



Abbildung 6: ADF Mobile Data Controls

einer mobilen Integration in Oracles Fusion Middleware. Unternehmen, die

bereits SOA im Einsatz haben, profitieren an dieser Stelle von den bei ihnen vorhandenen Voraussetzungen für die Anbindung von mobilen Endgeräten. Der Autor betont an dieser Stelle die Wichtigkeit, wartbare Architekturen zu schaffen und nur über eine zentrale Infrastruktur auf die Unternehmensdienste zuzugreifen (etwa mittels OSB oder OEG).

In der aktuellen Version hat das Framework zwar noch mit kleineren Kinderkrankheiten zu kämpfen, etwa mit mangelnden Lösungen für die Synchronisation von offline erfassten Daten. Allerdings könnte bereits im kommenden Jahr eine neue Version erscheinen, die dann dank einer schlanken Java 8 VM und neuer Features wohl auch die letzten Kritiker überzeugen wird.

Literatur

- [1] R. C. Basole: Mobilizing the enterprise: A conceptual model of transformational value and enterprise readiness, 26th ASEM National Conference Proceedings, 2005
- [2] A. Pousttchi, und K. Gump: The Mobility-M-Framework for Application of Mobile Technology in Business Pro-

cesses, 35. GI-Jahrestagung Informatik, 2005

- [3] A. Pousttchi und K. Turowski: Mobile Commerce, Springer, 2003
- [4] R. Kuhlen, R. Hammwöhner und J. Herget: Informationsmarkt: Chancen und Risiken der Kommerzialisierung von Wissen, Universitätsverlag, 1995
- [5] U. Wellen: Process Landscaping – Eine Methode zur Modellierung und Analyse verteilter Softwareprozesse, Dissertation, 2003

Enno Schulte

enno.schulte@opitz-consulting.com



Web-Services in der Oracle-Datenbank

Detlev Schütte, ORBIT Gesellschaft für Applikations- und Informationssysteme mbH

Dieser Artikel beschreibt Web-Services auf der Oracle-Datenbank, die auf der Funktionalität „XML DB“ basieren. Zielgruppe sind Entwickler und Administratoren. Einer allgemeinen Beschreibung folgt eine systematische Abhandlung der Schritte zum Einrichten eines funktionsfähigen Web-Service. Sicherheitsaspekte sind ebenfalls berücksichtigt.

Web-Services sind ein gängiger Weg des Informationsaustauschs über das Internet (B2B, B2C) und spielen daher in der heutigen IT-Welt eine bedeutende Rolle. Im Wesentlichen laufen folgende Prozesse ab:

- Der Konsument ruft beim Anbieter eine Information ab, indem er an dessen URL eine Anfrage sendet.

- Der Web-Service des Anbieters bearbeitet die Anfrage des Konsumenten und sendet die Antwort zurück.

Üblicherweise geschieht das alles mittels einer XML-Anfrage unter Nutzung des Netzwerkprotokolls SOAP (Simple Object Access Protocol). Seit der Datenbank-Version 10g bietet die neuen Funktionalität „XML DB“ die Möglichkeit,

Web-Services mit Datenbank-Mitteln anzubieten und zu konsumieren. Oracle XML DB unterstützt SOAP in der Version 1.1. Mit einem solchen Service lassen sich SQL- und XQuery-Abfragen erstellen, und eine Darstellung der Resultate im XML-Format ist ebenfalls möglich. Zu letzterem Zweck können PL/SQL-Prozeduren genutzt werden, deren In- und Out-Parameterliste das Format

der ein- beziehungsweise auslaufenden Dokumente festlegen. Die zur Parametrisierung des Web-Service benötigten WSDL-Dokumente (WSDL = Webservice Description Language) werden automatisch durch die Funktionalität „XML DB“ der Oracle-Datenbank generiert.

Oracle-XML-DB-Web-Services

Aus Sicherheitsgründen sind Web-Services in der Datenbank normalerweise nicht aktiviert. Für eine Nutzung muss man sie extra konfigurieren und aktivieren. Es wird dringend empfohlen, hierzu einen eigenen User namens „<<WS_USER>>“ ohne DBA-Privilegien anzulegen.

