

ORDIX news

Das IT-Magazin der ORDIX AG



Verwendung von „Binary XML“

6 | Oracle und XML - Vorstellung der neuen Speicherungs-methode

10 | PHAR - PHP Archive
Vorstellung der PHP-Erweiterung

22 | XEN - Alles sicher auf der Festplatte?
Festplattenverwaltung mit XEN

13 | Build Management und IDE
Maven und Eclipse in Java Projekten

44 | Java Server Faces 2.0
Neuheiten ab Java EE Version 6

Suchen Sie die Lösung?

Vortragsprogramm

Wir laden Sie herzlich zu unserer kostenfreien Veranstaltungsreihe „Treffpunkt RheinMainIT“ in unser Trainingszentrum Wiesbaden ein.

■ 07.09.2010

Virtualisierung mit VMware - Konzepte, Mehrwerte, Praxiseinsatz

Einblick in das Thema Virtualisierung, Darstellung von Konzepten und Erfahrungsbericht aus einem Großkundenprojekt.

Referent:
Christof Amelunxen, ORDIX AG

■ 09.11.2010

Projektmanagement ist Management

In welchen Rollen ist ein Projektmanager aktiv und welchen Anforderungen muss er gerecht werden?

Referent:
Benedikt Georgi, ORDIX AG

Von Gurken und Wildsäuen

Paderborn, Juni 2010

Das ist schon erstaunlich, da gewinnt ein angebliches Dreamteam eine Bundestagswahl (und auch noch mit akzeptabler Mehrheit) und macht sich noch nicht einmal durch notwendige unpopuläre Entscheidungen unbeliebt, wie das in der zweiten Legislaturperiode von Rot-Grün passierte, die die Agenda 2010 und die Hartz Gesetze durchsetzten. Nein, durch nichts tun, nichts entscheiden, ständiges Gezänk und unerträgliche Beschimpfungen, die man eher von einer Opposition denn einer Koalition erwarten würde, verliert die bürgerliche Regierung jede Woche Wählerstimmen.

Selbst ein WM-Gewinn der deutschen Nationalelf kann die gesamte Gurkentruppe eigentlich nicht mehr retten (O-Ton Seehofer, der das allerdings nur auf die FDP bezog).

Früher hätte man bei Gurken eher an schlechte Programme gedacht als an Politiker. Aber stattdessen wird die eine oder andere IT-Gurke zum Renner. Das iPad wirbelt alles auf. Jeder braucht es, aber keiner weiß so richtig warum, denn als Telefon ist es zu groß und eigentlich kann man gar nicht damit telefonieren. Wie schon beim iPhone oder früher Bill Gates bei Microsoft versteht es Steve Jobs Nullnummern oder angebliche Neuerungen, die es schon lang bei anderen Herstellern gibt, als geniale Neuerfindung des Apfels zu präsentieren.

Aber solche Nachrichten verblassen jetzt völlig, denn es ist WM. Da könnten nun Griechenland, Ungarn oder wer auch immer Pleite gehen. Wenn Fußball läuft, verschwindet alles andere von den Titelseiten und aus den Nachrichtensendungen. Vermutlich verpassen wir sogar, wenn der neue Bundespräsident gewählt wird, was vielleicht auch besser ist. Auch hier zeigt die Gurkentruppe an unserer Spitze wenig Fingerspitzengefühl. Machterhalt und -kalkül ist eben alles.

Aber warten wir mal ab, was in Südafrika nun wirklich passiert (der Auftakt war allerdings erfolgversprechend). Wer neben dem Fernsehen noch Zeit hat zu lesen, kann das mit dem iPad tun (ok, es gibt noch nicht so richtig viel) oder aber mit den ORDIX News.

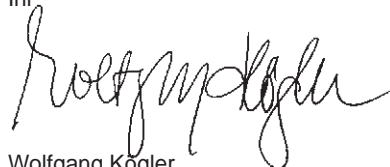
Wir bieten Interessantes von den Datenbanksystemen Oracle („Binary XML“ eine neue Variante, das Thema XML zu speichern, in den Griff zu bekommen, Security-Neuheiten und Testdatengenerierung aus Produktivdaten), DB2 (Perl und DB2 im Zusammenspiel) und Express (ent-)Laden mit externen Tabellen bei Informix. Wir entführen Sie in die virtuelle Welt von XEN und beschäftigen uns mit PHP-Entwicklung (PHAR Archive), Java Webservices (JSF 2.0) und Java Entwicklungswerkzeugen. Im zweiten Teil unserer Historie werfen wir einen Blick in die Vergangenheit und zum 20-jährigen Jubiläum gibt es dieses Mal ein bisschen weniger Technik und etwas mehr Bilder. Aber das darf auch mal sein.

Einige unserer Leser werden sich in den nächsten Wochen in den großen Urlaub begeben. Flugausfälle wegen Vulkanausbrüchen sind momentan nicht mehr zu erwarten. Wer mit dem Auto unterwegs ist, dem kann ich nur Zeitpunkte mit deutschen Spielen empfehlen, da sind die Autobahnen völlig frei.

Ich wünsche allen, die jetzt in den Urlaub starten, gute Erholung, allen anderen viel Spaß bei der Arbeit egal ob mit oder ohne Fußball. Ich bin gespannt, welche Gemüsesorten und Tiere in ca. drei Monaten en Vogue sind, wenn die nächste ORDIX News erscheint. Vielleicht entscheidet man sich aber auch noch kurzfristig für Jogi Löw als Kandidaten für den Posten des Bundespräsidenten, das Bundestraineramt will er ja aufgeben.

Bleiben Sie dran, ich informiere Sie rechtzeitig, wenn die erste ORDIX News fürs iPad erscheint.

Ihr

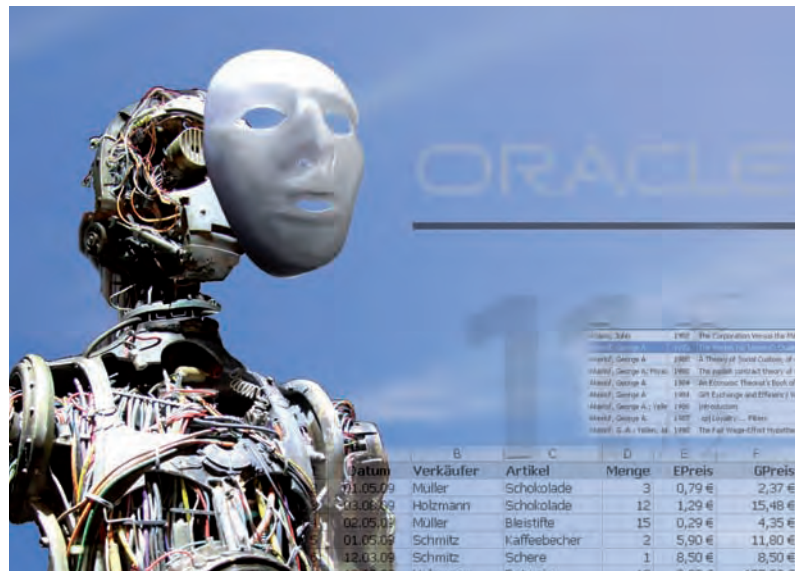


Wolfgang Kögler





6 | Verwendung von „Binary XML“



36 | Oracle Data Masking

Java/JEE

- 13..... **Maven und Eclipse in komplexen Java-Enterprise-Projekten - Build Management und IDE effizient einsetzen**
Mit Maven lässt sich der Entwicklungsprozess von großen Java Enterprise Projekten maßgeblich vereinfachen.
- 44..... **Java Server Faces 2.0 (JSF) - der neue Standard?**
In den meisten Java-Web-Entwicklungen ist JSF zu einer der wichtigsten Technologien geworden. Ab der kommenden Java Version (EE 6) ist JSF als Standard eingebunden.

Betriebssysteme

- 22..... **Server-Virtualisierung mit XEN (Teil II): Alles sicher auf der Festplatte?**
Dieser Artikel stellt die Festplattenverwaltung, die unterstützten Festplattenformate und die Backup-Möglichkeiten rund um XEN vor.

Open Source

- 10..... **PHP-Archive - Komplettpakete mit PHAR**
Vorstellung der PHP-Archive, die ein vereinfachtes Ausliefern von PHP-Anwendungen möglich macht. PHAR ist ab der PHP-Version 5.3 ein integrierter Bestandteil.

Datenbanken

- 6..... **Oracle und XML (Teil IV) Verwendung von „Binary XML“**
Lesen Sie, wie man XML-Dokumente in einer Oracle Datenbank abspeichern kann.
- 18..... **Oracle 11g Security (Teil II): Maßnahmen für eine erhöhte Passwortsicherheit**
Vorstellung der neuen Maßnahmen von Oracle, um Passwörter besser vor Hackern zu schützen.
- 36..... **Oracle Data Masking - Testdatengenerierung leicht gemacht**
Ab der Version 11g existieren zwei Varianten der einfachen Testdatengenerierung aus Produktivdaten: Oracle Data Masking und Oracle Datapump.
- 40..... **Perl und DB2 im Verbund (Teil II): Übersicht durch Ordnung**
Dieser Artikel erläutert wie man mit Perl in der View List Application Show Detail aufräumt.
- 50..... **Informix IDS (Teil II): External Tables Wie der Blitz raus und rein**
Ein interessantes neues Leistungsmerkmal birgt der Sprung von xC5 auf xC6, obwohl es für Kenner des IBM DBMS gar nicht so neu sein dürfte.





40 | Perl und DB2 im Verbund - Übersicht durch Ordnung



44 | Java Server Faces 2.0 - der neue Standard?

Aktuell

- 28..... **20 Jahre ORDIX (Teil II): 2000 - 2010.**
- 30..... **20 Jahre Qualität - einfach. gut. gefeiert.**
Impressionen der Jubiläumsfeier.
- 35..... **Rückblick auf das DOAG Regionaltreffen und den IUG Workshop**
- 39..... **Larry Ratlos: Larry hat ein PHP-Problem mit der Select-Box**

Standards

- 03..... **Editorial**
- 04..... **Inhalt**
- 05..... **Impressum**
- 26..... **Seminarübersicht: Juni bis November 2010**

Impressum

Herausgeber:

ORDIX AG Aktiengesellschaft für Softwareentwicklung, Beratung, Schulung und Systemintegration, Paderborn

Redaktion: Jens Pothmann

V.i.S.d.P.: Benedikt Georgi, Wolfgang Kögler

Anschrift der Redaktion:

ORDIX AG
Westernmauer 12 - 16
33098 Paderborn
Tel.: 05251 1063-0
Fax: 0180 1673490

Gestaltung/Layout:

Jens Pothmann, Stefanie Heither-Bürger

Auflage: 9.900

Druck: Media-Print Informationstechnologie GmbH, Paderborn

Bildnachweis:

© sxc.hu / Unique © Clix
© sxc.hu / Laptop eyes © Jef Bettens
© aboutpixel.de / Buchstaben #2 © Uwe Dreßler
© sxc.hu / Masks
© shutterstock.com / The big old wooden catalogue © ID1974
© shutterstock.com / Closeup portrait of a happy © Yuri Arcurs

Autoren dieser Ausgabe:

ChristianFertsch, DennisFörster, KlausGünther, KathrinHammerschmidt, Patrick Hecker, Stefanie Heither-Bürger, Kathleen Hock, Oliver Kaluza, Wolfgang Kögler, Björn Konrad, Lars Hendrik Korte, Norbert Munkel, Ulf Papenfuß, Michael Skowasch

Die Zeitschrift ORDIX News wird von der ORDIX AG an ausgewählte Kunden verteilt und kann für 2,20 Euro bestellt werden. Sie können die Zusendung der ORDIX News jederzeit ohne Angabe von Gründen schriftlich (z. B. Brief, Fax, E-Mail) abbestellen. Die neueste Ausgabe wie auch ältere Ausgaben finden Sie im Archiv der ORDIX News im Internet unter: <http://www.ordix.de>. Schauen Sie mal rein!

Der Kontakt zu unseren Lesern ist uns sehr wichtig. Für Anregungen, Kritik und Anmerkungen zu den Themen, aber auch für interessante Ideen sind wir immer offen und dankbar. Wir freuen uns auf Ihr Feedback an redaktion@ordix.de.

Copyright:

ORDIX AG. Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung der Artikel oder von Teilen daraus, bleiben uns vorbehalten. Kein Teil der Artikel darf ohne unsere schriftliche Genehmigung in irgendeiner Form reproduziert, insbesondere unter Verwendung elektronischer Systeme verarbeitet, verbreitet, vervielfältigt oder zu öffentlichen Wiedergaben benutzt werden.

Haftung:

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

Warenzeichen:

Einige der aufgeführten Bezeichnungen sind eingetragene Warenzeichen ihrer jeweiligen Inhaber. ORDIX® ist registrierte Marke der ORDIX AG.



Oracle und XML: Datentyp XMLType (Teil IV)

Verwendung von „Binary XML“

Dieser Artikel richtet sich an Datenbankentwickler und Software-Architekten, die XML im Zusammenspiel mit Oracle-Datenbanken einsetzen möchten.

Seit der Oracle-Version 9i steht XMLType als neuer Datentyp in der Datenbank zur Verfügung. In der Version Oracle 11g gibt es darüber hinaus die neue Speicherungs- methode „Binary XML“ für XMLType-Daten. Diese Speicherungsform und ihre Verwen- dung werden wir in diesem Artikel vorstellen.

Speichermethoden bisher

XMLType-Daten konnten bisher auf zwei ver- schiedene Arten gespeichert werden:

- in CLOBs (unstrukturiert)
In der Datenbank wird ein verstecktes CLOB angelegt. Das XML-Dokument wird als Ganzes darin gespeichert. Wurde beim Anlegen einer Spalte vom Datentyp XMLType kein XML-Schema angegeben, so wird automatisch ein CLOB erzeugt.
- in Tabellen und Views (strukturiert)
Diese Art der Speicherung ist nur möglich, wenn das passende XML-Schema mit Hil- fe des Packages DBMS_XMLSCHEMA in der Datenbank registriert wurde. In die- sem Fall werden aufgrund der Informa- tionen aus dem XML-Schema die ent- sprechenden objektrelationalen Tabellen oder Views erzeugt.

Bei der strukturierten Speicherung ist das Abspeichern der XML-Dokumente und der Zugriff auf die Dokumente wesentlich per-

formanter, da nicht mehr das ganze XML-Do- kument gelesen werden muss. Zudem kön- nen Indizes bei der strukturierten Speicherung für eine bessere Performance sorgen. Darüber hinaus wird weniger Speicherplatz benötigt, da nicht zu jedem in der XMLType-Spalte ab- gelegten XML-Dokument jeder Tag-Name ab- gespeichert werden muss.

Binary XML

Binary XML ist eine neue Speicherungsme- thode für XMLType-Daten. Dabei soll diese Speicherungsform die bisherigen nicht ablö- sen, sondern ergänzen. Die Speicherungsme- thode Binary XML speichert XML-Dokumente in einem speziell für XML-Daten entworfenen Binärformat in der Datenbank.

Ressourcen

Die Verwendung von Binary XML führt zu ei- ner effizienteren Speicher-, CPU- und Netz- werkauslastung. Da die XML-Dokumente in speziellen Binärformat in der Datenbank ab-

gelegt werden, wird der Speicherplatz, der für das Speichern der XML-Dokumente benötigt wird, reduziert. Es ist nicht mehr nötig, andere Kompressionstechniken, wie z. B. das Zippen von Dokumenten, zu verwenden. Die Speicherung als Binary XML sorgt für eine effektivere Speicher- und CPU-Auslastung, da das Parsen und Serialisieren der XML-Daten beim Austausch der Daten zwischen den einzelnen Applikationsschichten bzw. vor dem Speichern in der Datenbank wegfällt. Bei der Verwendung von Binary XML als Speichermethode für XML-Dokumente wird der Netzwerk-Overhead verringert, da für die Übertragung das kompakte, interne Format verwendet wird und nicht das traditionelle serialisierte Textformat.

Nachteilig ist dabei allerdings, dass die XML-Dokumente nicht mehr ohne weiteres editiert werden können. Hier müssen dann die entsprechenden Funktionen des Datentyps XMLType verwendet werden. Weitere Informationen über XMLType finden Sie in dem ORDIX News Artikel der Ausgabe 1/2007 [1].

Streaming XPath Evaluierung

Nutzt man die Speichermethode Binary XML, so kann man Knoten und Dokumentteile effizient aus dem XML-Dokument extrahieren. Das Format Binary XML erlaubt Streaming XPath Evaluierung, womit mehrere Knoten oder mehrere Teile des XML-Dokuments in einem einzigen Dokumentdurchlauf extrahiert werden können.

Neuer Parser / XML-Schema-Validator

Für die Speichermethode Binary XML liefert Oracle einen neuen Parser und einen neuen XML-Schema-Validator mit. Weder der Parser noch der Validator nutzen einen DOM-Baum zum Parsen bzw. Validieren der Daten. Dies führt zu einer geringeren CPU-Auslastung, weil der DOM-Baum immer im Hauptspeicher aufgebaut wird. Zudem wird der Overhead beim Update von XML-Dokumenten verringert, da nach einem Update nicht mehr das komplette XML-Dokument sondern nur der aktualisierte Teil des Dokuments neu validiert wird.

Flexibler XML-Schema-Support

Binary XML bietet einen flexiblen XML-Schema-Support. So können beim Anlegen einer Tabelle verschiedene Encodings angegeben werden. Über diese Encodings wird geregelt, ob

```
BEGIN
  DBMS_XMLSCHEMA.registerSchema(
    SCHEMAURL => 'http://www.ordix.de/binary_xml.xsd',
    SCHEMADOC => '<XMLSCHEMA>',
    LOCAL      => FALSE,
    GENTYPES   => FALSE,
    GENBEAN    => FALSE,
    GENTABLES  => FALSE,
    FORCE       => FALSE,
    OPTIONS    => DBMS_XMLSCHEMA.REGISTER_BINARYXML,
    OWNER      => USER);
END;
```

Abb. 1: Registrierung des XML-Schemas.

die zu speichernden XML-Dokumente einem XML-Schema genügen müssen oder nicht. Im Gegensatz zur strukturierten Speicherung des XMLType kann hier aber nicht nur ein Schema angegeben werden, das in der gesamten Tabelle verwendet wird. Es können mehrere XML-Schemata angegeben werden, denen die zu speichernden XML-Dokumente genügen müssen.

