

Upgrade von Java Anwendungen von OracleAS 10g auf FMW 11g

Michael Bräuer & Thomas Robert
Oracle Deutschland B.V. & Co. KG

Schlüsselworte:

OC4J, Oracle AS 10g, Upgrade, Migration, Oracle WebLogic Server, WLS, FMW 11g

Einleitung

Mit der Übernahme von BEA Systems durch Oracle hat der WebLogic Server den im Oracle Application Server 10g verwendeten OC4J (Oracle Containers for J2EE) als Fundament für die FMW 11g Plattform abgelöst. Damit Kunden von den umfangreichen Produkteigenschaften der neuen Version profitieren können, müssen bestehende Java Anwendungen vom OC4J auf den Oracle WebLogic Server gebracht werden. Der Vortrag beschreibt die Vorgehensweise für so einen Upgrade und geht dabei auch auf Toolunterstützung ein.

Perspektiven

Ein Upgrade vom OC4J auf den WebLogic Server kann aus unterschiedlichen Perspektiven betrachtet werden. Es ergeben sich für verschiedenen Nutzungsgruppen andere Fragestellungen. Die folgende Tabelle fasst einige dieser Fragen zusammen, die vor bzw. während eines Upgrades beantwortet werden müssen.

Benutzergruppe	Fragen
Entwicklung	Welche Features kann ich einsetzen, welche sind überholt? Welche Entwicklungswerkzeuge kann ich benutzen? Welche Werkzeuge stehen für den Upgrade zur Verfügung? Wie benutze ich 3rd Party Bibliotheken? Muss ich Veränderungen am Code durchführen? Welche Oracle Bibliotheken muss ich nutzen?
Administration und Überwachung	Welche konzeptionellen Unterschiede gibt es in der Architektur? Was muss ich wie installieren? Wie mache ich meine Umgebung hochverfügbar? Welche Überwachungsmöglichkeiten gibt es? Kann ich DMS Metriken weiterhin benutzen? Wo und wie kann ich tunen? Wie konfiguriere ich Logging? Welche Command Line Utilities gibt es?
Deployment	Wie müssen Anwendungen paketiert werden? Welche Tools stehen für das Deployment zur Verfügung? Gibt es Unterstützung für gängige Deployment-Werkzeuge (ANT, Maven)? Kann ich meine bisherigen Deploymentpläne benutzen? Wie funktioniert das Class-Loading? Gibt es Shared Libraries?

Unser Vortrag kann im Rahmen der verfügbaren Zeit nicht auf alle diese Punkte eingehen. Vielmehr konzentriert er sich auf den eigentlichen Upgrade von Java EE Anwendungen, die unterstützenden Werkzeuge und gibt Tipps aus Erfahrungen bereits durchgeführter Upgrade-Projekte.

Vergleich OC4J – Oracle WebLogic Server: 1*1 für Administratoren

Im Oracle Application Server 10g ist die Verwaltungseinheit eine *Farm* (10.1.2.x) bzw. ein *Cluster* (10.1.3.x). Farm bzw. Cluster bestehen aus einem oder mehreren OC4J Containern (JVM Prozesse), die über einen zentralen Administrationsserver verwaltet werden. Ein OC4J Container betreibt eine oder mehrere JVMs. Laufen in einem OC4J Container mehrere JVMs, so nutzen sie dieselbe Konfiguration. Im Unterschied zur Farm bietet das Cluster neben der gemeinsamen Administration zusätzliche Loadbalancing und Failover Funktionalitäten an.

Die Verwaltungseinheit der FMW 11g wird *Domäne* genannt. In einer Domäne werden ein oder mehrere WebLogic Managed Server (JVM Prozesse) über einen zentralen Administrationsserver verwaltet. Gruppen von Managed Servern innerhalb einer Domäne können zu Clustern verbunden werden, um zusätzlich Lastverteilung und Ausfallsicherheit zu erreichen.

Im Oracle Application Server 10g übernimmt der Oracle Process Manager and Notification (OPMN) das Starten und Stoppen sowie die Überwachung aller Prozesse *eines ORACLE_HOMES*. In der FMW 11g wird diese Funktionalität *rechnerweit* durch den WebLogic NodeManager bereit gestellt.