Beim Austausch sensibler Informationen ist die Nutzung eines Web-Service über HTTPS angeraten. Falls die Kommunikationspartner einander vertrauen, kann die Abfrage von Benutzername und Passwort entfallen. Dies ist bei Job-gesteuerten Abfragen immer der Fall, da hier eine manuelle Authentifizierung nicht praktikabel ist.

Um einen vertrauensvollen Zugang zu ermöglichen, müssen die beteiligten Partner Sicherheitszertifikate austauschen und an bestimmter Stelle ablegen. Ablageort für Sicherheitszertifikate bei XML DB ist das sogenannte „Wallet“. Diese Datei liegt im Verzeichnis „ORACLE_HOME/admin/<<Name der Datenbank Instanz>>“ des Datenbank-Servers; ihr Ort kann individuellen Wünschen angepasst werden. Der Pfad ist der Datenbank bekannt zu machen; in den Code-Beispielen ist er mit „<<Pfad für Oracle-Wallet>>“ gekennzeichnet.

Falls ein Oracle-Wallet noch nicht existiert, muss es erstellt werden. Dies kann entweder durch eine Client- oder eine Server-seitige GUI geschehen oder auf der Kommandozeile des Datenbank-servers. Voraussetzung ist ein X-fähiges Terminal, es muss also die grafische Version von UNIX vorhanden sein. Das Dienstprogramm zum Aufruf der GUI lautet „owm“. Es ist im Verzeichnis „ORACLE_HOME/bin“ zu finden.

Im Folgenden wird das Verfahren über die Kommandozeile beschrieben. Zur Ausführung der Kommandos ist eine Anmeldung als User „ORACLE“ auf dem Datenbank-Server erforderlich. Das Dienstprogramm zum Anle-

gen, Konfigurieren und Anzeigen eines Wallet lautet „orapki“ und ist im Verzeichnis „ORACLE_HOME/bin“. Es ist in mehrfacher Weise über eine Reihe von Parametern einsetzbar. Insbesondere wird ein Passwort erwartet, das in den Code-Beispielen als „<WALLET_Passwort>“ gekennzeichnet wird. **Listing 1** zeigt das Kommando zum Anlegen eines leeren Wallet.

Wie bereits erwähnt, lassen sich im Wallet Zertifikate der Kommunikationspartner ablegen. Bei Bedarf können sie zusammengesetzt sein, so können sie beispielsweise aus dem Zertifikat einer autorisierten, ausstellenden Organisation (Certification Authority, kurz CA) und einem User-Zertifikat bestehen (siehe **Listing 2**).

Zu Testzwecken lässt sich auch ein eigenes, selbst signiertes Zertifikat erstellen, das man dem Kommunikationspartner zur Verfügung stellen kann.

Seine Gültigkeitsdauer lässt sich begrenzen. Für seine Erstellung ist ein „Distinguished Name“ (DN) erforderlich. Dieser enthält identifizierende Angaben wie den Namen des Zertifikats, der Organisation, der Organisationseinheit, des Standorts sowie die Namen beliebiger geographischer Angaben. Die Angaben sind durch bestimmte Kürzel gekennzeichnet. Ein DN könnte beispielsweise so aussehen: „CN = Name_des_Zertifikats, O = Name_der_Organisation, OU = Name_der_Organisationseinheit, L = Stadt, ST = Bundesland, C = DE“ (siehe **Listing 3**). Der Inhalt eines Wallet lässt sich mit dem in **Listing 4** gezeigten Kommando darstellen.