Standard W3C XLIFF

Binary XML unterstützt den Standard W3C XLIFF. Wenn das Dokument sprachspezifische Inhalte enthält, so wird beim Zugriff auf das XML-Dokument nur der Inhalt zurückgeliefert, der zu den entsprechenden Locale-Einstellungen passt.

Verwendung

Nachdem wir erfahren haben, was Binary XML ausmacht, wenden wir uns nun der eigentlichen Verwendung dieser Speicherungsform zu. Dabei wird vor allem auf die einzelnen Encoding-Optionen dieser Speicherungsform eingegangen.

Registrierung des XML-Schemas

Damit ein XML-Schema für das Encoding einer XMLType-Spalte, die die Speicherungsform Binary XML verwendet, eingesetzt werden kann, muss es zunächst in der Datenbank registriert werden. Es reicht dabei nicht, das Schema auf dem herkömmlichen Weg zu registrieren. Es muss die Option `DBMS_XMLSCHEMA.REGISTER_BINARYXML` mitgegeben werden. Erst dann kann das XML-Schema beim Anlegen von Tabellen und beim Einfügen von Werten verwendet werden (siehe Abbildung 1).

```
CREATE TABLE binary_xml_test1(
id NUMBER(2),
doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS BINARY XML;
```

Abb. 2: STORE AS BINARY XML.

```
CREATE TABLE binary_xml_test2(
id NUMBER(2),
doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS BINARY XML
XMLSCHEMA "http://www.ordix.de/binary_xml.xsd"
ELEMENT "Auto";
```

Abb. 3: STORE AS BINARY XML XMLSCHEMA.

```
CREATE TABLE binary_xml_test3(
id NUMBER(2),
doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS BINARY XML
XMLSCHEMA "http://www.ordix.de/binary_xml.xsd"
ELEMENT "Auto" ALLOW NONSCHEMA;
```

Abb. 4: STORE AS BINARY XML XMLSCHEMA ... ALLOW NONSCHEMA.

```
CREATE TABLE binary_xml_test4(
id NUMBER(2),
doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS BINARY XML
ALLOW ANYSCHEMA;
```

Abb. 5: STORE AS BINARY XML ALLOW ANYSCHEMA.

```
CREATE TABLE binary_xml_test5(
id NUMBER(2),
doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS BINARY XML
ALLOW ANYSCHEMA ALLOW NONSCHEMA;
```

Abb. 6: STORE AS BINARY XML ALLOW ANYSCHEMA ALLOW NONSCHEMA.

```
INSERT INTO binary_xml_test2
VALUES (1,XMLType('<?xml version="1.0" encoding="UTF-8"?>
<Auto xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.ordix.de/
binary_xml.xsd">
  <Herstellerliste>
    <Hersteller id="1">
      <Name>Motorenwerke R&#252;desheim</Name>
    </Hersteller>
  </Herstellerliste>
</Auto>'));
```

Abb. 7: Einfügen von Werten.

Wird das XML-Schema auf dem herkömmlichen Weg (`DBMS_XMLSCHEMA.registerSchema('http://www.ordix.de/binary_xml.xsd',xml_schema);`) registriert, so kommt es beim Anlegen einer Tabelle mit einer XMLType-Spalte mit der Speicherungsform Binary XML, zu folgendem Fehler:

FEHLER in Zeile 1: ORA-44424: BINARY-XML-Speicher erfordert, dass XML-Schema für BINARY-Gebrauch registriert ist.

Um den Fehler zu beheben, muss das XML-Schema, wie in Abbildung 1 beschrieben, registriert werden.

Anlegen einer Tabelle

Beim Anlegen von Tabellen mit XMLType-Spalten gibt man im Storage Clause an, welche Speicherungsform verwendet werden soll. Wird die Speicherungsform nicht angegeben, so nimmt die Datenbank die textbasierte Speicherung. Soll die Speicherungsform Binary XML verwendet werden, so gibt man im Storage Clause die Option „STORE AS BINARY XML“ an. Zusätzlich zu dieser Option kann auch noch angegeben werden, ob und wie ein XML-Schema verwendet werden soll. Dazu muss beim Anlegen der Tabelle das Encoding mit angegeben werden. Die einzelnen Optionen werden in den folgenden Absätzen beschrieben.

STORE AS BINARY XML

Beim Speichern von XML-Dokumenten in einer mit diesem Encoding angelegten Spalte, wird kein XML-Schema verwendet. Die XML-Dokumente werden beim Insert und beim Update nicht gegen ein XML-Schema validiert. Wenn im XML-Dokument ein XML-Schema angegeben worden ist, wird dieses ignoriert (siehe Abbildung 2).

STORE AS BINARY XML XMLSCHEMA

Diese Option wird verwendet, wenn das zu speichernde XML-Dokument einem einzelnen XML-Schema entsprechen soll. Das entsprechende `CREATE TABLE` Statement ist in Abbildung 3 dargestellt. Dieses Schema muss zunächst, wie oben beschrieben, in der Datenbank registriert werden. Möchte man in eine solche Spalte XML-Dokumente einfügen, die nicht dem registrierten XML-Schema entsprechen, so kommt es zu einem Fehler.

STORE AS BINARY XML XMLSCHEMA ... ALLOW NONSCHEMA

Beim Speichern von XML-Dokumenten in einer mit diesem Encoding angelegten Spalte, wird für alle XML-Schema-basierten XML-Dokumente ein auf dem referenzierten Schema basierendes Encoding verwendet. Für alle nicht-XML-Schema-basierten XML-Dokumente wird ein nicht-schema-basierendes Encoding verwendet. Die Abbildung 4 zeigt das entsprechende **CREATE TABLE** Statement. Das angegebene Schema muss zunächst, wie in Abbildung 1 beschrieben, in der Datenbank registriert werden. Versucht man in eine Spalte ein XML-Dokument einzufügen, welches ein XML-Schema referenziert, diesem aber nicht entspricht, kommt es zu einem Fehler. Versucht man in eine Spalte ein XML-Dokument einzufügen, das kein XML-Schema referenziert, kommt es nicht zu Fehlern.

STORE AS BINARY XML ALLOW ANYSCHEMA

Beim Speichern von XML-Dokumenten in einer mit diesem Encoding angelegten Spalte, wird für alle XML-Dokumente ein auf dem referenzierten XML-Schema basierendes Encoding verwendet. Es wird jeweils das XML-Schema verwendet, das im XML-Dokument referenziert wird. Abbildung 5 zeigt das entsprechende **CREATE TABLE** Statement. Das angegebene Schema muss zunächst in der Datenbank registriert werden. Versucht man in eine solche Spalte ein XML-Dokument einzufügen, das kein XML-Schema referenziert oder das nicht dem referenzierten XML-Schema entspricht, so kommt es zu einem Fehler.

STORE AS BINARY XML ALLOW ANYSCHEMA ALLOW NONSCHEMA

Beim Speichern von XML-Dokumenten in einer mit diesem Encoding angelegten Spalte, wird für alle XML-Schema-basierten XML-Dokumente ein auf dem referenzierten Schema basierendes Encoding verwendet. Es wird jeweils das XML-Schema verwendet, das im XML-Dokument referenziert wird. Das angegebene Schema muss zunächst an der Datenbank registriert werden. Für alle nicht-XML-Schema-basierten XML-Dokumente wird ein nicht-schema-basierendes Encoding verwendet. Das entsprechende **CREATE TABLE** Statement ist in Abbildung 6 dargestellt.

Glossar

DOM	Document Object Model. DOM ist ein API für den Zugriff auf XML-Dokumente. Das API erlaubt, dynamisch die Struktur, das Layout und den Inhalt von XML-Dokumenten zu ändern.
XML	Extensible Markup Language (XML). XML ist eine so genannte META-Sprache zur Beschreibung von Dokumenten. Ein Vorteil von XML ist der vereinfachte Austausch von Daten, da XML-Formate in einer strengen Grammatik definiert werden können und so die Implementierung von zuverlässigen Schnittstellen erlauben.
XML-Schema	Ein XML-Schema beschreibt die Struktur von XML-Dokumenten und erlaubt ihre inhaltliche Überprüfung.
XMLType	XMLType ist ein Oracle Datentyp, durch den SQL-Operationen auf XML-Inhalte genauso ermöglicht werden wie XML-Operationen auf SQL-Inhalte.
XPath	XPath stellt Funktionen und Ausdrücke zur Verfügung, um Knoten innerhalb von XML-Dokumenten zu lokalisieren. Mit XPath können auch Ausdrücke ausgewertet und Berechnungen durchgeführt werden. Es dient als Grundlage für XSLT, XPointer und Xquery.

Links

- ▶ [1] ORDIX News Artikel „XMLType“:
http://www.ordix.de/ORDIXNews/1_2007/Datenbanken/xmltype.html
- ▶ [2] Seminarempfehlung „Oracle und XML“
<http://training.ordix.de/siteengine/action/load/kategorie/Programmierung/nr/271/index.html>

Einfügen von Werten

Wenn nun ein XML-Dokument in die Tabelle eingefügt werden soll (siehe Abbildung 7), muss dieses XML-Dokument nicht nur wie bisher wellformed sondern auch valid sein, d. h. es muss dem registrierten XML-Schema entsprechen. Ansonsten kann das XML-Dokument nicht in die Datenbank eingefügt werden und es wird ein Fehler ausgegeben.

Fazit

Die neue Speicherungsform Binary XML bietet sehr viele Möglichkeiten, XML-Dokumente in der Datenbank abzuspeichern. Der flexible Schema-Support sorgt dafür, dass komplett unterschiedliche XML-Dokumente in einer Spalte abgespeichert werden können und trotzdem gegen ein Schema validiert werden. Doch auch wenn kein XML-Schema verwendet wird, bietet die neue Speicherungsform durch die geringere Nutzung von Ressourcen große Vorteile.

Wann welche Speicherungsform einsetzen?

Nun gibt es drei Speicherungsformen für den Datentyp XMLType. Da stellt sich die Frage, welche Speicherungsform wann eingesetzt werden sollte. Diese Frage werden wir im nächsten Teil dieser Reihe beleuchten.



Kathrin Hammerschmidt
(info@ordix.de).

PHP-Archive – Komplettpakete mit PHAR

Dieser Artikel richtet sich an PHP-Entwickler, die ihre Anwendungen gerne als Komplettpaket ausliefern möchten und nicht als „lose“ Sammlung von Dateien.

PHAR (PHP-Archive) hat sich von einem PEAR-Paket hoch zu einem integrierten Bestandteil von PHP 5.3 entwickelt. Als standardmäßig aktivierte PHP-Erweiterung ermöglicht sie den Zugriff und die Verarbeitung von Dateien im komprimierten PHAR-Format. Es können dadurch z. B. komplexe (Web-)Anwendungen, die als PHAR vorliegen, direkt aus diesem Archiv gestartet werden.

Einleitung

PHP-Anwendungen bestehen nicht mehr nur aus wenigen HTML-Seiten, einigen PHP-Skripten und einer Datenbank. Insbesondere im Enterprise-Umfeld finden sich Anwendungen mit mehreren hundert Klassen-, Template-, HTML- und XML-Dateien. Beim Ausrollen einer derartigen Anwendung muss dann natürlich darauf geachtet werden, dass keine Datei „verloren“ geht. Dazu gibt es sicherlich verschiedene Automatismen. Aber wäre es nicht viel einfacher, alle für die Laufzeit notwendigen Dateien in nur einer Archiv-Datei zu bündeln und dann zu verteilen?

Mit der PHAR-Erweiterung für PHP stellt dies kein Problem mehr dar. Es handelt sich hierbei nicht um eine komplett neue Errungenschaft, denn schon seit 2005 gibt es das PEAR-Paket PHP-Archive, was sich zum Ziel gesetzt hat, PHP-Anwendungen gebündelt und in komprimierter Form in nur einer Datei auszuliefern. Jedoch sind in PHP programmierte Erweiterungen selten so schnell, wie in C programmierte. Daher wurde in 2007 mit der Entwicklung der PECL-Extension `phar` begonnen. Mittlerweile ist diese Extension fest in PHP 5.3 integriert und aktiviert. In diesem Artikel wird die Benutzung dieser Erweiterung beschrieben.

Erstellung eines PHAR

Zunächst muss sichergestellt werden, dass der `php.ini`-Parameter `phar.readonly` auf

`off` gesetzt ist, um schreibenden Zugriff auf PHARs zu ermöglichen. Dieser Parameter ist standardmäßig auf `on` gestellt und muss umgesetzt werden. Die Erzeugung eines PHAR geschieht über die Klasse `Phar` und ist in Abbildung 1 zu sehen. Mit nur zwei Code-Zeilen wird aus dem Inhalt des Verzeichnisses `app1` eine PHAR-Datei erzeugt. Diese kann nur mit PHP-Mitteln verändert oder wieder dekomprimiert werden. Mit dem Aufruf der Methode `convertToExecutable(Phar::ZIP)` auf das PHAR-Objekt kann diese in ein komprimiertes Format umgewandelt werden. Diese Datei kann dann mit einem beliebigen ZIP-Programm (z. B. WinZIP) bearbeitet und eingesehen werden. Eine Verarbeitung müsste dann aber mit Mitteln der ZIP-Extension gehandhabt werden.

Einlesen eines PHAR

Liegt bereits ein PHP-Archiv vor, so kann es zur Benutzung eingelesen werden. Dazu wird wieder ein Objekt vom Typ `Phar` erzeugt und dem Konstruktor werden diesmal drei Parameter übergeben. Der erste Parameter ist wieder der Pfad zur Datei. Als zweiter Parameter wird 0 übergeben. Im dritten Parameter definieren wir einen Alias-Namen. Mit diesem sprechen wir später das PHAR über den Streamwrapper `phar://` an, der von der PHAR-Extension neu zur Verfügung gestellt wird. Auf diese Weise können einzelne Dateien direkt aus dem Archiv in einen bestehenden PHP-Code eingebunden und benutzt werden, so als ob diese im Dateisystem

liegen würden. Dabei findet jedoch keine Dekomprimierung in ein temporäres Verzeichnis statt, sondern das PHAR bleibt unberührt. In Abbildung 2 ist ein Beispiel für das beschriebene Vorgehen zu sehen.

Ein kleiner Stub's

Im PHP-Archiv kann eine `stub`-Datei eingefügt werden. Der Code dieses Stub wird automatisch ausgeführt, wenn das PHAR über die Kommandozeile gestartet oder über einen Webserver aufgerufen wird, was eine Anpassung der Webserver-Konfiguration für PHP voraussetzt. Bisher wurde noch keine spezielle `stub`-Datei eingefügt, von daher hat PHP beim Erstellen des PHAR selbst einen minimalen `stub` generiert, der nur aus der Anweisung `__HALT_COMPILER()` besteht. Sie ist zwingend erforderlich, um PHP daran zu hindern, die komprimierten Binärdaten des eigentlichen PHP-Archivs die nach dem Stub Code folgen, als PHP Code zu interpretieren. Auch bei selbst geschriebenen Stubs muss dies unbedingt als letzte Anweisung vorhanden sein. Abbildung 3 zeigt, wie ein PHAR mit einem eigenen `stub` erstellt wird.

PHP kann einen Standard-Stub erzeugen (siehe Abbildung 4). Hier wird ein PHAR erstellt, der sich an der Kommandozeile und über Webserver gestartet anders verhält. Je nachdem wird eine unterschiedliche „Startseite“ aufgerufen. In diesem Fall entweder `cli.php`, wenn der Aufruf über die Kommandozeile erfolgt, oder `web.php` beim Aufruf über den Webserver. Die Methode `createDefaultStub()` bekommt als ersten Parameter den Dateinamen für den Kommandozeilenstart und als zweiten Parameter den Dateinamen für den Aufruf per Web übergeben. Auch die Methode `setDefaultStub()` kann zu diesem Zweck genutzt werden. Der `Default-stub` hat den Vorteil, dass er den Aufrufmechanismus auswerten kann. Allerdings ist relativ viel Code im `stub` vorhanden, was das PHAR etwas aufbläht. An dieser Stelle lohnt sich ein Blick in die PHAR-Datei mit einem Editor.

Sicherheitsaspekte

Eine Signatur innerhalb eines PHAR sichert es gegen Veränderungen außerhalb von PHP ab. Somit kann ein PHAR nicht mit einem Editor verändert werden, um z. B. das Verhalten des Stubs anzupassen. Wird dies dennoch getan, so kann das Archiv nicht korrekt eingelesen und benutzt werden.

```
$p = new Phar('appl.phar');
$p->buildFromDirectory('appl');
```

Abb. 1: Erzeugen eines PHAR.

```
$p = new Phar('appl.phar', 0, 'alias');
print file_get_contents('phar://alias/test.php');
include 'phar://alias/test.php';
```

Abb. 2: Einlesen und Benutzen eines PHAR.

```
$p = new Phar('appl.phar');
$p->buildFromDirectory('appl');
$p->setStub('<?php
Phar::mapPhar(\'alias\');
include \'phar://alias/test.php\';
__HALT_COMPILER(); ?>
');
```

Abb. 3: Erstellen eines PHAR mit eigenem Stub.

```
$r = new Phar('appl.phar');
$r->buildFromDirectory('appl');
$r->setStub($r->createDefaultStub('cli.php', 'web.php'));
```

Abb. 4: Erstellen eines PHAR mit PHP-generiertem Default-Stub.

```
$content = file_get_contents('cakey.pem');
$private = openssl_get_privatekey($content);
$pkey = '';
openssl_pkey_export($private, $pkey);
$p->setSignatureAlgorithm(Phar::OPENSSL, $pkey);
```

Abb. 5: Signieren eines PHAR mit OpenSSL.

