-config.xml-Datei konfiguriert sind, auf den neuen CDI 1.0 Standard umgestellt.

Auch in der Präsentationsschicht müssen die alten Deployment-Deskriptoren umgestellt werden. Hierfür muss eine weblogic.xml-Datei erstellt werden. In dieser kann dann die Konfiguration des Web-Moduls erfolgen. Eine der häufigsten Konfigurationen ist beispielsweise das Festlegen von Security Role Assignments. Diese Assignments werden für die Authentifizierung von Benutzern der Web-Applikation benötigt. Weiterhin können auch in diesem Deskriptor bevorzugte Packages und Shared Libraries angegeben werden.

### **Fazit**

Es wird aufgezeigt, wie die neue 12c-Infrastruktur aufgebaut und eingerichtet werden kann. Neue Möglichkeiten wie die Verwendung von Oracle WLST werden erläutert.

Danach wird der Schwerpunkt auf die Migration bestehender OC4J Applikationen gelegt. Hierfür wird zunächst aufgezeigt, welche Möglichkeiten bei der Wahl der Entwicklungsumgebung bestehen, woraufhin dann erläutert wird, welche Schritte bei der Migration der Applikationen nötig sind. Dies wird anhand eines Migrationsszenarios der SüdLeasing GmbH erläutert.

**Kontaktadressen:**

Andreas Wussler  
Software Ingenieur und Consulting  
PROMATIS software GmbH  
Pforzheimer Str. 160  
76275 Ettlingen

Telefon: +49(0)7243-2179-0  
Fax: +49(0)7243-2179-99  
E-Mail: andreas.wussler@promatis.de  
Internet: www.promatis.de

Dr. Joachim Reising  
Senior Principal Consultant  
PROMATIS software GmbH  
Pforzheimer Str. 160  
76275 Ettlingen

Telefon: +49(0)7243-2179-0  
Fax: +49(0)7243-2179-99  
E-Mail: joachim.reising@promatis.de  
Internet: www.promatis.de

Marc Brenkmann  
Dipl.-Wirtschaftsinformatiker (DH)  
SüdLeasing GmbH  
Pariser Platz 7  
70155 Stuttgart

Telefon: +49(0)621-4281-1185  
Fax: +49(0)621-428651-1185  
E-Mail: marc.brenkmann@suedleasing.com  
Internet: www.suedleasing.com