Die OC4J Container des Oracle Application Servers 10g unterscheiden zwischen Datasources und Connection Pools. Connection Pools definieren den Zugriff auf die Datenquellen, während DataSources den Zugriff auf die Connection Pools beschreiben. Da sie unabhängig voneinander konfigurierbar sind, können verschiedene DataSources denselben Connection Pool benutzen. Beim Zugriff auf eine RAC Datenbank werden über den Oracle JDBC Treiber zusätzliche Funktionalitäten wie „Runtime Connection Loadbalancing“ und „Fast Connection Failover“ bereit gestellt.

Der WebLogic Server der FMW 11g definiert seine Connection Pools jeweils innerhalb der DataSources. Somit gibt es immer eine 1:1-Verbindung zwischen diesen beiden Schichten. Der Zugriff auf eine RAC Datenbank wird über sogenannte „Multi-DataSources“ geregelt. Die Multi-DataSource ist ein logisches Konstrukt, das über mehrere DataSources gelegt wird. Jede DataSource definiert dabei den Zugriff auf einen spezifischen RAC Knoten. Die Multi-DataSource beschreibt den Zugriff aus der Anwendung auf diese einzelnen DataSources und definiert, ob der Zugriff auf die RAC Knoten im „Loadbalancing“ oder „Failover“ Modus erfolgen soll. Seit der Version 11g Patchset 4 stellt Oracle mit „Active GridLink for RAC“ zusätzlich eine neue DataSource bereit, die neben der Lastverteilung auf die RAC Knoten auch „Runtime Connection LoadBalancing“ und „Fast Connection Failover“ unterstützt.

Im Gegensatz zum OC4J verarbeitet der WebLogic Server normalerweise alle Protokolle auf demselben Port. Eine separate Port-Konfiguration für RMI, JMS und HTTP kann aber über sogenannte Network Channels konfiguriert werden. Im Gegensatz zum OC4J unterstützt WebLogic auch nicht das AJP Protokoll. Die Kommunikation zwischen Webserver und Java EE Container, die im Oracle Application Server 10g auf dem AJP Protokoll basierte, läuft in der FMW 11g Umgebung somit ausschließlich über das HTTP Protokoll.

In einer Oracle Application Server 10g Installation werden alle Konfigurationsdateien innerhalb des ORACLE_HOMEs der Installation abgelegt. Konfigurationen und Oracle Software liegen also gemeinsam innerhalb einer Verzeichnisstruktur.

Die FMW 11g trennt strikt zwischen Binaries und Konfigurationen. Eine Installation kann also von mehreren Domänen – bei Nutzung eines Shared Filesystems sogar über mehrere Rechner hinweg – verwendet werden. Die Konfigurationen jeder Domäne liegen getrennt von der Software. Der NodeManager kann ebenfalls getrennt von Software und Domänen konfiguriert werden. Dieses Konzept entspricht dem der Oracle Datenbank, in dem die Oracle Datenbank Software ebenfalls getrennt von der Datenbank und ihrer Konfiguration liegt.

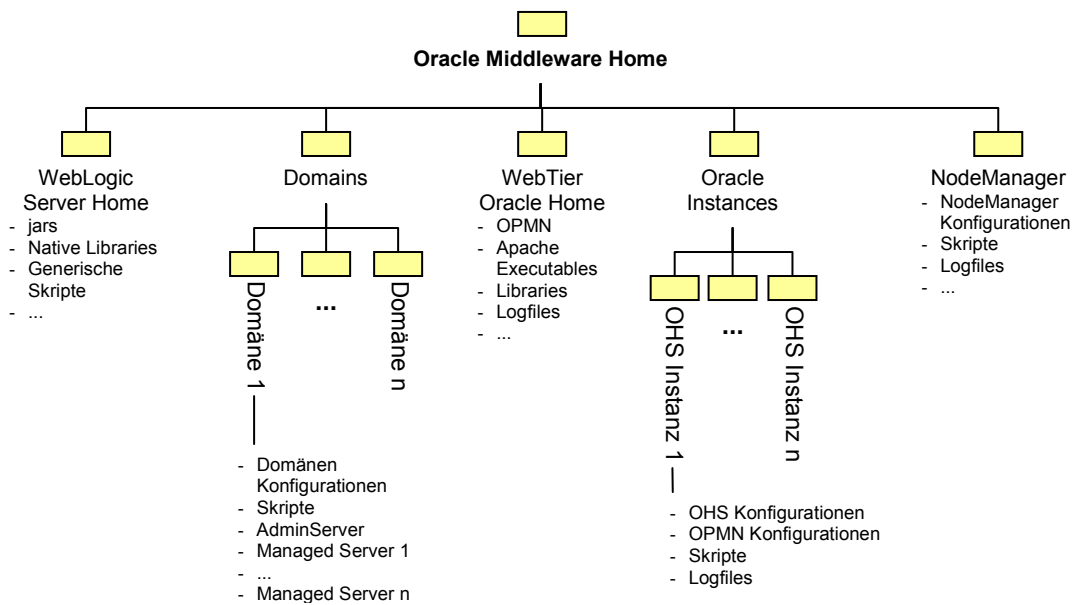


Bild 1: Verzeichnis Layout FMW 11g (Beispiel): Trennung von Binaries und Konfigurationen