PHP erzeugt beim Erstellen eines PHAR immer eine Signatur, die mit der Methode `getSignature()` ausgelesen wird. Aktuell können fünf Signaturtypen genutzt werden, die über `getSupportedSignatures()` ermittelt werden können:

- MD5
- SHA-1
- SHA-256
- SHA-512
- OpenSSL

OpenSSL stellt hierbei sicherlich die interessanteste Methode dar. Das PHAR wird mit einem privaten Schlüssel signiert und kann nur mit dem zugehörigen öffentlichen Schlüssel geöffnet werden. Hierzu wird der private Schlüssel zunächst mit PHP-OpenSSL-Funktionen eingelesen, bearbeitet und anschließend

Links

- ▶ [1] PHAR
<http://www.php.net/PHAR/>
- ▶ [2] PHP_Archive (PEAR Paket)
http://pear.php.net/package/PHP_Archive
- ▶ [3] PHAR-PECL-Extension
<http://pecl.php.net/package/phar>
- ▶ [4] OpenSSL
<http://www.openssl.org/>



Lars Hendrik Korte
(info@ordix.de).

der Methode `setSignatureAlgorithm()` mit dem Signatortyp `Phar::OPENSSL` übergeben, wie in Abbildung 5 zu sehen. Das so signierte PHAR kann jetzt nur eingelesen werden, wenn der öffentliche Schlüssel in einer Datei parallel zum PHAR abgelegt wird und die Endung `.pubkey` hat. Liegt das Archiv also unter `/pfad/zu/mein.phar`, so muss der Dateiname des öffentlichen Schlüssels `/pfad/zu/mein.phar.pubkey` lauten.

„PHAR-zit“

Eine komplette Anwendung in einer einzigen Datei auszuliefern, ist sehr nützlich und bereits in der Java-Welt gang und gäbe. Nicht umsonst ähneln sich die beiden Abkürzungen PHAR (PHP-Archiv) und JAR (Java-Archiv) in gewisser Weise. Ein PHAR kann direkt von der Kommandozeile oder über einen Webserver ausgeführt werden, ohne dass die darin befindlichen Dateien entpackt werden müssen. Zum Aufrufen eines PHAR per Browser muss jedoch die Webserverkonfiguration entsprechend angepasst werden, so dass `*.phar`-Dateien durch PHP interpretiert werden.

Die Möglichkeit, PHP-Archive mittels OpenSSL zu signieren, erhöht die Vertrauenswürdigkeit und auch die Sicherheit bei der Handhabung. Mit der festen Integration der PHAR-Extension in PHP-Distributionen ab Version 5.3.0 ist PHP wieder einen weiteren Schritt nach vorne gegangen, um im Vergleich mit anderen Programmiersprachen mithalten zu können.

Seminarempfehlung: PHP Programmierung Aufbau

- ▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Programmierung/nr/445/index.html>

In diesem Seminar lernen Sie die erweiterten Technologien von PHP kennen und anwenden. Hierbei wird der Schwerpunkt auf die objektorientierte Programmierung gelegt, mit deren Hilfe sich in PHP 5 modulare und flexible Web-Applikationen entwickeln lassen.

Seminarinhalte

- Einführung in die objektorientierte Entwicklung (OO-Entwicklung) mit PHP
- Unterschiede zwischen PHP 4 und PHP 5
- Klassen und Objekte
- SimpleXML
- MySQLi und Transaktionshandling
- Sicherheit von PHP Web-Anwendungen
- Session Management
- Funktionserweiterungen
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

23.08. - 25.08.2010 in Wiesbaden
18.10. - 20.10.2010 in Wiesbaden

Seminar-ID: P-PHP-02

Dauer: 3 Tage

Preis pro Teilnehmer: 1.190,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.071,00 € (zzgl. MwSt.)

Wir führen unsere Seminare auch jederzeit an einem geeigneten Ort Ihrer Wahl durch und bringen, wenn nötig, auch das entsprechende Equipment mit. Informieren Sie sich am besten im Internet über unsere Kundenseminare und die mobilen Schulungen: <http://training.ordix.de>.

Maven und Eclipse in komplexen Java-Enterprise-Projekten

Build Management und IDE effizient einsetzen

Die Beliebtheit von Maven wächst immer mehr. Kaum ein Java-Projekt möchte auf die Vorteile des mächtigen Build-Werkzeuges verzichten. Die wesentlichen Erfolgsfaktoren von Maven sind ein vorgegebener Build Lifecycle, Dependency Management, IDE-Unterstützung und Build Profile. Der Artikel zeigt, wie diese Faktoren den Entwicklungsprozess von größeren Java-Enterprise-Projekten, die häufig aus vielen parallel zu entwickelnden Komponenten bestehen, maßgeblich vereinfachen und beschleunigen können.

Dieser Artikel richtet sich an JEE-Entwickler, die Build Management und IDE in Ihren Projekten einsetzen möchten.

Einleitung

Java-Enterprise-Projekte bestehen typischerweise aus vielen einzelnen Komponenten, die in eigenständigen Software-Projekten parallel entwickelt werden. Innerhalb der Komponenten werden Verantwortlichkeiten in getrennten Schichten implementiert, die oftmals ebenso in eigene Subprojekte ausgelagert sind. Eine typische Geschäftsanwendung könnte beispielsweise aus einem Swing Client, einem Web-Modul, zwei EJB-Modulen und einer Datenbank bestehen. Jedes Submodul hätte hier einen individuellen Build-Prozess. Bei größeren Entwicklungsteams und speziellen technischen Anforderung bewirkt eine Trennung der Build-Prozesse einen wesentlich effizienteren Gesamtentwicklungsprozess.

Enterprise-Projekte mit vielen unterschiedlichen Modulen werden jedoch schnell komplex und lassen die Anforderungen an das Build Management steigen. Neben klassischen Build-Aufgaben muss sichergestellt werden, dass Konfigurationen möglichst nicht redundant definiert werden und das sich aus allen Submodulen das Gesamtsystem erzeugen lässt. Weiterhin müssen sämtliche Module einfach und effizient in einer IDE zusammenspielen. Außerdem sollten die komplexen Laufzeitabhängigkeiten zwischen den einzelnen Modulen automatisch aufgelöst werden können. Bezogen auf obiges Beispiel, in dem die Präsentationsschicht des Web-Moduls auf die Businesslogik der EJB-Module zugreift, muss der Build-Prozess die

abhängigen EJB-JAR-Dateien im Klassenpfad des Web-Moduls verfügbar machen.

Multi-Module

Multi-Module in Maven sind die Antwort auf diese Problemstellung. Hierunter versteht man eine Komposition aus verschiedenen Submodulen, die zusammen das Gesamtprojekt ergeben. Super- und Submodule haben jeweils einen eigenen Build-Prozess. Wird ein Build auf Supermodulebene gestartet, führt Maven diesen ebenfalls für alle Submodule aus. Abhängigkeiten zwischen den einzelnen Modulen des Gesamtprojekts können wie jede andere Abhängigkeit (Dependency) definiert werden. Bezogen auf das Beispiel würden wir eine Abhängigkeit von dem Web-Modul zum EJB-Modul definieren (siehe Abbildung 1). Der Build-Prozess des Multi-Modul-Projekts sorgt nun dafür, dass das abhängige JAR-Archiv automatisch dem Build-Prozess unseres Web-Moduls zur Verfügung steht.

Vererbung

Neben den Multi-Modul-Funktionalitäten tragen Vererbungsbeziehungen zu weiteren Vereinfachungen bei. Hierdurch lassen sich Elemente eines Elternprojekts an verschiedene Kindprojekte weitervererben. Abhängigkeiten und Konfigurationen, die für zwei Projekte gleichermaßen gelten (beispielsweise zwei Web-Module, die das gleiche Web-Framework nutzen), wer-

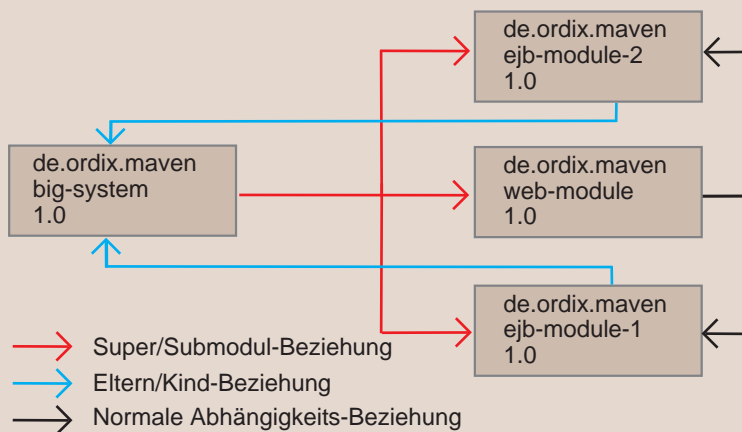


Abb. 1: Multi-Modul und Vererbungsbeziehungen.

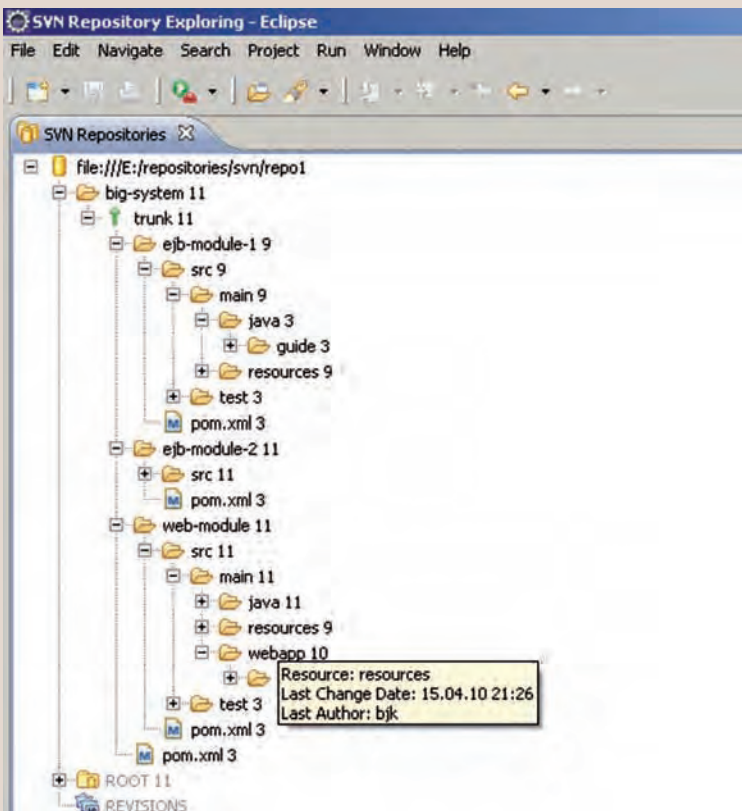


Abb. 2: Multi-Modul-Projektstruktur.

den einmalig im Elternprojekt festgelegt und sind somit nicht redundant vorhanden. Natürlich lassen sich auch Eltern/Kind- und Supermodul/Submodul-Beziehungen kombinieren. Wie in Abbildung 1 dargestellt, legen wir für die beiden EJB-Submodule das Modul Big-System

als Eltern-Projekt fest. Würden wir hier eine Java 1.5 Compiler-Konfiguration definieren, so wäre diese für beide Submodule gültig.

Die Maven-Projektstruktur aufsetzen

Eine der Grundvoraussetzungen von Maven ist, dass der Projektverzeichnisbaum eine fest vorgegebene Struktur einhält. Maven verlangt beispielsweise die Existenz des Ordners `src/main/java` für Dateien mit Java Source Code. Typischerweise hält man gewisse Template-Projekte bereit, um nicht jedesmal eine neue Projektstruktur aufsetzen zu müssen. Fängt man auf der grünen Wiese an, hilft das Maven Archetype Plug-In, um sich je nach Projekttyp eine gewünschte Projektstruktur und POMs erzeugen zu lassen.

Abbildung 2 zeigt die Code Repository Verzeichnisstruktur unseres exemplarischen Multi-Modul-Projekts. Zu beachten ist, dass die Submodule lediglich Unterverzeichnisse des Wurzelknotens sind und dem Modulnamen im Supermodul POM entsprechen müssen (siehe Abbildung 3). IDE-spezifische Steuerdateien (z. B. `.classpath`, `.project`, `.settings` für Eclipse) sollten möglichst nicht im Repository enthalten sein, da diese im Normalfall durch ein entsprechendes Maven PlugIn (z. B. das Maven-Eclipse-Plug-In) erzeugt werden sollten.

POMs konfigurieren

Nachdem die verschiedenen Abhängigkeitsbeziehungen eines Java-Projekts sorgfältig geplant worden sind, müssen die entsprechenden Konfigurationen in den POMs verankert werden. Die Abbildungen 3, 4 und 5 zeigen, wie die für unser Projekt Big-System angestellten Abhängigkeitsüberlegungen aus Abbildung 1 in die POM-Dateien eingeflossen sind. Hiermit ist die initiale Maven-Projektstruktur fertig. Die eigentliche Implementierung des Systems und Verwendung des Maven Build-Prozesses kann nun starten.

Eclipse-Projekte erzeugen

Sinnvollerweise sollte man für jedes Maven Submodul ein entsprechendes Eclipse-Projekt erzeugen. Dazu müssen die einzelnen Submodule in Eclipse importiert werden. Statt diese direkt aus dem Code Repository in Eclipse zu importieren, checkt man zunächst das Gesamtprojekt unabhängig von der IDE in ein beliebiges lokales Verzeichnis aus und importiert es später in den Eclipse Workspace. Auf diese Weise behält man sich

die Möglichkeit vor, auch auf Supermodul-Ebene einen Build zu starten, während dies bei einem direkten Import der Submodule in den Eclipse Workspace verloren gegangen wäre. Zu beachten ist also, dass die Arbeitskopie in keinem Eclipse Workspace Directory abgelegt wird.

Man kann nun unterschiedlich vorgehen: Entweder arbeitet man in Eclipse mit allen Submodulen oder man beschränkt sich nur auf bestimmte Module, für die man beispielsweise zuständig ist. Ist ersteres der Fall, lassen sich durch das Maven Eclipse-Plug-In [1] die nötigen Steuerdateien ohne weitere Optionen generieren. Maven wertet dabei spezifische Elemente des POM aus, um die Ressourcen `.classpath` (aus den Abhängigkeiten hergeleitet), `.project` und `.settings` (aus den Compiler-Plug-In-Angaben hergeleitet) zu erzeugen.

Ändert sich im Laufe des Projekts das POM, muss der Generierungsprozess erneut gestartet werden. Neben den normalen Abhängigkeiten werden auch Abhängigkeiten zwischen Submodulen durch das Plug-In erkannt und als Eclipse-Projektabhängigkeit dem Klassenpfad hinzugefügt. Hierdurch sind Änderungen im Source Code in abhängigen Modulen bereits zur Eclipse Build-Zeit sichtbar. Kompilierfehler fallen dem Entwickler so direkt auf. Sonst hätten diese erst bemerkt werden können, wenn der Maven Build die abhängige JAR-Datei aus dem Maven Repository geladen und mit der Kompilierung der Klassen begonnen hätte.

Entscheidet man sich hingegen für das Arbeiten mit bestimmten Modulen, aktiviert man im Maven Eclipse-Plug-In zusätzlich die Option `useProjectReferences=false`. Maven generiert in diesem Fall Steuerdateien, die keine Eclipse-Projektabhängigkeiten beinhalten. Einzelne Submodule lassen sich so fehlerlos in Eclipse importieren. Im Umkehrschluss zur ersten Variante sind Änderungen im Source Code von Abhängigkeiten erst zur Maven Build-Zeit sichtbar. In jedem Fall sollte vor dem Import der Projekte in Eclipse die Klassenpfadvariable `M2_REPO` angelegt sein und auf das lokale Maven Repository verweisen. Dies ist erforderlich, damit Eclipse die zum Kompilieren benötigten JAR-Dateien lokalisieren kann.

Maven in Eclipse

Abbildung 6 zeigt die einzelnen Submodule unseres Beispielprojekts als eigene Eclipse-Projekte. Die beiden Source-Verzeichnisse

```
<groupId>de.ordix.maven</groupId>
<artifactId>big-system</artifactId>
<packaging>pom</packaging>
<version>1.0</version>
<modules>
  <module>ejb-module-1</module>
  <module>ejb-module-2</module>
  <module>web-module</module>
</modules>
...
<Plug-In>
  <groupId>org.apache.maven.Plug-Ins</groupId>
  <artifactId>maven-compiler-Plug-In</artifactId>
  <version>2.0.2</version>
  <configuration>
    <source>1.5</source>
```

Abb. 3: Konfiguration des Supermoduls big-system/pom.xml.

```
<parent>
  <groupId>de.ordix.maven</groupId>
  <artifactId>big-system</artifactId>
  <version>1.0</version>
</parent>
<groupId>de.ordix.maven</groupId>
<artifactId>ejb-module-1</artifactId>
<version>1.0</version>
<packaging>ejb</packaging>
...
<build>
  <Plug-Ins>
    <Plug-In>
      <groupId>org.apache.maven.Plug-Ins</groupId>
      <artifactId>maven-ejb-Plug-In</artifactId>
    ...
  ...
  ...
```

Abb. 4: Konfiguration des Submoduls big-system/ejb-module-1/pom.xml.

```
<groupId>de.ordix.maven</groupId>
<artifactId>web-module</artifactId>
<version>1.0</version>
<packaging>war</packaging>
...
  <Plug-In>
    <groupId>org.apache.maven.Plug-Ins</groupId>
    <artifactId>maven-war-Plug-In</artifactId>
  ...
  <dependencies>
    <dependency>
      <groupId>de.ordix.maven</groupId>
      <artifactId>ejb-module-1</artifactId>
      <version>1.0</version>
    </dependency>
  ...
```

Abb. 5: Konfiguration des Submoduls big-system/web-module/pom.xml.