Konfiguration eines Servlet

Der User „SYS“ muss die Servlet-Konfiguration erstellen. Dies geschieht mit der Konsolen-Anwendung „SqlPlus“.

```
$ORACLE_HOME/bin/orapki wallet create -wallet <Pfad für Oracle-Wallet> -pwd <WALLET_Passwort> -auto_login
```

Listing 1: Wallet anlegen

```
# User-Zertifikat importieren (Teil 1 von 2 der Zertifikats-Kette)
$ORACLE_HOME/bin/orapki wallet add -wallet <Pfad für Oracle-Wallet>
-trusted_cert -cert <User-Zertifikat> -pwd <WALLET_Passwort>
# CA-Zertifikat importieren (Teil 2 von 2 der Zertifikats-Kette)
$ORACLE_HOME/bin/orapki wallet add -wallet <Pfad für Oracle-Wallet>
-trusted_cert -cert <CA-Zertifikat> -pwd <WALLET_Passwort>
```

Listing 2: Zertifikate ins Wallet importieren

```
# zu Testzwecken eigenes, selbst signiertes Zertifikat erstellen
# Gültigkeitsdauer des Zertifikats: 3650 Tage (10 Jahre)
$ORACLE_HOME/bin/orapki wallet add -wallet <Pfad für Oracle-Wallet>
-keysize 1024 -dn '<distinguished Name>' -self_signed -pwd <WALLET_Passwort>
-validity 3650
# Eigenes, selbst signiertes Zertifikat exportieren in Datei
# Hinweis: die Datei kann dem Kommunikationspartner für Testzwecke
zur Verfügung gestellt werden
$ORACLE_HOME/bin/orapki wallet export -wallet <Pfad für Oracle-Wallet>
-dn '<distinguished Name>' -cert <Dateiname für Zertifikat> -pwd
<WALLET_Passwort>
```

Listing 3: Selbst signiertes Zertifikat erstellen und in Datei exportieren

```
$ORACLE_HOME/bin/orapki wallet display -wallet <Pfad für Oracle-Wallet>
-pwd <WALLET_Passwort>
```

Listing 4: Inhalt eines Wallet anzeigen

```

-- Servlet und Servlet-Mapping erstellen
DECLARE
  SERVLET_NAME VARCHAR2(32) := 'orawsv';

BEGIN
  DBMS_XDB.addServlet(NAME      => SERVLET_NAME,
                     LANGUAGE => 'C',
                     DISPNAME => 'Oracle Query Web Service',
                     DESCRIPT => 'Servlet for issuing queries as a Web Service',
                     SCHEMA   => 'XDB');
  DBMS_XDB.addServletSecRole(SERVNAME => SERVLET_NAME,
                             ROLENAM  => 'XDB_WEBSERVICES',
                             ROLELINK => 'XDB_WEBSERVICES');
  DBMS_XDB.addServletMapping(PATTERN => '/orawsv/*',
                              NAME    => SERVLET_NAME);

END;
/

-- Ergebniskontrolle-1: Check Servlet
SELECT XMLSerialize(content
  XMLQuery('declare namespace ns="http://xmlns.oracle.com/xdb/xdbconfig.xsd"; (: : )
    $r//ns:servlet[ns:servlet-name="orawsv"]'
    passing DBMS_XDB.CFG_GET() AS "r" returning content)
  AS CLOB ) servlet_config
FROM dual
;

-- Ergebniskontrolle-2: Check Webservice Mapping "orawsv"
select extractValue(value(x),'/servlet-mapping/servlet-name', 'xmlns="http://xmlns.oracle.com/xdb/xdbcon-
fig.xsd"') "WebServiceMapping"
  from table(xmlsequence(extract(xdburitype('/xdbconfig.xml').getXML(),'xdbconfig/sysconfig/protocolcon-
fig/httpconfig/webappconfig/servletconfig/servlet-mappings/servlet-mapping'))) x
;

```

Listing 5: Konfiguration Web-Service – Servlet erstellen

Der Name des Servlet ist in diesem Beispiel „orawsv“ (siehe Listing 5).

Falls keine andere Konfiguration vorgenommen wurde, läuft die Kommunikation bei der Oracle-Funktionalität „XML DB“ unverschlüsselt über den Standard-Port 8080. Aus Sicherheitsgründen ist aber eine verschlüsselte Kommunikation vorzuziehen, beispielsweise über den Port 8893 (siehe Listing 6).