Grundsätzliche Vorgehensweise

Der Upgrade einer Java Anwendung vom Oracle Application Server 10g auf den WebLogic Server der FMW 11g gliedert sich in mehrere Schritte. Zunächst muss die neue Umgebung bereit gestellt werden. Die Applikation muss auf die neue Version des Application Servers upgegradet werden. Hierfür sind Modifikationen – z.B. an Deployment Deskriptoren, eventuell auch am Code – notwendig. Bei dieser Gelegenheit kann auch überlegt werden, Funktionalitäten, die die neue Umgebung bereit stellt, im Zuge des Upgrades zu implementieren. Anschließend muss das Deployment auf die neue Umgebung angepasst werden. Dies bedeutet ggf. eine Anpassung der ant-Skripte bzw. der Deployment-Profile in der Entwicklungsumgebung.

Das folgende Bild veranschaulicht die unterschiedlichen Schritte, die in der Regel für den Upgrade durchlaufen werden.

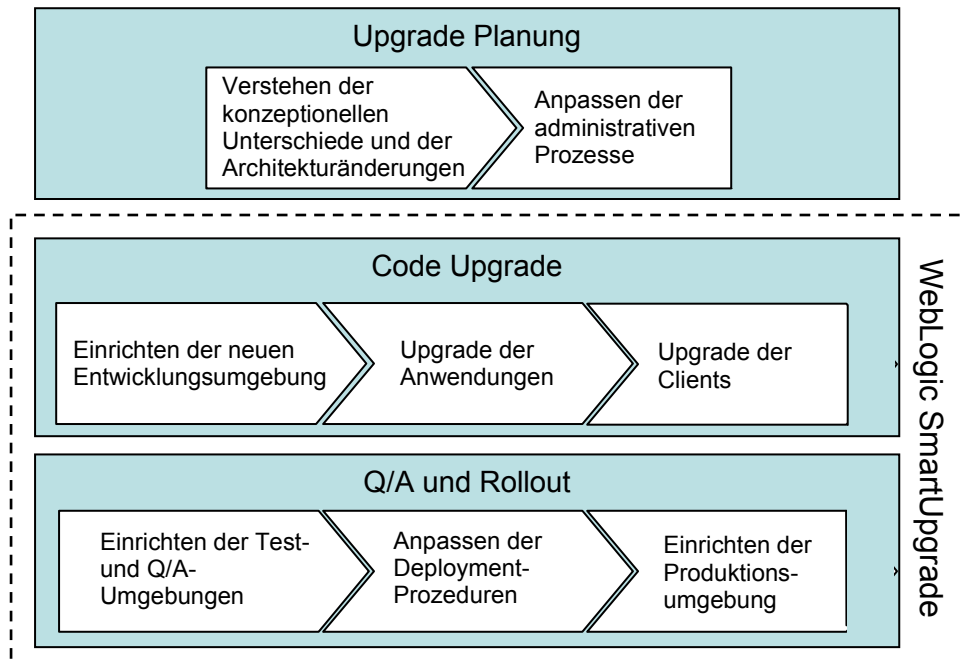


Bild 2: Projekt-Schritte für den Upgrade von OC4J auf WLS

Upgrade von Java EE Anwendungen ^[5]

Sowohl der in der FMW 11g enthaltene WebLogic Server 10.3 als auch der OC4J 10g unterstützen Java EE 5. Somit ist der Java Code weitgehend ohne Änderung in beiden Umgebungen lauffähig. Neben den standardisierten Java EE Deployment Deskriptoren werden in beiden Umgebungen jedoch auch Container-spezifische Deskriptoren verwendet. Werden diese in der Applikation genutzt, müssen sie beim Upgrade entsprechend angepasst werden.

Ressourcen wie JDBC Verbindungen oder JMS Objekte müssen im WebLogic Server vor dem Deployment angelegt werden. In der Regel gibt es hier ein Mapping der OC4J Ressourcen auf die WebLogic Services.