`java` und `test` wurden bereits durch das Maven Eclipse-Plug-In in den Klassenpfad der IDE eingefügt. Standardmäßig landen sowohl die von Eclipse als auch die von Maven kompilierten Klassen im Verzeichnis `target/classes` (bzw. `target/test-classes`). Das Verzeichnis `target/classes` dient als Behälter für Klassenpfad-Ressourcen des zu erzeugenden Zielarchivs. Für ein WAR-Archiv werden die Inhalte von `target/classes`

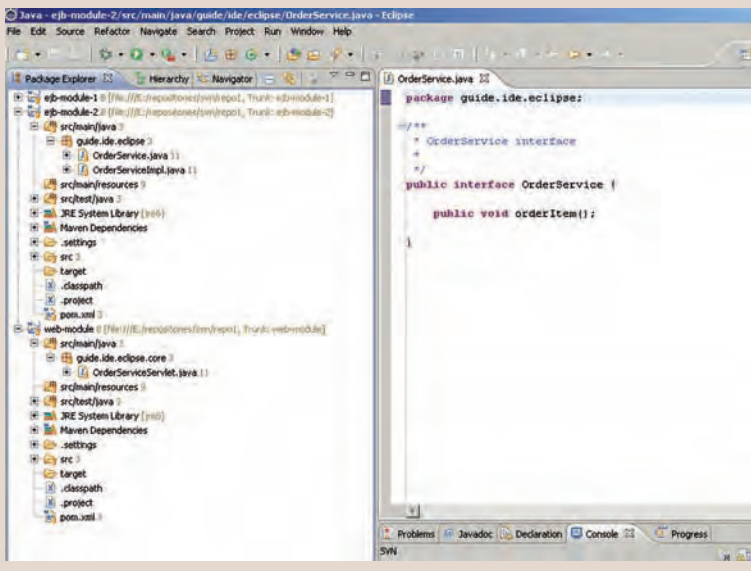


Abb. 6: Maven Multi-Module in Eclipse.

nach `webapp/WEB-INF/classes` gepackt. Inhalte von `src/main/resources` (z. B. eine `log4j.xml`-Datei) gelten ebenfalls als Ressourcen, die später im Klassenpfad nötig sind und werden ebenfalls nach `target/classes` kopiert. Um den Klassenpfad zwischen Eclipse und dem Zielarchiv konsistent zu halten, sollte man daher das Verzeichnis `src/main/resources` ebenfalls dem Eclipse Klassenpfad hinzufügen. Ressourcen, die man zwar in Eclipse aber nicht im Zielarchiv benötigt (beispielsweise ein `test-applicationContext.xml` für JUnit) sollten in dem Klassenpfad-Verzeichnis `src/test/resources` untergebracht werden.

Das M2Eclipse Plug-In

Die Ausführung von Maven Builds ist neben der Kommandozeile auch von der IDE heraus möglich. Um einen Build über Eclipse zu starten, wird das M2Eclipse Plug-In [2] benötigt. Ist das Plug-In installiert, können über das Run-Menü verschiedene Maven Builds konfiguriert und gestartet werden. Statt POM-Konfigurationen direkt per XML vorzunehmen, bietet das Plug-In außerdem die Möglichkeit, einen grafischen Editor bei der Bearbeitung eines POM zu nutzen. Eines der wesentlichen Features des Plug-In ist das automatische Dependency Management. Ist ein Projekt mit dieser Option versehen, wird der Klassenpfad in Eclipse durch den Maven ClasspathContainer erweitert. Der Container sorgt dafür, dass der Klassenpfad

diejenigen Bibliotheken enthält, die unmittelbar aus den Dependencies des POM hergeleitet werden können. Die statische Erzeugung der Klassenpfadeinträge durch das Maven Eclipse-Plug-In ist somit obsolet, da diese Aufgabe nun dynamisch zur Laufzeit der IDE von M2Eclipse übernommen wird.

Weitere Vorteile des M2Eclipse Plug-In stecken in dem Maven Builder. Hierdurch erreicht man, dass M2Eclipse auf die IDE Build Events `clean` und `resource change` reagiert und im Hintergrund ein gewünschtes Maven Goal ausführt. `Resource change` bezieht sich dabei nur auf Änderungen in den Ressourcen-Verzeichnissen und nicht etwa auf eine Java Source Code Modifikation. Ändert man beispielsweise die `log4j.xml`-Datei des Ressourcen-Verzeichnisses, würde der Maven Builder nach dem Speichern der Datei automatisch das Goal `process-resources` starten. Damit lassen sich komplexere Build-Funktionalitäten hinter einfachen Eclipse Builds verstecken. Würde die `log4j.xml`-Datei einen Platzhalter `${logDirectory}` definieren, der bei Ausführung des Goals `process-resources` durch das umgebungsspezifische Log-Verzeichnis ersetzt wird, erreichen wir durch ein Speichern der Datei, dass diese mit der gewünschten Ersetzung im Klassenpfad vorliegt.

Nutzung des Maven Build Lifecycles

Maven funktioniert grundsätzlich nach dem Plug-In-/Goal-Prinzip. Ein Plug-In bietet eine gewisse Build-Funktionalität (z. B. Compiler) und stellt verschiedene Aktionen (= Goals) zur Verfügung, die ausgeführt werden können (z. B. `compile` oder `help`). Goal-spezifische Optionen (z. B. `showWarnings`) werden mit der Option `-D` angehängt. Optionen, die dauerhaft verwendet werden, können einfach in das POM verlagert werden.

Da ein Build aus einer fest vorgegebenen Sequenz von Aktionen bestehen sollte, stellt Maven vordefinierte Lifecycles bereit [3]. Es gibt drei verschiedene Lifecycles: `clean`, `Default` und `site`.

Jeder Lifecycle besteht aus Phasen, die eine bestimmte Reihenfolge haben. Bei dem Aufruf einer Phase werden alle vorherigen sequentiell aufgerufen. Hinter jeder Phase steckt ein von Maven vorgegebener Plug-In-/Goal-Aufruf. Für die Phase `compile` ruft Maven zum Beispiel implizit `compiler:compile` auf.

Ein Maven Build wird typischerweise durch die Angabe einer Phase pro Lifecycle gestartet.

Statt einer Phase kann aber auch ein direkter Plug-In-/Goal-Aufruf erfolgen. Die wichtigsten Phasen stecken im Default LifeCycle. Neben der Generierung des eigentlichen Zielarchivs unterstützt der Default Lifecycle den Entwickler dabei, möglichst stabile Working Copy-Änderungen in das Code Repository committen zu können. Das erfolgreiche Durchlaufen der Phasen `test` und `integration-test` dient hierbei als wesentliches Kriterium, ob die lokalen Änderungen eingeecheckt werden sollten. Voraussetzung ist hier natürlich, dass die Phasen mit entsprechenden Test-Frameworks und ausreichend Testläufen konfiguriert sind. Um Integrationstests überhaupt lauffähig zu machen, sollten weitere Plug-Ins in das POM integriert und den Integrations-test-Phasen zugeordnet sein. Nennenswert ist hier insbesondere das Maven-SQL-Plug-In [4] zur Erzeugung von Datenbanktabellen und Testdaten sowie das Maven-Cargo-Plug-In [5] für das Deployment von WAR- oder EAR-Archiven in gängige Applicationserver.

Build-Profile und Properties

Durch die Verwendung von Maven Profilen [6] können Software-Builds möglichst umgebungsunabhängig durchgeführt werden. Das Level of Portability bestimmt, wie einfach es, ein Projekt erstmals in einer bestimmten Umgebung zu bauen. Das Level „Environment Portability“ ist erreicht, wenn sich das Projekt durch das alleinige Setzen eines Schalters sowohl in die Test- als auch Produktionsumgebung einsetzen ließe. Maven unterstützt dies durch die Möglichkeit, eigene Build-Profile zu definieren. Hinter einem Profil verbergen sich in der Regel umgebungsspezifische Properties (z. B. Datenbank-URLs), die im Build-Prozess herangezogen werden, wenn das spezifische Profil aktiviert worden ist.

Typischerweise müssen umgebungsspezifische Properties nicht nur im Build-Prozess vorliegen, sondern auch der Anwendung bekannt sein. Verwendet ein Build das Property URL-DB, um eine Test-Datenbank aufzusetzen, so muss die URL auch in der Anwendung verfügbar sein, damit eine Verbindung zu dieser Datenbank aufgebaut werden kann. Maven unterstützt dies durch einen speziellen Filtermechanismus.

Filter werden während der Phase Process Resources angewendet und sorgen dafür, dass Platzhalter innerhalb von Ressourcen-dateien durch die im Build eingelesenen Properties ersetzt werden. Realisieren lässt sich dieses Vorhaben dadurch, dass eine Property-Datei im Ressourcenverzeichnis als Vermittler zwischen Build und Anwendung fungiert. Die

Glossar

IDE	Integrated Development Environment, auch Integrated Design Environment. Eine integrierte Entwicklungsumgebung ist ein Anwendungsprogramm zur Entwicklung von Software.
POM	Project Object Model. Hier werden sämtliche Informationen über das Software-Projekt in einer XML-Datei mit dem Dateinamen pom.xml (für Project Object Model) gespeichert. Diese Datei enthält alle Informationen zum Softwareprojekt und folgt einem standardisierten Format.

Links

- ▶ [1] Maven Eclipse-Plug-In: <http://maven.apache.org/plugins/maven-eclipse-plugin/>
- ▶ [2] M2Eclipse Plug-In: <http://m2eclipse.sonatype.org/>
- ▶ [3] Lifecycles: <http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>
- ▶ [4] Maven-SQL-Plug-In: <http://mojo.codehaus.org/sql-maven-plugin/>
- ▶ [5] Maven-Cargo-Plug-In: <http://cargo.codehaus.org/Maven2+plugin>
- ▶ [6] Maven Profile: <http://maven.apache.org/guides/introduction/introduction-to-profiles.html>

Values einer solchen Property-Datei bestehen aus Platzhaltern, die im Build-Prozess durch umgebungsspezifische Properties ersetzt werden. Als Key stehen in der Datei Property-Namen, die der Anwendung bekannt sind und beim Start der Applikation zur Variableninitialisierung eingelesen werden.

Fazit

Durch den Einsatz von Maven Multi-Modulen lassen sich Build-Prozesse größerer Java-Projekte einfach und effizient verwalten. Abhängigkeiten vieler einzelner Submodule werden automatisch per Dependency Management zugeordnet, außerdem werden redundante Konfigurationen durch Eltern/Kind-Beziehungen verhindert. Die Unterstützung von Maven bei der Eclipse-Integration ist eine weitere wesentliche Erleichterung für den Entwickler. Das Maven Eclipse-Plug-In generiert hier für das gesamte Multi-Modul-Projekt die nötigen Steuerdateien. Das M2Eclipse Plug-In bietet darüber hinaus eine optimale Möglichkeit, den Maven Build Lifecycle aus Eclipse heraus zu steuern sowie abhängige Bibliotheken einfach durch den Maven ClasspathContainer zu verwalten. Natürlich sollten nach Konfiguration der Maven-Umgebung die Vorteile des Build Lifecycles effizient genutzt werden. Insbesondere die Ausführung von JUnit- und Integrationstests mit entsprechenden Plug-Ins und Build-Profilen bieten dem Entwickler eine gute Möglichkeit, einfach und schnell die Qualität seiner Code-Änderungen zu überprüfen.



*Björn Konrad
(info@ordix.de).*



Oracle 11g Security (Teil II)

Maßnahmen für eine erhöhte Passwortsicherheit

Dieser Artikel richtet sich an Datenbankadministratoren, Sicherheitsbeauftragte und Datenbankentwickler.

Neben der Initiative „Security by Default“, die wir Ihnen im letzten Teil dieser Reihe vorgestellt haben [1], hat Oracle auch am Thema Passwortschutz/-verwaltung gearbeitet und liefert mit der Version 11g Release 1 einige interessante Neuerungen. Hierauf liegt der Fokus in diesem Artikel.

Motivation für eine erhöhte Passwortsicherheit

Die Arbeit in unseren Kundenprojekten hat gezeigt, dass Sicherheitslücken, die durch eine zu einfache bzw. vernachlässigte Benutzer- und Passwortverwaltung entstehen können, oftmals unterschätzt werden.

Gerade vor dem Hintergrund, dass Passwort-Cracker immer schneller werden, dass in der Praxis stets noch viele Standard- oder Applikationsbenutzer mit wohlbekanntem Passwörtern existieren oder administrative Systemkennungen zu einfach und auch gerne für mehrere Produktionsinstanzen identisch gesetzt werden, lässt die Problematik in puncto Datenbanksicherheit leicht erkennen.

Dementsprechend willkommen sind die neu eingeführten Maßnahmen, die in der aktuellen Oracle 11g Datenbank zu einer erhöhten Passwortsicherheit beitragen sollen.

Stärkerer Passwortschutz durch case-sensitive Passwörter

Die Erkennung von Groß- und Kleinschreibung bei Kennwörtern, was bei Unix-Systemen längst zur üblichen Notation gehört, wird ebenfalls mit Version 11g eingeführt.

Gemäß dem neuen Standard werden Passwörter jetzt zusätzlich case-sensitiv erstellt und sind aufgrund des hierdurch erweiterten Zeichensatzes stärker gegenüber Angriffen – z. B. per Brute-Force oder über Wörterbuch – gewappnet.

Diese Funktionalität lässt sich über den dynamischen Parameter `sec_case_sensitive_logon` zu- und abschalten (Default: `true`).

Bei Benutzern, die aus früheren Datenbankversionen migriert oder importiert wurden, bleiben die Kennwörter zunächst case-insensitiv erhal-

ten, bis das Passwort erstmalig in Oracle 11g geändert wird. Erst danach zieht die neue case-sensitive Authentifizierung.

Passwortversionen

Welche Kennwortmethode pro Benutzer verwendet wird, kann der zusätzlichen Spalte `PASSWORD_VERSIONS` in der View `DBA_USERS` entnommen werden (siehe Abbildung 1). Wie das Beispiel in Abbildung 1 zeigt, wurde Benutzer `RON` über ein Upgrade übernommen, das Kennwort aber noch nicht umgesetzt, so dass die alte Kennwortmethode über Großbuchstaben (10G) hier weiterhin gilt.

Demgegenüber wird bei den beiden Benutzern `JOE` und `BOB` die neue case-sensitive Methode befolgt, sollte sie denn über `sec_case_sensitive_logon=TRUE` eingestellt sein. Der Hash-Wert des case-sensitiven Passwortes (11G) wurde hier entweder durch ein unter 11g neu erstelltes oder auf 11g gewechseltes Passwort erzeugt.

Bei `BOB` wird zudem die Case-Sensitivität des Passwortes erzwungen, indem nur der 11g Oracle Hash als Wert (`IDENTIFIED BY VALUES`) verwendet wurde; ein Wechsel auf das alte Verfahren ist hier nicht mehr möglich, was vielleicht die sicherste Einstellung an dieser Stelle darstellt.

Datenbank-Links und Case-Sensitivität

Passwörter in früheren Releases sind nicht case-sensitiv. Sie werden ausschließlich in Großbuchstaben abgelegt. Aus diesem Grund kann sich hier der Verbindungsaufbau zu einer „case-sensitiven“ Oracle 11g Datenbank mittels DB-Link als problematisch herausstellen.

Sobald Groß- und Kleinschreibung auf der entfernten Gegenseite abgeprüft wird, gilt es, für die Angabe des Passwortes folgende Besonderheiten zu berücksichtigen:

- 11g → 11g
Case-Sensitivität bleibt erhalten
→ exakte Passwort-Eingabe
- 10g → 11g
Problematisch, da 10g nur Großbuchstaben liefert
→ Passwort-Neuanlage in Großbuchstaben auf 11g Zielsystem

```
SQL> select username, password_versions
      from dba_users;
```

USERNAME	PASSWORD
RON	10G
JOE	10G 11G
BOB	11G

Abb. 1: Kennwortmethoden der einzelnen Oracle Versionen.

```
orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_SID ignorecase=y
Enter password for SYS: <password>
```

Abb. 2: Befehl zum Anlegen einer case-insensitiven Passwortdatei.

```
SQL> select password, spare4 from user$ where name='KHO';
```

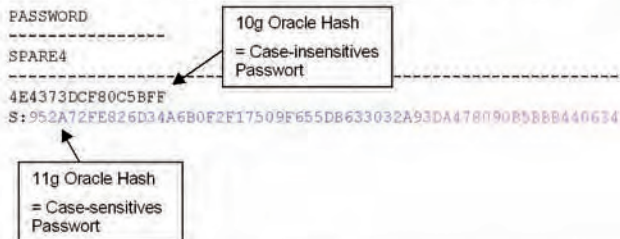


Abb. 3: Ablage der Hash-Werte beider Passwort Versionen im Data Dictionary.

```
SQL> select * from dba_users_with_defpwd;
```

USERNAME
OUTLN
SCOTT
SYSTEM
XDB
...

Abb. 4: Abfrage zum Auffinden von Standardkennwörtern.

- 11g → 10g
Case-Sensitivität geht verloren, da authentifizierende 10g Datenbank das Passwort in Großbuchstaben umwandelt
→ beliebige Passwort-Eingabe

```
select username
  from dba_users_with_defpwd
 where username in (select username
                   from dba_users
                   where account_status = 'OPEN');
```

Abb. 5: Abfrage zum Auffinden von Standardkennwörtern von aktiven Benutzern.

Glossar

Brute-Force	Angriffsmethode zum „Knacken“ von Passwörtern. Es werden hierbei alle möglichen Kombinationen eines Passwortes (= Prüfung von „random passwords“) ausgetestet. Diese Methode ist besonders erfolgreich für Passwörter geringerer Länge (bis 8 Stellen, je nach Passwort-Algorithmus).
Data Dictionary	Systemtabellen und Views zur Verwaltung der Oracle-Datenbank.
DB-Link	Ein Datenbank-Link ermöglicht den Zugriff aus einer Datenbank heraus über Oracle Net auf eine beliebige, andere Datenbank.
DES	Data Encryption Standard. Symmetrischer Algorithmus zur Bildung eines 64-bit langen Hash-Wertes über <code>username upper(password)</code> . Username dient hierbei als SALT-Ersatz.
Hashing	Mathematische Funktion, die auf einen Klartext angewendet wird. Ergebnis ist ein Hash-Wert, also eine neue – unverständliche – Zeichenkette fester Länge. Hashing funktioniert nur in einer Richtung. Da es keine Möglichkeit gibt, aus dem Hashwert auf den ursprünglichen Wert zurückzuschließen, ist Hashing quasi eine Einbahnstraße.
SALT	Gleiche Werte, z. B. identische Passwörter, werden beim SALT-Prinzip durch das Anfügen von zufällig generierten Zeichen („random bits“) unterschiedlich verschlüsselt, so dass keine Schlussfolgerungen bzgl. der Gemeinsamkeiten von Werten gezogen werden können.
SHA-1	Secure Hash Algorithmus, der auf das übergebene Passwort inkl. echtem SALT einen Hash-Wert von 160 Bit Länge erzeugt.

Stärkerer Passwortschutz: Case-sensitive Passwortdatei

Die entfernte Authentifizierung von Administratoren mit SYSDBA- oder SYSOPER-Privilegien wird außerhalb der Datenbank mit Hilfe der Passwortdatei vorgenommen.