Oracle-Listener

Der Oracle-Listener-Prozess muss über die konfigurierten Kommunikations-Ports informiert werden. Das geschieht zum einen innerhalb der Datenbank (Listener Endpoint) und zum anderen in den Dateien „listener.ora“ und „sqlnet.ora“. In der Datenbank wird der Listener Endpoint mit dem in Listing 7 gezeigten SQL-Kommando gesetzt.

In den Dateien „listener.ora“ und „sqlnet.ora“ (standardmäßig im Verzeichnis „ORACLE_HOME/network/admin“ zu finden) muss die Netzwerk-Konfiguration angepasst werden. Die

Datei „listener.ora“ ist wie folgt zu ergänzen (siehe Listing 8). Darüber hinaus ist in der Datei „sqlnet.ora“ folgender Eintrag zu ergänzen (siehe Listing 9).

Konfiguration der Zugangskontrolle

Aus Sicherheitsgründen empfiehlt es sich, die Nutzung eines Web-Service nur bestimmten Nutzern zu gestatten, die in der Zugangskontroll-Liste (engl.: Access Control List, kurz ACL) geführt werden. In der Datenbank 10g war die Liste beschränkt auf den User „PUBLIC“. Somit stand ein Web-Service allen Nutzern zur Verfügung. Mit der

Version 11g wurde dieses Konzept folgendermaßen verfeinert:

- Es können einzelne User in die Zugangsliste aufgenommen werden.
- Die ACL kann auf bestimmte Hosts eingeschränkt werden. Möglich ist die Angabe eines Host-Namens, einer Domäne, einer IP-Adresse oder eines Sub-Netzes. Beim Host-Namen ist die Groß- und Kleinschreibung zu beachten. Bei IP-Adressen und Domänen sind Wildcards möglich.
- Es gibt eine Beschränkung auf bestimmte Port-Bereiche.

```

-- SET XDB Port (HTTP)
exec DBMS_XDB.setHTTPPort(8893);

-- Ergebniskontrolle
select dbms_xdb.getHttpPort() as "HTTP-Port"
  from dual
;

```

Listing 6: Konfiguration Web-Service – Kommunikation über Port 8893

```

-- Listener ENDPOINT setzen
exec dbms_xdb.setListenerEndPoint(1,null,8080,1);
exec dbms_xdb.setListenerEndPoint(2,null,8893,2);

-- Ergebniskontrolle:
declare
  a1 varchar(2);
  b1 number;
  c1 number;
  a2 varchar(2);
  b2 number;
  c2 number;

BEGIN
  dbms_xdb.getlistenerendpoint(1, a1, b1, c1);
  dbms_xdb.getlistenerendpoint(2, a2, b2, c2);

  dbms_output.put_line(a1 || ' ' || b1 || ' ' || c1);
  dbms_output.put_line(a2 || ' ' || b2 || ' ' || c2);
END;
/

-- Check WS Ports
select extractValue(value(x),'httpconfig/http-protocol', 'xmlns="http://xmlns.oracle.com/xdb/xdbconfig.xsd"')
"HTTP Protocol"
, extractValue(value(x),'httpconfig/http-port',
'xmlns="http://xmlns.oracle.com/xdb/xdbconfig.xsd"') "HTTP Port#"
, extractValue(value(x),'httpconfig/http2-protocol', 'xmlns="http://xmlns.oracle.com/xdb/xdbconfig.xsd"')
"HTTPS Protocol"
, extractValue(value(x),'httpconfig/http2-port', 'xmlns="http://xmlns.oracle.com/xdb/xdbconfig.xsd"') "HT-
TPS Port#"
from
table(xmlsequence(extract(xdburitype('/xdbconfig.xml').getXML(),'/xdbconfig/sysconfig/protocolconfig/httpcon-
fig'))) x
;
```

Listing 7: Konfiguration Web-Service – Listener Endpoint

Listing 10 zeigt die gespeicherte Prozedur zum Anlegen einer ACL.