Wird für JDBC Zugriffe eine RAC Datenbanken benutzt, so müssen die DataSources im WebLogic Server als Multi-Datasource oder als Active GridLink for RAC Datasource angelegt werden. Der Anwendungscode ist hiervon nicht betroffen.

JMS Objekte des OC4J (z.B. für MDBs) müssen ebenfalls vor dem Deployment als JMS Module im WebLogic Server angelegt werden. Die Version 11g bietet hier eine Reihe neuer Features an, die die Skalierbarkeit und Ausfallsicherheit von Messages deutlich verbessern.

Startup und Shutdown Klassen sind in Java EE nicht spezifiziert. Im Oracle Application Server 10g mussten hier spezielle OC4JStartup und OC4JShutdown Interfaces implementiert werden. Dies ist im WebLogic Server nicht mehr notwendig. Vielmehr erlaubt der WebLogic Server mit der Implementierung von Listener Interfaces auf eine Reihe von Pre- und Post-Events zu reagieren.

Weitere Anpassungen sind eventuell beim Packen von jar-Libraries in die Applikationsarchive notwendig. So kann z.B. der OC4J in seinem orion-application.xml über das <library> Attribut auf Java Libraries verweisen. Im WebLogic Server können diese Libraries einfach in das APP-INF/lib Verzeichnis des Enterprise Archivs kopiert werden.

Toolunterstützung für den Upgrade

Nach der Installation der FMW 11g stehen eine Reihe von Werkzeugen zur Verfügung, um eine bestehende 10g Umgebung auf die neue Version zu bringen. Dieser Vortrag konzentriert sich auf den Upgrade von Java EE Applikationen. Der Upgrade Assistant und Migration Wizard werden hier nur der Vollständigkeit halber erwähnt:

1.) Oracle Fusion Middleware Upgrade Assistant^{[1] [2] [3]}

Der Upgrade Assistant unterstützt den Upgrade einer 10g Portal, Forms, Reports, Discoverer oder Identity Management Installation. Er übernimmt sowohl den Upgrade der Meta-Informationen in der Datenbank (Portal-, Reports-, Discoverer, B2B-, BAM- bzw. IDM-Schemata) als auch die Übernahme der produktspezifischen Konfigurationen in die 11g Umgebung (formsweb.cfg, reports.conf, etc.)

2.) JDeveloper Migration Wizard^[3]

Prozesse der Oracle SOA Suite 10g sowie Webcenter 10g Applikationen werden automatisch durch Öffnen der entsprechenden Projekte im JDeveloper 11g migriert. Diese Projekte müssen dann auf die neue 11g Umgebung deployt werden.

3.) WebLogic SmartUpgrade^[4]

Das WebLogic SmartUpgrade Utility unterstützt den Entwickler beim Upgrade einer Anwendung von OC4J 10.1.2.x oder 10.1.3.x auf den WebLogic Server der FMW 11g. Außerdem übernimmt es die Konvertierung von OC4J Datasources und gibt konkrete Empfehlungen für das Einrichten weiterer Java EE Komponenten wie JMS Servern, Queues, Topics etc.