Auch bei entfernten Anmeldungen über SYSDBA oder SYSOPER gilt ab Oracle 11g grundsätzlich die Nutzung case-sensitiver Passwörter. Diese Wirkungsweise kann hier über die Definition des Password Files mit dem Befehl aus Abbildung 2 wieder deaktiviert werden.

Zu beachten ist, dass Passwörter von aus früheren Releases migrierten oder importierten Benutzern mit SYSDBA- bzw. SYSOPER-Privileg vorerst case-insensitiv bleiben. Erst die nächste Änderung ihres Passwortes unter

Verwendung einer neu angelegten, case-sensitiven Passwortdatei berücksichtigt auch die Groß- und Kleinschreibung bei Anmeldungen über das Password File.

Passwort-Hashing in Oracle 11g

In Oracle 11g wird das verschlüsselte Passwort nicht mehr in der Spalte PASSWORD der Data Dictionary View DBA_USERS angezeigt.

Benutzer mit dem SELECT-Privileg auf DBA_USERS (via `SELECT_CATALOG_ROLE`) werden somit ab jetzt daran gehindert, über den Hash-Wert der Kennwörter Manipulationen unter fremden Benutzerkennungen vorzunehmen.

Verfahren, die sich mittels `ALTER USER <username> IDENTIFIED BY VALUES...;` weiterhin dem verschlüsselten Kennwort bedienen möchten, benötigen stattdessen den Lesezugriff auf die interne Tabelle SYS.USER\$ (via `SELECT ANY DICTIONARY`).

In USER\$.PASSWORD wird nach wie vor das case-insensitive Passwort in Form des Oracle 10g Hash Wertes (= 64-bit DES Hash über `USERNAME || UPPER(PASSWORD)`) abgelegt.

Für USER\$.SPARE4 wird das case-sensitive Passwort nun zusätzlich mit einem „echten“ SALT zusammengesetzt und hierüber ein Hash-Wert mit dem stärkeren Hash-Algorithmus SHA-1 (160-bit Key, siehe blaue Markierung in Abbildung 3) gebildet. Anschließend wird dieser Hash-Wert nochmals mit selbigem SALT (10 Bytes - siehe rote Markierung in Abbildung 3) zusammengesetzt und in SPARE4 gespeichert. Der abgelegte SALT wird später, im Rahmen des Anmeldeprozesses, für einen Passwortabgleich an den zu authentifizierenden Client übertragen.

Stärkerer Passwortschutz: Prüfung auf Standardpasswörter

Aus den bisherigen Erfahrungen gehören Standardbenutzer, die beim Einrichten einer Oracle Datenbank automatisch mitangelegt werden und über wohlbekannt Standardkennwörter verfügen, mit zu den Hauptrisiken für einen unberechtigten Zugriff auf die Datenbank durch Dritte.

Demzufolge sollte das Auffinden von Standardkennwörtern in jedem Security-Projekt gleich am Anfang durchgeführt werden.

Mit der in Oracle 11g eingeführten View DBA_USERS_WITH_DEFPWD trägt Oracle dieser Aufgabenstellung Rechnung. Eine einfache Abfrage auf diese View ergibt eine Liste der von Oracle angelegten Benutzer, die noch über ihr Standardkennwort verfügen (siehe Abbildung 4).

Da die neue View auch gesperrte (LOCKED) Datenbank-Accounts beinhaltet, empfiehlt sich, die Abfrage mit der View DBA_USERS zu verknüpfen (siehe Abbildung 5).

Als Ergebnis werden nur noch Standardbenutzer mit einem Standardkennwort und aktivem Account Status (OPEN) angezeigt, was hier von eigentlichem Interesse ist und regelmäßig kontrolliert werden sollte.

Fazit

Oracle hat in der aktuellen Datenbankversion einige sinnvolle Maßnahmen im Angebot, um das Datenbanksystem und dessen Benutzer zu härten. Datenbankadministratoren und vermehrt auch Entwickler sind aufgerufen, von diesem Angebot Gebrauch zu machen. Auch

Links

- ▶ [1] ORDIX News Artikel „Secure by Default“ in Oracle 11g – Die sichere Datenbank out-of-the-box?“ http://www.ordix.de/ORDIXNews/1_2010/Datenbanken/secure_by_default.html.html
- ▶ [2] Buch „HOWTO Secure and Audit Oracle 10g and 11g“, Autor: Ron Ben Natan, ISBN-10: 1420084127, ISBN-13: 978-1420084122
- ▶ [3] Buch „Oracle Database 11g New Features für DBAs and Developers“, Autor: Sam R. Alapati, Charles Kim, ISBN-10: 1590599101, ISBN-13: 978-1590599105
- ▶ [4] Präsentation „Oracle 11g Security“ von Pete Finnigan; <http://www.petefinnigan.com>

wenn dies mit gewissem Mehraufwand verbunden ist, wie die eigenen Password Policies anzupassen, Accounts in migrierten Datenbanken umzustellen, Standard-Accounts regelmäßig zu überprüfen und auf komplexere Passwörter bzgl. Gültigkeit, Case-Sensitivität und Länge umzusteigen. Passwörter, die nicht mehr so einfach zu „hacken“ sind und somit die Datenbank nach außen hin besser absichern, werden den Mehraufwand lohnen.



*Kathleen Hock
(info@ordix.de)*

Seminarempfehlung: Oracle Security

▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/837/index.html>

In diesem Seminar werden potenzielle Sicherheitsschwachstellen aufgezeigt und Lösungsansätze erarbeitet. Somit können Sie anschließend Daten vor unberechtigtem Zugriff schützen und in der Datenbank sowie im Netz verschlüsseln. Außerdem können Sie dafür sorgen, dass Applikationen je nach Berechtigung Daten individuell präsentieren.

Seminarinhalte

- Auditing
- Datenzugriffskontrolle
- Identifizierung und Authentifizierung
- Verschlüsselung
- Absicherung der Datenübertragung
- Überprüfen von Sicherheitslücken
- Database Security in Oracle 11g
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

- 16.08. - 19.08.2010 in Wiesbaden
- 18.10. - 21.10.2010 in Wiesbaden
- 06.12. - 09.12.2010 in Wiesbaden

Seminar-ID: DB-ORA-33A

Dauer: 4 Tage

Preis pro Teilnehmer: 1.590,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.431,00 € (zzgl. MwSt.)

Wir führen unsere Seminare auch jederzeit an einem geeigneten Ort Ihrer Wahl durch und bringen, wenn nötig, auch das entsprechende Equipment mit. Informieren Sie sich am besten im Internet über unsere Kundenseminare und die mobilen Schulungen: <http://training.ordix.de>.

XEN – Alles sicher auf der Festplatte?

Dieser Artikel richtet sich an erfahrene XEN-Administratoren, Entscheider und Berater, die sich mit neuen Funktionalitäten der aktuellen Version vertraut machen wollen.

In der ORDIX News 1/2010 haben wir eine Artikelreihe [1] über neue und spezielle XEN-Funktionen begonnen. Diese Reihe setzen wir nun mit dem Thema Festplattenverwaltung rund um XEN fort. Hierbei gehen wir auf die unterstützten Festplattenformate, die Arten der Bereitstellung von Festplatten an Gäste, aber auch auf Backup-Möglichkeiten dieser Speichertechniken ein. Der Artikel basiert auf der Version 4 der freien Open Source Virtualisierungslösung XEN, die Anfang April 2010 erschienen ist.

Viele unterstützte Plattenformate

XEN unterstützt unterschiedliche Arten von Speicher-Hardware, um die anfallenden Daten der Gastbetriebssysteme und Anwendungen zu speichern. Da XEN an vielen Stellen normale Betriebssystembefehle und Treiber verwendet, wird hier quasi alles unterstützt, was das Betriebssystem selbst auch ohne Virtualisierung unterstützt. Die Entscheidung, welches Speichermedium genutzt wird, hängt natürlich von unterschiedlichen Faktoren ab, wie z. B. der Art des Backup, Einsatz im Cluster, Nutzung der Funktionen der Live-Migration, Möglichkeit der Vergrößerung der Dateisysteme, und so weiter. Folgende Übersicht zeigt die generell unterstützten Verfahren:

- **Datei-Image**
Die virtuelle Festplatte speichert die Daten in einer Datei, die im Dateisystem der Domain 0 liegt. Hier werden verschiedene Arten von Formaten anderer Hersteller, wie z. B. VMware unterstützt. Ein weiteres Beispiel wäre ein ISO Image einer CD-ROM.
- **Komplette Partition**
Der Domain U wird eine ganze Festplatte oder eine Partition zugewiesen.
- **Logical Volume**
Die Domain U greift auf Volumes des Linux Logical Volume Managers (LVM) zu.

- **Network File System**

Die Domain U's liegen als Datei-Image auf einem NFS-Server, auf die mehrere Domain U's gleichzeitig zugreifen können.

- **Storage Area Network**

Die Domain U's liegen auf SAN Storage.

- **ISCSI/NBD**

Die Domain U's liegen auf ISCSI oder Network Block Device.

Zugriff der Gäste

Es ist vollkommen unabhängig davon, ob ein Gast nun ein Datei-Image, eine komplette Partition oder ein anderes blockorientiertes Gerät zugewiesen bekommt. Innerhalb des Gastes wird eine Gerätedatei benötigt, um auf das entsprechende Gerät zuzugreifen. Arbeitet man mit klassischen Konfigurationsdateien, würde ein Eintrag in der Konfigurationsdatei des Gastes – wie in Abbildung 1 dargestellt – dem Gastsystem drei Festplatten auf drei verschiedene Arten und Weisen zur Verfügung gestellt.

Mit `phy` wird dem Gast eine physikalische Festplatte der Domain 0 zur Verfügung gestellt. In diesem Beispiel wird dem Gast die Partition `/dev/sdc1` als Festplatte `/dev/sdc` zur Verfügung gestellt. Mit `file` und `tap:aio` werden dem Gast dateibasierte Images zur Verfügung gestellt, mit dem Unterschied,

dass bei `tap:aio` durch die Domain 0 kein Caching vorgenommen wird. Unabhängig davon, ob man eine Festplatte als `/dev/hda` oder `/dev/sda` zur Verfügung stellt, wird innerhalb des Gastes immer der `xenblk`-Treiber für den Zugriff auf die virtuelle Festplatte genutzt.

Eigene Nummer, eigener Name

Damit der Administrator die virtuellen Festplatten auch als solche erkennt, gibt es die Möglichkeit, spezielle XEN-eigene Gerätenamen innerhalb des Gastes zu nutzen. Taucht bei einem `df -k` der Gerätename `/dev/xvda1` auf, ist es für den Administrator sehr einfach zu erkennen, dass es sich hierbei um eine virtualisierte Festplatte innerhalb eines XEN-Gastes handelt. Hierfür haben sich die Entwickler eine eigene Major-Nummer innerhalb des Linux Kernels reservieren lassen. Die Major Nummer 202 steht demnach für „XEN Virtual Block Device“. Mit der Minor-Nummer 0 beginnt das Gerät `/dev/xvda`, bei der Minor-Nummer 16 das Gerät `/dev/xvdb`. Pro Festplatte können hier bis zu 15 Partitionen angelegt werden. Mit Minor-Nummer 240 beginnt die letzte Gerätedatei `/dev/xvdp`.

Platten kommen und gehen

XEN wäre nicht XEN, wenn die Festplattenverwaltung nicht auch online funktionieren würde. Für die Festplattenverwaltung von XEN-Gästen haben die Entwickler drei Befehle zur Verfügung gestellt:

- `xm block-attach`
- `xm block-list`
- `xm block-detach`

Soll einem Gastssystem nun online eine zusätzliche Festplatte zugewiesen werden, kann dies einfach mit dem Befehl

```
xm block-attach server1
file:/xen/cdrom.iso /dev/xvdc r
```

bewerkstelligt werden. In diesem Fall bekommt der Gast `server1` ein ISO-Image als Gerät `/dev/xvdc` zur Verfügung gestellt. Der Zugriff ist hierbei nur lesend (r) gestattet. Mit `xm block-list server1` können alle Festplatten, die einem Gastssystem zugewiesen sind, angezeigt werden (siehe Abbildung 2).

Leider haben die Entwickler vergessen, diesem Befehl eine für den Administrator lesbare Ausgabe zu spendieren, so dass dieser sofort erkennen könnte, um welche Geräte es

```
disk = [
'file:/xen/gast1.img,/dev/hda,w', 'tap:aio:/xen/data1.img,/dev/
sda,w', 'phy:/dev/sdc1,/dev/sdc,w' ]
```

Abb. 1: Konfiguration virtueller Festplatten eines XEN-Gastes.

Vdev	BE	handle	state	evt-ch	ring-ref	BE-path
51712	0	0	4	7	770	/local/domain/0/backend/tap/2/51712
5632	0	0	4	9	795	/local/domain/0/backend/vbd/2/5632
2048	0	0	4	8	842	/local/domain/0/backend/tap/2/2048

Abb. 2: Ausgabe des Befehls `xm block-list server1`.

<code>/dev/hda</code>	768
<code>/dev/hdb</code>	832
<code>/dev/sda</code>	2048
<code>/dev/sdb</code>	2064
<code>/dev/xvda</code>	51712
<code>/dev/xvdb</code>	51728

Abb. 3: Übersicht häufig verwendeter XEN-Geräte.

```
0000:05:05.0 SCSI storage controller: LSI Logic / Symbios Logic LSI1068 (rev 01)
0000:08:04.0 Ethernet controller: Broadcom Corporation Broadcom BCM5715 Gigabit
Ethernet (rev a3)
```

Abb. 4: Ausgabe des Befehls `lspci -D` zur Ermittlung der PCI-ID.

```
title Xen -- opensUSE 11.2 - 2.6.31.12-0.2
root (hd0,0)
kernel /boot/xen.gz dom0_mem=2048M noreboot
module /boot/vmlinuz-2.6.31.12-0.2-xen root=/dev/sda1
pciback.hide=(0000:05:05.0)(0000:08:04.0)
module /boot/initrd-2.6.31.12-0.2-xen
```

Abb. 5: Anpassung des Bootmanagers, um ein PCI-Gerät exklusiv in einem Gast zu verwenden.

sich bei 51712, 5632 oder 2048 handelt. Mit etwas Mathematik kommt man hier zu einer Lösung: Das Gerät `/dev/sda` hat die Major-Nummer 8 und die Minor-Nummer 0. Rechnet man nun $(8 * 256) + 0$ kommt man auf 2048. Vdev 2048 ist also das Gerät `/dev/sda` innerhalb des Gastes. Abbildung 3 zeigt häufig verwendete Geräte. Mit dem Befehl `xm block-detach server1 2048` kann das Gerät dem Gastssystem wieder entzogen werden. Arbeitet man mit dem in der letzten ORDIX News beschriebenen Xenstore, ist nach `xm block-attach`, oder `xm block-detach` auch keine Konfigurationsdatei mehr anzupassen. Die Festplattenzuweisung ist dann sofort ohne Anpassung von Konfigurationsdateien, dauerhaft zugewiesen.

```
# xm pci-list-assignable-devices
0000:05:05.0
0000:08:04.0
```

Abb. 6: Ausgabe des Befehls `xm pci-list-assignable-devices`.

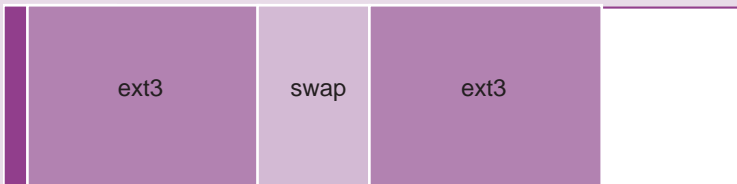


Abb. 7: Partitioniertes Festplatten-Image eines Gastsystems.

Bitte noch mehr Platten!

Häufig benötigen XEN-Gäste sehr viele Festplattengeräte. Wählt man den bisher beschriebenen Weg, muss eine Festplatte, die auf der Domain 0 existiert, immer per Konfigurationsdatei oder Xenstore zugewiesen werden. Der Gast nutzt dann zum Zugriff den Treiber `xenblk`. Unter XEN existiert nun auch die Möglichkeit, PCI-Geräte an Gäste durchzureichen. Wurde ein SCSI Controller oder ein anderes PCI Gerät einem Gastsystem zugewiesen, ist der Zugriff auf das Gerät nur noch von diesem und nicht mehr von Domain 0 möglich. Zu beachten ist hierbei, dass ein PCI Controller immer nur von einer Domain genutzt werden kann. Da der Linux Kernel beim Starten der Domain 0 aber die Angewohnheit hat, für alle gescannten Geräte den Treiber zu laden, muss dies zukünftig unterbunden werden. Hierzu muss zuerst die PCI-ID des Gerätes herausgefunden werden, die dem Gast zugewiesen werden soll. Führt man den Befehl `lspci -D` in der Domain 0 aus, sind die notwendigen Werte schnell ermittelt. Abbildung 4 zeigt eine gekürzte Ausgabe dieses Befehls. Abbildung 5 zeigt die notwendige Anpassung des Bootmanager Grub, die das Initialisieren der Geräte durch die Domain 0 unterbindet.

Wichtig ist hierbei, dass der `pciback`-Treiber schon beim Booten der Domain 0 mit Hilfe der Initial Ramdisk geladen wird. Dies wird je nach eingesetztem Linux-Derivat unterschiedlich realisiert.

Bei SuSE-Derivaten ist das Setzen der Variable `INITRD_MODULES="pciback"` in der Datei `/etc/sysconfig/kernel` mit an-

schließendem Aufruf des Befehls `mkinitrd` zielführend. Wird mit aktiviertem `pciback`-Treiber nun der Befehl `xm pci-list-assignable-devices` aufgerufen, werden die möglichen Geräte, die für Gäste zur Verfügung stehen, angezeigt. Abbildung 6 zeigt die Ausgabe des Befehls. Mit `xm pci-attach server1 0000:05:05.0` kann der SCSI Controller dem Gast online zugewiesen werden. `xm pci-list server1` zeigt die von einem Gast verwendeten PCI-Geräte an und `xm pci-detach server1 0000:05:05.0` entfernt ein PCI-Gerät wieder von einem Gast. Und da XEN immer wieder für eine Überraschung gut ist, kann ein Gastsystem, das mit dem Schalter `blkif=yes` gestartet wurde, auch wieder als Block Device Backend fungieren und anderen Gästen Festplatten zur Verfügung stellen.