Sonstige Schritte

Wie bereits erwähnt wurde, muss bei Benutzung eines Wallet dessen Pfad in der ACL der Datenbank bekannt sein. Zur Bekanntgabe dient das in **Listing 11** gezeigte Kommando. Sobald dies erledigt ist, muss die XDB-Konfiguration aktualisiert werden (siehe **Listing 12**).

Aus Sicherheitsgründen sollte der Anbieter eines Web-Service nicht der User „SYS“ sein und über keine DBA-Privilegien verfügen. Vielmehr sollte ein geeigneter Datenbank-User ohne jegliche DBA-Privilegien angelegt werden. Sowohl Anbieter als auch Nutzer sind mit den in **Listing 13** gezeigten Privilegien auszustatten.

Sind die erforderlichen Privilegien zugewiesen, so steht dem User die Nutzung eines Web-Service über HTTPS zur Verfügung. Er hat dann den glei-

```

WALLET_LOCATION=
  (SOURCE=
    (METHOD=FILE)
    (METHOD_DATA=
      (DIRECTORY=<Pfad für Oracle-Wallet>)
    )
  )

SSL_CLIENT_AUTHENTICATION=FALSE

(DESCRIPTION=
  (ADDRESS = (PROTOCOL=tcp) (HOST=<hostname>)(PORT=8080))
  (Presentation=HTTP)(Session=RAW)
)
(DESCRIPTION=
  (ADDRESS = (PROTOCOL=tcps) (HOST=<hostname>)(PORT=8893))
  (Presentation=HTTPS)(Session=RAW)
)
```

Listing 8: Netzwerk-Konfiguration „listener.ora“

chen Zugang zu seinen Objekten wie bei einer manuellen Anmeldung, allerdings sind die PUBLIC-Objekte für den Web-Service gesperrt. Sollte der Web-Service des Users auch Zugang

zu PUBLIC-Objekten benötigen, so ist dies ab Version 11g gesondert zu veranlassen (siehe **Listing 13**). Falls nötig, kann zusätzlich ein Zugang über HTTP gewährt werden, siehe ebendort.

XML-DB-Web-Services nutzen

Zur Nutzung eines Web-Service mit Datenbank-Mitteln eignen sich PL/SQL-Prozeduren oder Packages. Damit kann der Konsument des Service automatisch einen SOAP-Request erstellen und an den Anbieter senden. Auch die SQL-Injektion wird mit Prozeduren beziehungsweise Packages ausgeschlossen.

```

WALLET_LOCATION=
  (SOURCE=
    (METHOD=FILE)
    (METHOD_DATA=
      (DIRECTORY=<Pfad für Oracle-Wallet>)
    )
  )

SSL_CLIENT_AUTHENTICATION=FALSE
SQLNET.AUTHENTICATION_SERVICES = (TCPS,NTS, BEQ) # add TCPS

```

Listing 9: Netzwerk-Konfiguration „sqlnet.ora“

```

-- ACL oraws.xml anlegen
begin
  if (dbms_db_version.ver_le_10_2) then
    dbms_output.put_line('Oracle-Version kleiner / gleich 10.2');
    dbms_output.put_line('keine ACL Konfiguration noetig');
  else
    dbms_output.put_line('ACL oraws.xml anlegen');
    dbms_network_acl_admin.create_acl ( acl          => 'oraws.xml'
                                       , description => 'ACL fuer Ora-WebService'
                                       , principal   => '<WS_USER>' -- Schema der Anwendung
                                       , is_grant    => true
                                       , privilege   => 'connect');

    dbms_output.put      ('- add privilege ...');
    dbms_network_acl_admin.add_privilege ( acl          => 'oraws.xml'
                                           , principal   => '<WS_USER>' -- Schema der Anwendung
                                           , is_grant    => true
                                           , privilege   => 'resolve');

    dbms_output.put_line(' OK');
    dbms_output.put      ('- assign acl ...');
    dbms_network_acl_admin.assign_acl ( acl          => 'oraws.xml'
                                       , host         => '<URL des
Datenbankservers>' -- z. B. ,10.1.10.*'
                                       , lower_port  => 8080
                                       , upper_port  => 8893);

    dbms_output.put_line(' OK');

    dbms_output.put_line('ACL oraws.xml fertig');
  end if;
end;
/

-- Ergebniskontrolle
SELECT dna.host
      , dna.lower_port
      , dna.upper_port
      , dna.acl
--      , dna.aclid
FROM DBA_NETWORK_ACLS dna;

-- ACL-Privileges checken
select dnac.accl
--      , dnac.aclid
      , dnac.principal
      , dnac.privilege
      , dnac.is_grant
--      , dnac.invert
      , dnac.start_date
      , dnac.end_date
from DBA_NETWORK_ACL_PRIVILEGES dnac;
select dwa.wallet_path
      , dwa.acl
--      , dwa.aclid
from dba_wallet_acls dwa;