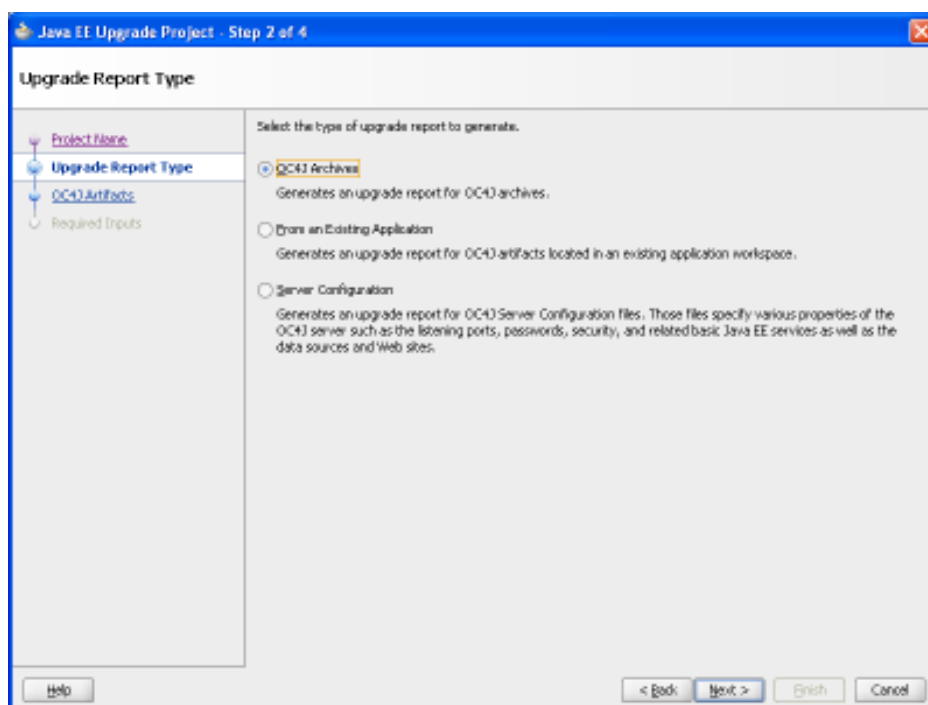


Bild 3: Oracle JDeveloper SmartUpgrade Wizard

Das Tool existiert als PlugIn für den JDeveloper 11g und als standalone Command-Line Tool. Es analysiert JDeveloper Projekte oder Applikationsarchive (ear, war, rar oder jar Files) und gibt

Empfehlungen für deren Upgrade. Soweit möglich schlägt SmartUpgrade äquivalente Einstellungen für die WebLogic spezifischen Deployment Deskriptoren vor.