Gestern VMware, heute XEN und morgen???

Arbeitet man mit dateibasierten Gästen, bringt das RPM-Paket `qemu` ein sehr praktisches Werkzeug namens `qemu-img` mit. Mit diesem Befehl kann der Administrator dateibasierte Images in unterschiedlichen Formaten erstellen. Aber was noch viel praktischer ist, er kann sie auch konvertieren. Beispielsweise erstellt `qemu-img create -f raw /xen/data1 8G` ein Image im raw-Format. Mit `qemu-img convert -O raw /xen/server1.vmdk /xen/server1.raw` kann ein VMware-Gast in das raw-Format umgewandelt werden. In der aktuellen Version werden alle namhaften Image-Formate unterstützt, so dass ohne Neuinstallation der Gäste von einer Virtualisierungslösung zu einer anderen gewechselt werden kann.

XEN kann mittlerweile auch direkt mit VMware Images umgehen, ohne das Format vorher zu konvertieren. Mit `xm block-attach server1 tap:vmdk:/xen/data2.vmdk w` kann einem Gast direkt eine `vmdk`-Datei als Speichermedium zur Verfügung gestellt werden. Besonders interessant ist auch das `qcow2`-Format, mit dem man Snapshots von Festplatten erstellen kann.

Der ausgesperrte Admin

Für ein Disaster Recovery oder ein Backup kann es sinnvoll sein, direkt von der Domain 0 auf die Festplatten eines Gastsystems zuzugreifen zu können. Befindet sich nur eine einzige

Partition mit Dateisystem auf dem Image, kann dieses meistens einfach gemountet werden. Häufig sind aber mehrere Dateisysteme oder aber auch eine Mischung aus Dateisystem und Swap auf einem Festplattenabbild. Abbildung 7 zeigt ein Beispiel eines derart partitionierten Festplatten-Images. Soll nun auf die dritte Partition mit dem EXT3-Dateisystem zugegriffen werden, kann dies mit dem Hilfswerkzeug `kpartx` bewerkstelligt werden. `kpartx -a /xen/server1.raw` stellt für jede Partition, die sich auf dem Dateimage befindet unterhalb des Verzeichnisses `/dev/mapper` des Host-Betriebssystems eine Gerätedatei zur Verfügung. Diese kann dann eingebunden werden. `kpartx -l /xen/server1.raw` zeigt an, was der Befehl mit `-a` ausführen würde. Und `kpartx -d /xen/server1.raw` löscht die bereitgestellten Gerätedateien wieder.

Backup und Flexibilität

Am elegantesten ist die Administration von Festplatten der virtuellen Gäste beim Einsatz des Linux Logical Volume Managers (LVM). Hier kann der Administrator viele Festplatten schon unter Domain 0 zu einem Volume verknüpfen. Bei einer Vergrößerung des Dateisystems muss nur das Volume vergrößert, nicht aber die Gastkonfiguration angepasst werden. Ein weiterer Vorteil liegt in der Erweiterbarkeit. Volumes und Dateisysteme können während des laufenden Betriebs online vergrößert werden, so dass es zu keinen Ausfallzeiten kommt. Eine besonders praktische Funktion des LVM ist das Anlegen von Snapshots. Ein Snapshot ist eine Momentaufnahme des aktuellen Volume-Zustandes.

Mountet man einen Snapshot, kann man hiervon in aller Ruhe ein Backup erstellen, ohne dass Dateien, die sich während des Backup verändern, negativen Einfluss darauf haben. Ein Snapshot eines bestehenden Volume kann unkompliziert mit dem Befehl `lvcreate -s -L 2G -n server1-backup /dev/datavg/server1` erstellt werden. Die angegebene Größe 2G entspricht hierbei jetzt aber nicht der Originalgröße des Volume. Im Gegensatz zu anderen Volume Managern ist ein LVM Snapshot keine Komplettkopie des Originals. Es wird vielmehr ein partieller Plattenplatz zur Verfügung gestellt, um die Änderungen zu speichern. In diesem Beispiel können auf dem Original 2 GB Daten geändert werden, bis der Snapshot ungültig wird. Der Snapshot kann nun gemountet, mit `lvextend` vergrößert und nach dem Backup mit `lvdelete` wieder gelöscht werden. Wenn

Glossar

Domain 0 (Wirt)	Physikalisches System, auf dem der XEN Kernel läuft und von dem die Gäste gesteuert und kontrolliert werden.
Domain U (Gast)	Virtuelles System, das auf derselben Hardware wie der Wirt läuft.
Initial Ramdisk	Enthält das Abbild eines Dateisystems mit dem zum Start des Systems benötigten Dateien und Treibern, quasi eine Art Mini-Linux.
Major/Minor Number	Der Kernel arbeitet beim Zugriff auf Geräte nicht mit den Namen (<code>/dev/sda</code>), sondern mit den dahinterliegenden Zahlen. Mit Major-Nummer 8 ist das SCSI-Teilsystem gemeint. Die Minor-Nummer 0 gibt an, dass es sich um das erste Gerät handelt.

Link

- ▶ [1] ORDIX News Artikelreihe „Virtualisierung mit XEN“ <http://www.ordix.de/ORDIXNews/artikelreihen.html#xen>

das LVM Volume partitioniert ist, kommt wieder `kpartx` zum Einsatz, das uns für die einzelnen Partitionen erneut Gerätedateien zur Verfügung stellt.

Fazit

Festplattenverwaltung mit XEN ist eine relativ simple Angelegenheit, da XEN hier größtenteils Bordinstrumente des eingesetzten Betriebssystems nutzt. Der Vorteil, neue Festplatten einem Gast online hinzuzufügen oder zu entfernen, vermindert die Downtime eines Gastes. Benötigt ein Gast dedizierten Storage, ist dies mit der Möglichkeit des direkten Zugriffs von Hardware-Geräten an den Gast schnell zu implementieren. Die aktuelle Version 4.0 von XEN bietet noch weitere neue Funktionen, über die wir in Zukunft weiter berichten werden. Die Entwickler scheinen sich nun auch besonnen zu haben, XEN passend zum aktuellen Linux Kernel zu entwickeln und nicht mehr an der Kernelversion 2.6.18 festzuhalten.



Christian Fertsch
(info@ordix.de).

Juni - November 2010

	Dauer	Preis	Juni		Juli					August				September				Oktober			November					
			KW 23	KW 24	KW 25	KW 26	KW 27	KW 28	KW 29	KW 30	KW 31	KW 32	KW 33	KW 34	KW 35	KW 36	KW 37	KW 38	KW 39	KW 40	KW 41	KW 42	KW 43	KW 44	KW 45	KW 46
Datenbanken																										
Datenbank-Hochverfügbarkeitslösungen für Entscheider	1 Tag	550,00 €																								
Datenbank-Modellierung	2 Tage	890,00 €																								
Oracle SQL	5 Tage	1.890,00 €																								
Oracle SQL für Experten	3 Tage	1.290,00 €																								
Oracle Datenbankprogrammierung mit PL/SQL Grundlagen	5 Tage	1.890,00 €																								
Oracle Datenbankprogrammierung mit PL/SQL Aufbau	5 Tage	1.890,00 €																								
Objektorientierung in Oracle	5 Tage	1.990,00 €																								
Oracle Datenbankadministration Grundlagen	5 Tage	1.990,00 €																								
Oracle Datenbankadministration Aufbau	5 Tage	1.990,00 €																								
Oracle Backup und Recovery	5 Tage	1.990,00 €																								
Oracle Tuning und Monitoring	5 Tage	1.990,00 €																								
Oracle AWR und ASH Analyse und Interpretation	3 Tage	1.290,00 €																								
Oracle Troubleshooting Workshop	5 Tage	1.990,00 €																								
Oracle Real Application Cluster (RAC)	5 Tage	1.990,00 €																								
Oracle 11g Neuheiten	5 Tage	1.990,00 €																								
Oracle Security	4 Tage	1.590,00 €																								
Oracle Data Guard	4 Tage	1.590,00 €																								
Oracle RMAN	4 Tage	1.590,00 €																								
Oracle Grid Control	3 Tage	1.290,00 €																								
Oracle Streams Advanced Queuing	3 Tage	1.290,00 €																								
Oracle Replikation	3 Tage	1.290,00 €																								
Oracle Migration und Patching	3 Tage	1.290,00 €																								
Oracle Capacity Planning	3 Tage	1.290,00 €																								
IBM Informix SQL	5 Tage	1.790,00 €																								
IBM Informix Dynamic Server Administration	5 Tage	1.990,00 €																								
IBM Informix Dynamic Server Tuning und Monitoring	5 Tage	1.990,00 €																								
IBM Informix Dynamic Server Backup und Recovery	3 Tage	1.290,00 €																								
IBM Informix Dynamic Server HV-Technologien unter Unix	4 Tage	1.590,00 €																								
IBM DB2 für Linux/Unix/Windows SQL Grundlagen	5 Tage	1.890,00 €																								
IBM DB2 für Linux/Unix/Windows Administration	5 Tage	1.990,00 €																								
IBM DB2 für Linux/Unix/Windows Monitoring und Tuning	3 Tage	1.290,00 €																								
MySQL Administration	3 Tage	1.190,00 €																								
Microsoft SQL Server Administration	5 Tage	1.790,00 €																								
Programmierung																										
PHP Programmierung Grundlagen	5 Tage	1.690,00 €																								
PHP Programmierung Aufbau	3 Tage	1.190,00 €																								
Perl Programmierung Grundlagen	5 Tage	1.690,00 €																								
Perl Programmierung Aufbau	5 Tage	1.690,00 €																								
Shell, Awk und Sed	5 Tage	1.690,00 €																								
AWK Intensiv-Workshop	3 Tage	1.190,00 €																								
Einführung in XML	3 Tage	1.190,00 €																								
XML Programmierung unter Java mit DOM und SAX	2 Tage	890,00 €																								
Oracle und XML	3 Tage	1.190,00 €																								
Web- und Applikations-Server																										
Apache Web-Server Installation und Administration	3 Tage	1.190,00 €																								
Tomcat Konfiguration und Administration	3 Tage	1.190,00 €																								
WebSphere Application Server Installation/Administration	3 Tage	1.390,00 €																								
Administration und Konfiguration für JBoss	3 Tage	1.190,00 €																								

Online-Anmeldung und stets aktuelle Seminarinhalte und Termine unter: <http://training.ordix.de>

	Dauer	Preis	Juni		Juli		August		September		Oktober		November														
			KW 23	KW 24	KW 25	KW 26	KW 27	KW 28	KW 29	KW 30	KW 31	KW 32	KW 33	KW 34	KW 35	KW 36	KW 37	KW 38	KW 39	KW 40	KW 41	KW 42	KW 43	KW 44	KW 45	KW 46	KW 47
Java-JEE																											
Einführung in die objektorientierte Programmierung	3 Tage	1.190,00 €																									
Java Programmierung Grundlagen	5 Tage	1.690,00 €																									
Java Programmierung Aufbau	5 Tage	1.690,00 €																									
Java GUI Entwicklung mit Swing	5 Tage	1.690,00 €																									
JEE für Entscheider	1 Tag	590,00 €																									
Einführung in JEE	3 Tage	1.290,00 €																									
JSP und Servlet Programmierung	5 Tage	1.590,00 €																									
EJB Programmierung	5 Tage	1.590,00 €																									
Web-Anwendungen mit JavaServer Faces (JSF)	5 Tage	1.590,00 €																									
Entwickeln mit dem Spring-Framework	3 Tage	1.190,00 €																									
Java Web Services	3 Tage	1.190,00 €																									
Hibernate und die Java Persistence API	5 Tage	1.690,00 €																									
Java Performance Tuning	3 Tage	1.290,00 €																									
Betriebssysteme																											
Unix/Linux Grundlagen für Einsteiger	5 Tage	1.690,00 €																									
Linux Systemadministration	5 Tage	1.690,00 €																									
Server-Virtualisierung mit XEN	3 Tage	1.190,00 €																									
Linux Hochverfügbarkeits-Cluster	3 Tage	1.290,00 €																									
OpenLDAP - Praxiseinsatz im Netzwerk	4 Tage	1.490,00 €																									
Solaris Systemadministration Grundlagen	5 Tage	1.990,00 €																									
Solaris Systemadministration Aufbau	5 Tage	1.990,00 €																									
Solaris kompakt für erfahrene Unix-Administratoren	5 Tage	1.990,00 €																									
Solaris 10 für erfahrene Systemadministratoren	5 Tage	1.990,00 €																									
Solaris Containers	3 Tage	1.290,00 €																									
Zettabyte Filesystem (ZFS) Workshop	2 Tage	890,00 €																									
IBM AIX Systemadministration Grundlagen	5 Tage	1.990,00 €																									
IBM AIX Installation, Backup und Recovery	3 Tage	1.290,00 €																									
IPv6 für Entscheider	1 Tag	590,00 €																									
IPv6 für Systemadministratoren	3 Tage	1.190,00 €																									
Systemmanagement																											
Systemüberwachung mit Nagios Grundlagen	3 Tage	1.190,00 €																									
Systemüberwachung mit Nagios Aufbau	2 Tage	890,00 €																									
Projektmanagement																											
IT-Projektmanagement	5 Tage	1.990,00 €																									
Systemisches Projektmanagement	4 Tage	1.850,00 €																									
Krisenmanagement in Projekten	2 Tage	1.100,00 €																									
IT-Projektcontrolling	3 Tage	1.290,00 €																									
IT-Management																											
IT-Architekturen	3 Tage	1.650,00 €																									
E-Business	2 Tage	1.100,00 €																									
Systemische Strategieentwicklung	3 Tage	1.650,00 €																									
IT-Management	5 Tage	1.990,00 €																									
IT-Risikomanagement	3 Tage	1.650,00 €																									
IT-Prozessmanagement	3 Tage	1.650,00 €																									
Konfliktmanagement	2 Tage	1.100,00 €																									

■ Seminar in Wiesbaden

■ Seminar in Bielefeld (IPS)

Informationen und Anmeldung

Für Informationen und Fragen zu individuell zugeschnittenen Seminaren, Ausbildungsreihen oder Inhouse-Schulungen stehen wir Ihnen gerne zur Verfügung. Auf Wunsch senden wir Ihnen auch unser komplettes Seminarprogramm zu.

ORDIX AG
 Westernmauer 12 - 16
 33098 Paderborn
 Tel.: 05251 1063-0

ORDIX AG
 Kreuzberger Ring 13
 65205 Wiesbaden
 Tel.: 0611 77840-00

Preise gelten pro Seminar pro Teilnehmer in Euro zzgl. ges. MwSt. Inhousepreise auf Anfrage.

CHRONIK 20 Jahre ORDIX

Im zweiten Teil unserer Chronik beschäftigen wir uns mit den Jahren 2000 - 2010. Durch ein solides Wachstum, erfolgreiche Projekte, bodenständige Entscheidungen und vor allem durch das Know-how und Engagement seiner Mitarbeiter hat die ORDIX AG so manche (Wirtschafts-)Krise erfolgreich umschifft.

Die ORDIX News im Wandel der Zeit



2001



2005



2010

Die starken Tochterunternehmen der ORDIX AG



Objekt- und Datenbanktechnologien sowie Projekt-, Test- und Qualitätsmanagement



IT-Management-Beratung, Beratung und Analyse technisch komplexer Projekte

ORDIX entwickelt sich ...

ORDIX erweitert das Schulungsangebot um den Bereich Projektmanagement. Das 10-jährige Firmenjubiläum wird gefeiert. Nach dem Y2K-Boom befürchten viele IT-Unternehmen zurückgehende IT-Budgets und haben Recht. ORDIX wächst jedoch weiter. Der zehnte Azubi beginnt seine Lehrjahre bei ORDIX.

ORDIX schließt einen der größten Aufträge für BMC PATROL Software in Deutschland ab. 09/11 hat auch Auswirkungen auf die IT, dennoch steigt der Umsatz der ORDIX AG im Geschäftsjahr 2001/2002 auf ca. 4,35 Mio. Euro. Erste Kontakte zu Object Systems.

Die ORDIX AG erwirbt 70 % der Anteile der Object Systems GmbH, Essen. Die Schulungsfläche im Trainingszentrum wird erneut um 60 % vergrößert. Die Mitarbeiterzahl liegt bei 65. ORDIX stellt einen Green Card-Mitarbeiter ein.

ORDIX eröffnet die Geschäftsstelle in Neu-Ulm. Das eigene IT-Magazin ORDIX News feiert 10-jähriges Bestehen. ORDIX wächst gegen den Trend bei Umsatz und Mitarbeitern.

Der Umsatz beträgt 8,28 Mio. Euro. Der zwölfte Auszubildende beginnt seine Ausbildung bei der ORDIX AG. Der zehnte Auszubildende schließt erfolgreich seine Ausbildung ab.

Christoph Lafeld und Benedikt Georgi erweitern den Vorstand der ORDIX AG. Der Umsatz steigt auf 8,77 Mio. Euro. ORDIX übernimmt weitere 15 % der Anteile von Object Systems, feiert 15-jähriges Bestehen und meldet 63 Mitarbeiter.

ORDIX wächst auf knapp 70 Mitarbeiter und steigert erneut den Umsatz. Das Tochterunternehmen Object Systems ist ebenfalls erfolgreich.

ORDIX eröffnet neues Schulungszentrum in Wiesbaden. Mit vier zusätzlichen Schulungsräumen und einem erweiterten Schulungsangebot wächst ORDIX im Gegensatz zu anderen Anbietern auch im Schulungsumfeld.

Die coniatos AG wird gegründet und ergänzt das ORDIX Portfolio. 15 Jahre ORDIX News und erstmals über 10 Mio. Euro Umsatz bei knapp 80 Mitarbeitern.

Auch in Krisenzeiten zeigt sich ORDIX wirtschaftlich stabil und wächst. Der Umsatz liegt bei knapp 11 Mio. EUR, die mit ca. 90 festangestellten Mitarbeitern erwirtschaftet werden. Auch die Töchter Object Systems und coniatos erweisen sich als stabil und profitabel.

ORDIX wird 20 Jahre.