```

Listing 10: Konfiguration Web-Service – ACL anlegen

```
-- ACL: Wallet-Lokation bekanntgeben
BEGIN
  DBMS_NETWORK_ACL_ADMIN.ASSIGN_WALLET_ACL ( acl => 'oraws.xml'
                                           , wallet_path => 'file:
<Pfad für Oracle-Wallet>' -- Wallet Path des Servers
                                           );
END;
/
```

Listing 11: Bekanntgabe des Wallet-Speicherorts

```
-- refresh XDB-Configuration ...
exec DBMS_XDB.CFG_REFRESH;
```

Listing 12: Auffrischung der XDB-Konfiguration

```
-- Zugang über HTTPS
GRANT XDB_WEBSERVICES TO <WS_USER>;
-- Zugang zu PUBLIC-Objekten
GRANT XDB_WEBSERVICES_WITH_PUBLIC TO <WS_USER>;
-- Zugang über HTTP (nicht nur HTTPS)
GRANT XDB_WEBSERVICES_OVER_HTTP TO <WS_USER>;
```

Listing 13: Zuweisung der Privilegien

Für die Anfrage wird ein XML-Dokument erstellt, das als SOAP-Request an den Anbieter gesendet werden kann.

Sein Format muss der WSDL (siehe oben) des Anbieters entsprechen. Anschließend wird eine Verbindung zum

Anbieter aufgebaut. Für vertrauensvolle Verbindungen muss der Speicherort des Wallet bekannt sein. Falls der Anbieter die Anfrage fehlerfrei verarbeiten konnte, sendet er eine Antwort zurück, die von der Prozedur abgeholt wird. Als Beispiel dient das in [Listing 14](#) gezeigte Package. Es steht aufgrund seiner Länge unter www.doag.org/go/doagnews/webservice zum Download bereit.



Detlev Schütte
detlev.schuette@orbit.de



DBMonipro: Die leistungsstarke, **kostenlose** Monitoring Lösung für **ALLE** Oracle Datenbank Editionen.

DBMonipro läuft via Nagios und enthält zahlreiche vorgefertigte Datenbank Checks. Neue Checks können in wenigen Minuten ohne Programmierkenntnisse erstellt werden.

Kostenloser Download: <http://www.dbmonipro.at>



Die Oracle Experten

Lizenz
Kosten

Praxis Seminar

Oracle Software Lizenzierung

✓ Klare Antworten

✓ Fehler vermeiden ✓ Kosten optimieren

✓ Zahlreiche Praxisbeispiele

Bei diesem eintägigen Seminar lernen Sie alle Stolpersteine eindeutig zu erkennen und zu vermeiden, sowie Einsparungspotentiale zu identifizieren und umzusetzen.



Ing. Klaus-Michael Hatzinger
CEO DBConcepts GesmbH, Oracle DB Lizenzierungs Experte

**25. März Düsseldorf, 26. März Frankfurt
27. März Stuttgart, 01. April Wien**

Shortlink zur Veranstaltung: <http://bit.ly/praxis-seminar>