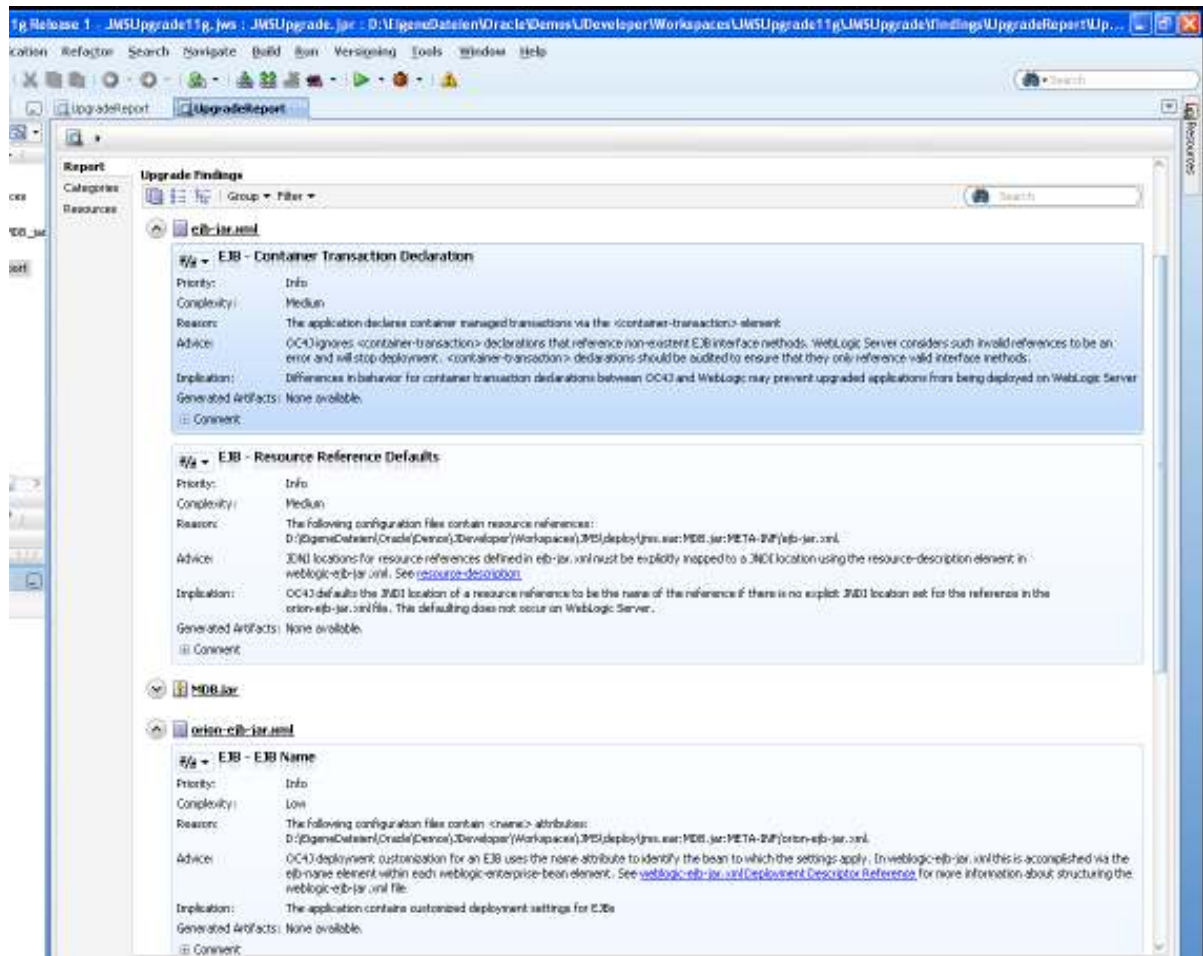


Bild 4: Upgrade Report einer Applikation

Optional können vom SmartUpgrade empfohlene Deployment Deskriptoren auch automatisch generiert werden.

SmartUpgrade bietet die gleiche Funktionalität auch als Command Line Tool. Das folgende Beispiel zeigt die automatische Generierung der JDBC Datasources anhand der OC4J datasources.xml.

```
[SmartUpgrade]$ $JAVA_HOME/bin/java -jar smartupgrade.jar \
  --server-config $ORACLE_HOME/j2ee/oc4j_soa/config/ \
  -generate > upgrade.log

[SmartUpgrade]$ ls upgrade/server_config.d/
BPELSamplesDataSource-jdbc.xml          ESBAQJMSDataSource-jdbc.xml
BPELServerDataSource-jdbc.xml          ESBDDataSource-jdbc.xml
BPELServerDataSourceWorkflow-jdbc.xml   OracleDS-jdbc.xml
config.xml
```

Bild 5: Command Line Version des SmartUpgrade Utilities

Upgrade der Clients

Clients einer Java EE Applikation müssen auf den geänderten JNDI Provider angepasst werden. Macht ein OC4J Client z.B. einen EJB Lookup, so sieht der in etwa folgendermaßen aus:

```
INITIAL_CONTEXT_FACTORY=com.evermind.server.RMIInitialContextFactory
//bzw. com.evermind.server.ApplicationClientInitialContextFactory
PROVIDER_URL=opmn:ormi://host:opmn_port:oc4j_container/application
```

Mit der oben dargestellten URL wird der OPMN angefragt, dem Client die Adresse aller OPMN Container im Cluster, in denen die Applikation deployt ist, zurückzugeben. Anschließend verbindet sich der Client auf einen dieser Container. Der gleiche Zugriff sieht unter WebLogic folgendermaßen aus:

```
INITIAL_CONTEXT_FACTORY=weblogic.jndi.WLInitialContextFactory
PROVIDER_URL=t3://host:admin_port
```

Da WebLogic im Gegensatz zum OC4J einen globalen JNDI Baum benutzt, ist darauf zu achten, dass innerhalb einer Domäne alle JNDI Namen eindeutig sind. Es können also nicht zwei unterschiedliche Applikationen (oder andere Objekte) unter gleichem Namen auf zwei verschiedene Managed Server innerhalb einer Domäne deployt werden.

Zusammenfassung

Mit dem Oracle WebLogic Server bietet Oracle einen hochperformanten, stabilen und leistungsfähigen Java EE 5 Container in dem OC4J Applikationen weiter betrieben werden können. Notwendige Anpassungen werden durch eine Reihe von Utilities und ausführliche Dokumentation unterstützt.

Anhang

- [1] Oracle Fusion Middleware Upgrade Guide for Oracle Portal, Forms, Reports, and Discoverer
- [2] Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management
- [3] Oracle Fusion Middleware Upgrade Guide for Oracle SOA Suite, WebCenter, and ADF
- [4] Oracle Fusion Middleware SmartUpgrade User's Guide
- [5] Oracle Fusion Middleware Upgrade Guide for Java EE

Kontaktadresse:

Thomas Robert
Oracle Deutschland B.V. & Co. KG
Kühnehöfe 15
D-22761 Hamburg
Telefon: +49 (0) 40-890 91 188
E-Mail: thomas.robert@oracle.com
Internet: www.oracle.de

Michael Bräuer
Oracle Deutschland B.V. & Co. KG
Schiffbauergasse 14
D-14467 Potsdam
Telefon: +49 (0) 30-435 795 206
E-Mail: michael.braeuer@oracle.com
Internet: www.oracle.de