20

20

20

20

20

20

20

20

20

20

20

... die (IT-)Welt auch!

00

Microsoft bringt Windows 2000 auf den Markt. FC Bayern München wird 100 Jahre alt. Die UMTS Lizenzen werden versteigert. Die Weltausstellung EXPO 2000 in Hannover wird eröffnet. Schwarzgeldaffäre ist Wort des Jahres. Helmut Kohl tritt als Ehrenvorsitzender der CDU zurück. Wladimir Putin wird neuer Präsident Russlands und damit Nachfolger von Boris Jelzin.

01

Die Raumstation MIR wird außer Dienst genommen. George W. Bush wird zum neuen Präsidenten der USA vereidigt. Terroristen zerstören das World Trade Center, indem sie zwei Flugzeuge in die Türme stürzen lassen. Apple stellt den ersten iPod vor.

02

Der Euro wird als neue Währung eingeführt. Flutkatastrophe entlang der Elbe. Rot-Grün gewinnt die Bundestagswahl. Sven Hannawald gewinnt als erster Skispringer alle vier Einzelspringen der Vierschanzentournee.

03

Deutschland wird Frauen-Fußball-Weltmeister in den USA. Linux Kernel 2.6 wird veröffentlicht. Letzter Flug der Concorde. Jürgen Möllemann stirbt bei einem Fallschirmsprung. Die Produktion des VW Käfer wird eingestellt. Arnold Schwarzenegger wird Gouverneur von Kalifornien.

04

In Irland tritt das weltweit erste von einem Staat erlassene Rauchverbot in Kraft. Horst Köhler wird Bundespräsident. Demonstrationen gegen Hartz IV Gesetze. Ein Tsunami im Indischen Ozean tötet ca. 230.000 Menschen.

05

Einführung der LKW Maut. Gerhard Schröder stellt Vertrauensfrage, vorgezogene Bundestagswahlen führen zur großen Koalition unter Angela Merkel. IBM kündigt Stellenabbau um 13.000 Mitarbeiter an. Jungfernflug des Airbus A380. Hurrikan Kathrina überflutet große Teile New Orleans.

06

Jahr der Informatik, Microsoft bekommt Bußgeld durch EU auferlegt. MwSt.-Erhöhung von 16 % auf 19 %. Deutschland scheidet bei der WM im eigenen Land im Halbfinale gegen Italien.

07

Der Bundestag verabschiedet die Rente mit 67 Jahren. Deutschland wird Handball-Weltmeister. In Heiligendamm findet der G8-Gipfel statt.

08

Das Briefmonopol wird aufgehoben. In Peking finden die umstrittensten Olympischen Spiele statt. 1 L Benzin kostet 1,53 Euro. Der Dow-Jones-Index verliert innerhalb eines Tages 6,98 Prozentpunkte; der höchste Tageseinbruch seit Bestehen des Aktienindex.

09

Vereidigung von Barack Obama, dem ersten afroamerikanischen Präsidenten der USA. US-Airways-Flug 1549 endet mit einer Notwasserung auf dem Hudson River in New York City. Der Automobilkonzern General Motors meldet Insolvenz an und wird teilverstaatlicht.

10

Lena Meyer-Landrut gewinnt den Eurovision Song-Contest in Oslo nach 28 Jahren wieder für Deutschland. Die Fußball-Weltmeisterschaft findet in Südafrika statt. Bundespräsident Köhler tritt zurück.

Windows 2000



iPod-Familie



Oracle Historie

2001 - Version 9i



2003 - Version 10g



2007 - Version 11g



2011 - Version 12g





Jung oder alt: Auf jeden Fall gut gefeiert

20 Jahre - immer eng am Kunden

Ob ein IT-Unternehmen, das seit 20 Jahren am Markt ist, nun alt oder jung ist, darüber lässt sich sicher streiten. Wichtig ist, dass in dieser Zeit stets erfolgreiche Projekte gemeinsam mit den Kunden durchgeführt wurden. Auf diese spannenden 20 Jahre wollte der ORDIX Vorstand, der Mitarbeiter, Kunden, alte Wegbegleiter und Partner ins Heinz Nixdorf MuseumsForum (HNF) eingeladen hatte, mit einem vergnüglichen Abend zurückblicken.

ORDIX im Fernsehen

Zur Begrüßung wurden Bilder aus einer Reportage des Senders SAT-1 gezeigt, die ORDIX ins Fernsehen gebracht hatten. Nicht direkt, sondern über den Paderborner Sternekoch Elmar Simon mit seinem Restaurant Balthasar, der bereits seit mehr als 10 Jahren das ORDIX Logo auf seiner Kochjacke prangen lässt.

Der einzige Koch in Ostwestfalen, der eine Sternauszeichnung des Guide Michelin besitzt, war auch verantwortlich für das Catering an diesem Abend.

Wie schon bei der 10-jährigen Jubiläumsfeier in Paderborn, bei der ein Kochkurs im Restaurant Balthasar und ein Museumsbesuch im HNF die Highlights bildeten, sorgten er und seine Mannschaft, deren Schürzen und Jacken nun

auch das ORDIX Logo zieren, für das Wohl von knapp 200 Gästen.

ORDIX immer eng am Kunden

Auch dieser Abend zeigte, dass selbst an solchen Tagen eines der Leitmotive von ORDIX gilt: Immer eng am Kunden.

Bei der Begrüßung der Gäste standen nur drei der vier Vorstände auf der Bühne. Entwicklungsvorstand Axel Röber hatte mit einem seiner Senior Consultants noch ein wichtiges Projektabschlussmeeting, das bis in die Abendstunden dauerte. Neben fast allen Angestellten und freien Mitarbeitern, die mit ihren Partnern eingeladen waren, hatten viele Kunden, Partner und alte Wegbegleiter den Weg ins Heinz Nixdorf MuseumsForum gefunden.



Ein bisschen wie in der Vergangenheit

Dann kam Leben in die „Museumshalle“ und die Bonner Springmäuse auf die Bühne. Ebenfalls wie bei der 10. Jahresfeier in Wiesbaden bildete die Standup- und Impro-Theatertruppe das Rahmenprogramm für einen vergnüglichen Abend. Die Zuschauer kamen dabei in den Genuß von hervorragenden Bühnenarbeitern, die als Schauspieler, Musicalstars und Comedians mit Improvisationen zu überzeugen wussten. Als die Zuschauer dann auf der Bühne auch noch mit einbezogen wurden, kam die Stimmung zum Höhepunkt.

Ein weiterer erfolgreicher Tag geht zu Ende

Während der Abend mit Musik und Tanz abgerundet wurde, konnten die Mitarbeiter auf einen ereignisreichen Tag zurückblicken. Am Morgen hatten sie einen kleinen Einblick in die 20 Jahre ORDIX Geschichte bekommen und sich im Laufe des Tages in unterschiedlichen Vorträgen und Workshops weiterbilden können. Weiterbildung wurde bei ORDIX von Anfang an groß geschrieben, was im Trainingszentrum in Wiesbaden inzwischen jedes Jahr nahezu 1.000 Teilnehmer genießen können.

ORDIX Mitarbeiter als Künstler

Für das Trainingszentrum in Wiesbaden gestalteten die ORDIX Mitarbeiter an diesem Tag auch ein Kunstwerk (siehe Bild oben), das die Gäste am Abend bewundern konnten. In Kleinstarbeit wurden die Köpfe der Mitarbeiter, das Wichtigste bei Beratungsunternehmen, nach Vorlagen gestaltet und auf eine Leinwand gebracht.

Auch wenn das Kunstwerk gelungen ist, so liegt die wahre Begabung der ORDIX Mitarbeiter doch eher in der Abwicklung erfolgreicher IT-Projekte. Während einige noch bis



in den frühen Morgen feierten, bereiteten sich andere schon wieder auf neue Aufgaben vor.

Diese Zuverlässigkeit, sowohl in der Vergangenheit als auch in der Zukunft, ist Basis für den Vertrauensvorsprung, den ORDIX bei Kunden und Partnern genießt. Und Vertrauen ist die Basis für den gemeinsamen Erfolg. In diesem Sinne warten alle auf die Feier zum 25-jährigen Bestehen eines erfolgreichen IT-Unternehmens aus Ostwestfalen.



ORDIX AG
einfach. gut. gefeiert.





28. Mai 2010
HNF, Paderborn





DOAG Regionaltreffen in Wiesbaden

Am 27.04.2010 fand das DOAG Regionaltreffen in unserem Seminarzentrum in Wiesbaden statt. Die ORDIX AG war mit einem Vortrag vertreten. Vanessa Prior referierte zum Thema „Minimierung der Ausfallzeiten beim Upgrade von Oracle Datenbanken mit Hilfe des Oracle Data Guard“.



Rolling Upgrade in Oracle

Administratoren und Systembetreuer haben den Anspruch, geplante Ausfallzeiten in Datenbanksystemen auf ein Minimum zu reduzieren.

Die Funktion „Rolling Upgrade“, die Oracle mit der Version 10.1.0.3 eingeführt hat, hilft, die Ausfallzeiten zu minimieren. Vanessa Prior erläuterte in ihrem Vortrag, wie man mit Hilfe eines Logical Data Guard, der nach einem Switchover

die Rolle der Primärdatenbank annimmt, ein Upgrade mit minimaler Ausfallzeit durchführen kann. Außerdem wurde erläutert, welche Vorüberlegungen getroffen werden müssen.

Sie hatten keine Möglichkeit, bei diesem Treffen dabei zu sein und den ORDIX Vortrag zu hören? Nehmen Sie mit uns Kontakt auf!

Wir beantworten Ihnen gerne Ihre Fragen. Sprechen Sie uns an:
info@ordix.de.

53. IUG Workshop in Hamburg

Die Informix-Gemeinde traf sich am 23. März 2010 zum 53. Workshop der Informix User Group (IUG) in Hamburg. Die Begrüßung der Teilnehmer erfolgte durch Wolfgang Kögler, Beiratsmitglied der IUG und Vorstandsvorsitzender der ORDIX AG.



ORDIX AG steuerte Vortrag bei.

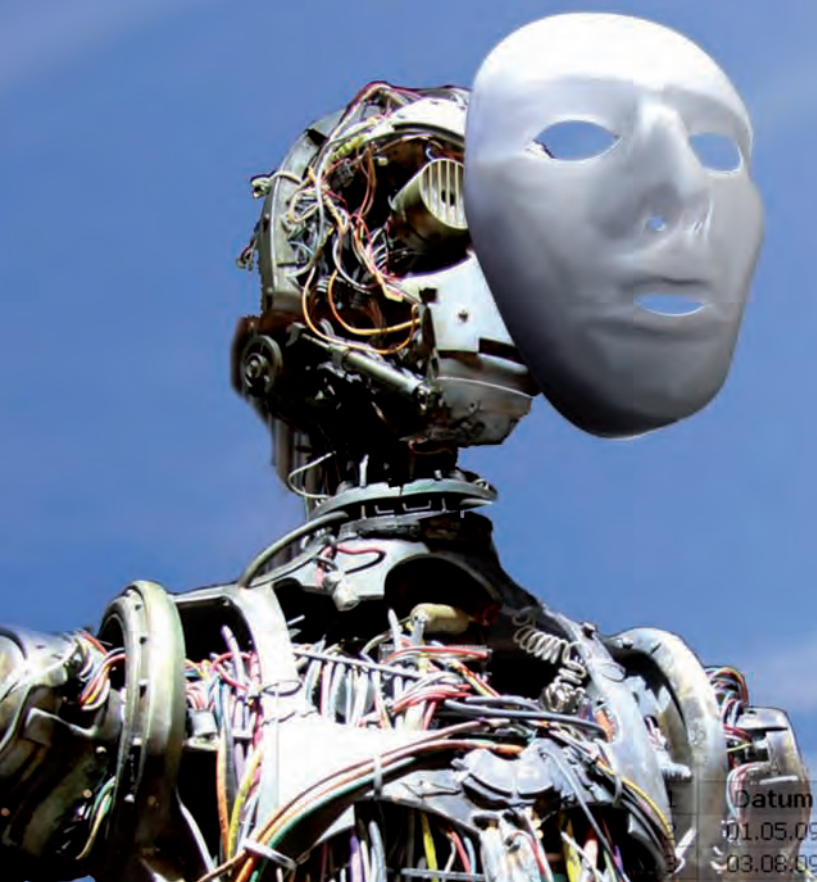
In zahlreichen Kundenprojekten und Informix Seminaren geben die ORDIX Berater und Trainer ihr Wissen an andere weiter. Diese Philosophie der Know-how-Weitergabe setzt ORDIX auch im Rahmen der IUG Workshops um.

Freuen durften sich die Teilnehmer über den Vortrag „Virtualisierung mit VMware“. Christof Amelunxen von der ORDIX AG gab in seinem Vortrag einen kurzen Überblick über aktuelle Virtualisierungstechniken im Betriebssystem-Umfeld sowie deren unterschiedliche technologische Ansätze.

Der Fokus des Vortrags lag auf der Virtualisierung im x86-Umfeld mit VMware und stellte Erfahrungen aus einem umfangreichen Virtualisierungsprojekt bei einem großen Unternehmen dar. Abschließend wurden die Einsatzgebiete für IDS in virtualisierten (x86-) Umgebungen betrachtet.

Sie hatten keine Möglichkeit, bei diesem IUG Workshop dabei zu sein und den ORDIX Vortrag zu hören?

Wir beantworten Ihnen gerne Ihre Fragen. Sprechen Sie uns an:
info@ordix.de.



ORACLE

Author	Year	Title
Adams, John	1992	The Corporation Versus the Market
Akerlof, George A.	1970	The Market for 'Lemons': Quality Uncertainty and the Market Process
Akerlof, George A.	1980	A Theory of Social Custom, of Which Specialization is a Part
Akerlof, George A.; Miyazaki, Hiroyuki	1980	The Implicit Contract Theory of the Firm
Akerlof, George A.	1984	An Economic Theorist's Book of Tales
Akerlof, George A.	1984	Gift Exchange and Efficiency Wages
Akerlof, George A.; Yellen, Janet L.	1986	Introduction
Akerlof, George A.	1987	Why Would Anyone Buy from a Stranger? Loyalty, Reputation, and the Limits of Self-Interest
Akerlof, G. A.; Yellen, Ja	1990	The Fair Wage-Effort Hypothesis

	B	C	D	E	F
Datum	Verkäufer	Artikel	Menge	EPreis	GPreis
01.05.09	Müller	Schokolade	3	0,79 €	2,37 €
03.08.09	Holzmann	Schokolade	12	1,29 €	15,48 €

Testdatengenerierung leicht gemacht

Oracle Data Masking

Dieser Artikel richtet sich an Datenbankadministratoren aus dem Testmanagement und an Security Manager im Testbereich.

Dieser Artikel beschäftigt sich mit der Testdatenerstellung unter Oracle. Ab der Version 11g existieren zwei Varianten der einfachen Testdatengenerierung aus Produktivdaten. Die eine ist eine kostenpflichtige Komponente, bietet aber auch eine größere Anzahl an Möglichkeiten. Die andere Variante nutzt das Utility Datapump. Im vorliegenden Artikel werden beide Möglichkeiten der Testdatenerstellung anhand von Beispielen dargestellt.

Definition

Das Data Masking ist die Transformation sensibler Unternehmensinformationen in realistische Testdaten. Realistisch steht hierbei für die Übernahme der Datenformate von Produktivdaten. Diese generierten Daten finden vor allem in der Software-Entwicklung Verwendung, um qualitative und quantitative Aussagen über das Verhalten von Neu- oder Weiterentwicklungen treffen zu können.

Merkmale

Als Merkmale des Data Masking gelten:

1. Die Erstellung von Testdaten auf der Basis von Produktivdatenformaten. Hierzu können Produktivdaten durch verschiedene Funktionalitäten manipuliert werden.

2. Die Verwendung von Produktivdaten durch Anonymisierung. Hierbei wird eine Anonymisierung im Sinne einer willkürlichen Zuordnung der einzelnen Attribute eines Datensatzes getroffen.
3. Die Verwendung von Data Masking ist bereits ab der Version des Enterprise Manager Grid Control 10.2.0.4 möglich. In der Version 11g Release 2 ist mit Database Control ein eigenständiges Schema hierfür hinzugekommen. Die Verwendung setzt die Lizenzierung der Enterprise Edition voraus.
4. Das Data Masking stellt eine separate, kostenpflichtige Option als so genanntes Management Pack dar!
5. Die Generierung der Testdaten ist ein irreversibler Prozess der Anonymisierung.

Funktionalitäten

Es sind folgende Funktionalitäten nutzbar.

1. **Erstellung von Bibliotheken**
mit selbst- und vordefinierten Maskierungsformaten und Definitionen zur Verwendung in verschiedenen Datenbanken.
2. **Einsatz selbstdefinierter Stored Procedures**
Hierbei handelt es sich um Funktionen, die individuell entwickelt werden und bei der Anonymisierung zum Einsatz kommen.
3. **Erstellung eines Skriptes für die Maskierung**
Durch den Einsatz des „Mask Wizard“ wird ein PL/SQL-Maskierungsskript erstellt, das alle nötigen Informationen und Statements für die Durchführung des Data Masking in Form von Stored Procedures enthält. Dieses Skript wird später durch einen Job ausgeführt und übernimmt dadurch die eigentliche Generierung der Testdaten für die Testumgebung.

Die Erstellung eines Prüfereports (Impact Report) ermöglicht, auf Fehler und/oder Prämissen hinzuweisen. Hierzu gehören Prüfungen auf Eindeutigkeit, Maskenformate, Speicherplatz, Check Constraints und Default-Werte für Partitionen.

Ablauf

Die Maskierung der Daten erfolgt grundsätzlich in mehreren Schritten:

Schritt 1:
Zuerst werden die sensitiven Daten identifiziert. Diese Analyse führt der Security Manager der Datenbank durch.

Schritt 2:
Im Anschluss daran definiert ein Business Analyst oder Applikationsverantwortlicher die Datentypmaske der relevanten Spalten und fügt einen entsprechenden Kommentar hinzu. Beispiele finden Sie in Abbildung 1.

Schritt 3:
Im nächsten Schritt erfolgt das Klonen der Produktionsdatenbank als Staging-Datenbank. Dies geschieht durch den DBA.

Schritt 4:
Jetzt werden die Definitionen für die Maskierungsformate mit Hilfe des Enterprise Manager Data Masking Pack durch den DBA erstellt. Darin findet sich schon eine Vielzahl an vordefinierten Formaten (siehe Abbildung 2).

```
COMMENT ON COLUMN ora00.mitarbeiternr IS 'ANONYMISIERUNG: Zufallsfunktion'
```

```
COMMENT ON COLUMN ora00.mitarbeitername IS 'ANONYMISIERUNG: Produktivdatenfunktion'
```

Abb. 1: Beispiele für Kommentare.

Array List	Liste mit Zufallswerten
Fixed Number	zu verwendende Zahl
Fixed String	zu verwendende Zeichenkette
Random Dates	Zufallsdatum
Random Digits	Zufallszeichen
Random Numbers	Zufallszahlen
Random Strings	Zufallszeichenketten
Shuffle	Originaldaten
Substring	Zufallswerte aus definierter Spalte
Table Column	Zufallswerte aus definierter Spalte
User defined Functions	Benutzerdefinierte Funktionen

Abb. 2: Vordefinierte Maskierungsformate im Enterprise Manager.



Abb. 3: Transformation von Daten einer Tabelle.

Schritt 5:
Der nächste Schritt beinhaltet das Ausführen eines Jobs, der in der Bibliothek enthalten ist. Hierdurch wird ein Skript generiert, das letztendlich dafür sorgt, dass

- Eindeutigkeit existiert
- Formatmasken zu den zugehörigen Daten passen
- eine Prüfung des benötigten Speicherplatzes erfolgt
- Warnungen durch Check Constraint Verletzungen erfolgen
- es eine Prüfung auf Default-Partitionen gibt


```
[<Schemaname>.<Tabellenname>.<Spaltenname>:
[<Schemaname>.<Packagename>.<Funktionsname>
```

Abb. 4: Die Syntax des Parameters REMAP_DATA.

```
create or replace package ora00.pkg_mask
as
  function fct_geh (p_origGehalt NUMBER)
  return number;
end;
/
create or replace package body ora00.pkg_mask
as
  function fct_geh (p_origGehalt NUMBER)
  return number
is
  p_newGehalt NUMBER(9,2) := 0;
  begin
    p_newGehalt := dbms_random.value(1000000.00,9999999.99);
    return p_newGehalt;
  end;
end;
/
```

Abb. 5: Funktion zur Generierung von Zufallswerten.

```
Export:

expdp ora00/ora00
tables=ora00.mitarbeiter
dumpfile=tbl_mitarbeiter.dmp
reuse_dumpfiles=y
directory=tmp_dir remap_data=ora00.mitarbeiter.gehalt:ora00.
pkg_mask.fct_geh

Import:

impdp ora00/ora00
tables=ora00.mitarbeiter
dumpfile=tbl_mitarbeiter.dmp
directory=tmp_dir remap_data=ora00.mitarbeiter.gehalt:ora00.
pkg_mask.fct_geh
```

Abb. 6: Beispiel für die Nutzung des Datapump bei der Datenmaskierung.

Link

- ▶ [1] ORDIX News Artikel „Neuerungen Data Pump in Oracle 11g“
http://www.ordix.de/ORDIXNews/4_2008/Datenbanken/oracle11g_data_pump.html

Der daraus resultierende Prüfreport (Impact Report) sollte fehlerfrei sein.

Schritt 6:

Abschließend wird das zuvor generierte Skript ausgeführt und die Staging-Datenbank zur Testdatenbank geklont.

Das Beispiel in Abbildung 3 zeigt die Transformation der Daten beim Data Masking. Hier-

zu werden drei Spalten mit ihren Datenformaten beibehalten und anschließend nach bestimmten Transformationsschritten maskiert. Die vierte Spalte (Gehalt) bleibt in ihren Daten unverändert. Dennoch ist ein Rückschluss auf die Originaldaten nicht möglich.

Datenmaskierung mit Datapump

Wie schon in der ORDIX News Artikelreihe „Oracle Database 11g“ [1] erwähnt wurde, ist durch die Einführung des Übergabeparameters REMAP_DATA auch eine Datenmaskierung bei der Nutzung des Tools Datapump mittels `expdp/impdp` möglich.

Die Syntax des Parameters finden Sie in Abbildung 4.

Hierdurch existiert nun ab 11g eine lizenzfreie Möglichkeit der Testdatengenerierung. Die einzige Voraussetzung hierfür ist das Erstellen der jeweiligen Funktionen. Es muss dabei auch der gleiche Datentyp berücksichtigt werden. Somit ist auch der Validität eines Testszenarios genüge getan. Die zu nutzende Funktion sollte über ein Package definiert werden (siehe Abbildung 5).

Die Kommandozeile für den Aufruf des Datapump Utility für diesen Fall finden Sie in Abbildung 6.

Fazit

Sowohl mit der Option Data Masking als auch mit der Nutzung des neuen Parameters REMAP_DATA im Tool Datapump existiert jeweils eine gute und performante Option der Testdatengenerierung. Die Entscheidung für eine der beiden unterschiedlichen Möglichkeiten wird in dem Spannungsfeld zwischen Kosten und Qualität bestimmt. Für die Bereitstellung von Testdaten für Modultests durch den Entwickler wird sicherlich die Nutzung der Datapump-Option im Vordergrund stehen. Sind darüber hinaus neben fachlichen Tests auch Performance-Analysen notwendig, sollte der Einsatz der Option Data Masking in Erwägung gezogen werden.



Klaus Günther
(info@ordix.de).

Larry Ratlos

Larry hat ein PHP-Problem mit der Select-Box



Heute steht Larry vor einer neuen Herausforderung. Das Intranet in Larry's Firma ist an eine Datenbank angebunden. Nun bekommt Larry von seinem Chef die Aufgabe, eine HTML-Seite zu erstellen, auf der man die Datenbankserver dynamisch aus einer Select-Box auswählen kann. Hierfür schreibt Larry ein kleines PHP-Script.

Datenbank- statt Servernamen

Als Basis liegt Larry eine Konfigurationsdatei (siehe Abbildung 1) mit den Server- und Datenbanknamen vor, die in der Select-Box angezeigt werden sollen - jeweils durch ein „|“-Zeichen getrennt.

Um die Selectbox im Intranet auszugeben, benutzt Larry HTML-Code, den er mit seinem PHP-Script erzeugt (siehe Abbildung 2). Die Select-Box soll eine Auswahl der Datenbanken anhand der Anzeigenamen (nicht der Servernamen!) ermöglichen. Als Name für die Optionen sollen die richtigen Datenbanknamen eingefügt werden. Larry bekommt beim Aufruf der Seite keinen Fehler. Jedoch bleibt die Anzeige in seiner Select-Box leer!

Können Sie Larry helfen?

Schicken Sie Larry bitte den richtigen PHP-Code, damit er die korrekte Anzeige in seiner Select-Box erhält. Auf Ihren Lösungsvorschlag freut sich Larry **bis zum 16. Juli 2010** an kniffel@ordix.de.

```
prod_db_1|Produktionsdatenbank 1
entw_db_1|Entwicklungsdatenbank 1
entw_db_2|Entwicklungsdatenbank 2
```

Abb. 1: Die Konfigurationsdatei db_name.txt.

```
<?php
$lines = file("./db_name.txt");

for ($j = 0; $j < count($lines); $j++) {
    $db_array[$j][0] = explode("|",$lines[$j]);
}

echo '
<label for="db_server">Datenbank:</label>
<select name="db_server">';

for ($i = 0; $i < count($db_array); $i++) {
    echo '<option name="'. $db_array[$i][0].'">'. $db_array[$i][1]. '</option>';
}

echo '</select>';
?>
```

Abb. 2: Das PHP-Script larry_ist_ratlos.php liest die Konfigurationsdatei aus und erstellt das HTML-Formular.

Lösung aus der Ausgabe 1/2010

- 1) ORDIX
- 2) 1
- 3) 1234
- 4) 12345
- 5) ORDIX 4711
- 6) Larry ist ratlos.
- 7) [Ljava.lang.String;@3e25a5und verwundert.

Die Ausgaben in 1) bis 6) sollten eindeutig sein.

Zu 7) Hier wird der Methode mit „s2“ ein Array von Objekten (String-Objekte) übergeben. Das bewirkt, dass das Array als Objekt behandelt wird und dass bei der Ausgabe die toString-Methode der Klasse Object aufgerufen wird. Dadurch wird die String-Repräsentation der Instanz ausgegeben: „Klasse@HashCodeInHex“.

Larry bedankt sich bei Herrn Detering und Herr Jäschke für die richtigen Lösungen und wünscht viel Freude mit den kleinen Präsenten.



Perl und DB2 im Verbund (Teil II)

Übersicht durch Ordnung

Dieser Artikel richtet sich an DB2-Administratoren, die daran interessiert sind, mit Perl gut strukturierte und wartbare Werkzeuge für ihre tägliche Arbeit zu erstellen.

Seit Beginn der DB2-Zeiten ist „LIST APPLICATIONS“ - meist mit dem Zusatz „SHOW DETAIL“ - das wahrscheinlich am häufigsten abgesetzte Kommando. Der Informationsgehalt der Ausgabe ist jedoch verhältnismäßig gering und Zeilenumbrüche stören die Übersicht. Mittlerweile stehen jedoch vielfältige Informationen auch über diverse administrative Views zur Verfügung. Perl hilft uns auch hier, unseren Werkzeugkasten schnell und flexibel zu erweitern.

Problemkind LIST APPLICATIONS

Wie findet der Administrator heraus, was auf seinem Datenbankserver gerade los ist? Jahrelang hieß die Antwort immer `LIST APPLICATIONS SHOW DETAIL`, oft gefolgt von einer ganzen Kette von `grep`- und gegebenenfalls `awk`-Aufrufen, gesammelt entweder in der Shell History oder in einzelnen Shell-Skripten. Das Ergebnis liefert bestenfalls einen Hinweis darauf, dass z. B. eine Anwendung im Status `LOCK WAIT` oder aber schon seit einer langen Zeit im Status `Executing` ist. Für detaillierte Informationen über eine einzelne Anwendung muss dann ein Snapshot der Anwendung mit `GET SNAPSHOT FOR APPLICATION APPLID <kryptische ID aus LIST APPLICATIONS>` gefertigt werden, in dem man dann wieder mit `grep` nach den interessanten Zeilen suchen muss.

Die Abbildung 1 zeigt den Output von `LIST APPLICATIONS SHOW DETAIL` auf einer DB2 V9.7. Beachten Sie, dass hier gerade mal zwei aktive Clients auf einer Datenbank verbunden sind. Die restlichen sechs Einträge stammen z. B. vom „Self Tuning Memory Manager“ oder dem „Workload Manager Daemon“, die immer gestartet sind, uns aber in der Regel nicht wirklich interessieren. Abbildung 6 zeigt den deutlich übersichtlicheren Output unserer Lösung.

Administrative Views

In den 7er-Versionen begann IBM mit der Einführung von Tabellenfunktionen zum direkten Zugriff auf Snapshot-Daten. Darauf basierend entstanden bis zur aktuellen Version 9.7 immerhin 64 administrative Views mit insgesamt 1.200 (teilweise redundanten) Spalten. Reichlich Informationen also, die man für in-

dividuelle Auswertungen nutzen kann (siehe Abbildung 2). Für unser Beispiel interessieren folgende Views:

- SYSIBMADM.SNAPAPPL
- SYSIBMADM.SNAPAPPL_INFO
- SYSIBMADM.APPLICATIONS
- SYSIBMADM.APPL_PERFORMANCE

Jede einzelne der insgesamt 179 Spalten ist im Information Center ausführlich dokumentiert. Allen gemeinsam ist die Spalte **AGENT_ID**, die wir zur sinnvollen Verknüpfung in Abfragen nutzen können und die zum Element **APPLICATION HANDLE** von **LIST APPLICATIONS** korrespondiert.

Ohne Monitor Switches nur mit halber Kraft

Ähnlich wie die unterschiedlichen Snapshot-Varianten benötigen auch die administrativen Views Zugriff auf Monitorelemente, die eingeschaltet sein sollten. Andernfalls sind die entsprechenden Spalten leer. Der Overhead für die Monitore wird unterschiedlich bewertet. Sollte das Einschalten der Monitor-Elemente zu Performance-Problemen führen, ist dies eher ein Hinweis, dass die Ressourcen insgesamt entweder knapp sind oder schlecht genutzt werden. Die Monitore schalten Sie dauerhaft in der DBM-Konfiguration ein (siehe Abbildung 3).

Auswahl der Elemente

Aufgrund der großen Auswahl der verfügbaren Monitorelemente erscheint es zunächst schwierig, die vermeintlich richtigen auszuwählen. Unser Tipp: Blättern sie ein wenig durch die Views (z. B. mit der Steuerzentrale) und wählen Sie jeweils nur wenige, zusammenpassende Elemente aus. Entweder Sie erzeugen dann mehrere Skripte oder Sie entscheiden sich, die entsprechende Abfrage z. B. über Kommandozeilen-Argumente auszuwählen. Für unser Beispiel verwenden wir folgende Informationen:

- Aus SYSIBMADM.APPLICATIONS: AGENT_ID, APPL_NAME, AUTHID, APPL_STATUS, CLIENT_NNAME
- Aus SYSIBMADM.SNAPAPPL_INFO: WORKLOAD_ID <> 0 (zur Unterscheidung von System- und Anwendungsprozessen)
- Aus SYSIBMADM.SNAPAPPL: UOW_COMP_STATUS, ROWS_READ, ROWS_WRITTEN
- Aus SYSIBMADM.APPL_PERFORMANCE: PERCENT_ROWS_SELECTED

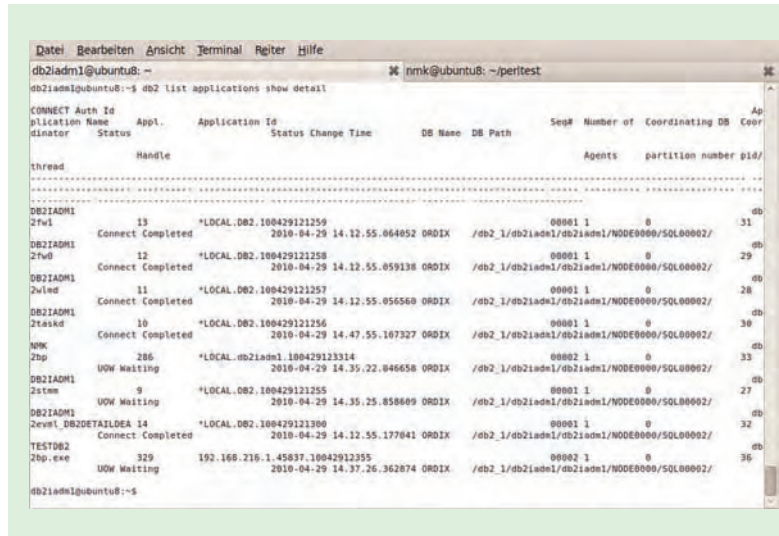


Abb. 1: Output von LIST APPLICATIONS SHOW DETAIL. Das geht schöner!

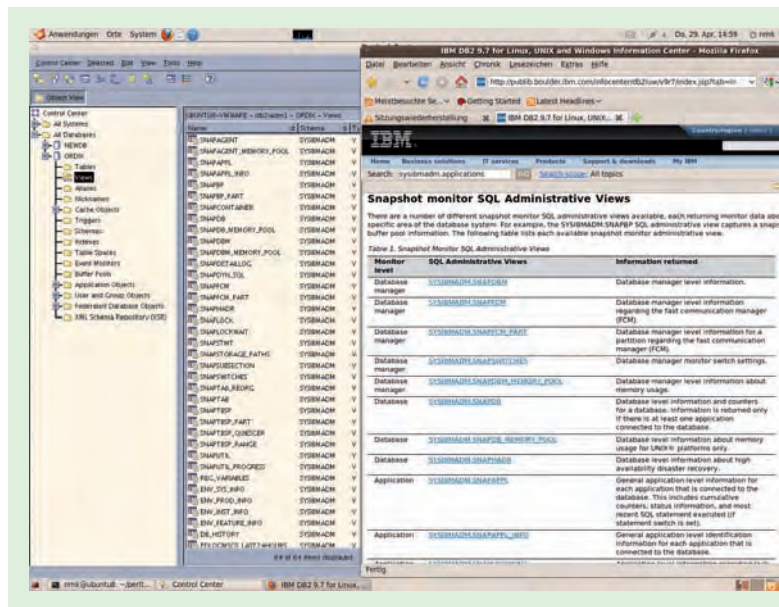


Abb. 2: SYSIBMADM-Views im Control Center und im Information Center.

UPDATE DBM CFG USING DFT_MON_BUFFPOOL ON DFT_MON_SORT ON DFT_MON_LOCK ON DFT_MON_TIMESTAMP ON DFT_MON_UOW ON

Abb. 3: Einschalten der Monitor Switches.

Glossar

CPAN Comprehensive Perl Archive Network: Online-Repository für Perl-Module, Anwendungen und Dokumentation.


```

Datei Bearbeiten Ansicht Terminal Hilfe
#!/usr/bin/perl -w
use strict; # zwingt uns u.a. zur Deklaration unserer Variablen
use DBI;
use DBD::DB2::Constants;
use DBD::DB2;
use SQL::Library;

my $database_name = $ARGV[0] or die "usage: $0 <database name>\n";
my $dsn = "dbi:DB2:$database_name";
my $database_user = ""; # ich bin lokal und verwende den Instanzuser
my $password = ""; # deswegen brauche ich User und Passwort nicht!
my $dbh; # Diese Variable wird das Objekt mit dem Database-Handle enthalten

# SQL::Library initialisieren
my $sqlLib = new SQL::Library { lib => "admin_queries.ini" } or die $!;
my $selectstring = $sqlLib->retr('APPL_OVERVIEW') or die $!;

# Datenbank-Connect
$dbh = DBI->connect( # Verbindung aufnehmen
    $dsn,
    $database_user,
    $password,
    {
        db2_txn_isolation => SQL_TXN_READ_UNCOMMITTED, # aus DBD::DB2::Constants
        db2_info_programname => "DBI-Skript" # Für LIST APPLICATIONS
    }
) or die "Konnte nicht zur Datenbank verbinden". $dbh->errstr;

my $sth = $dbh->prepare($selectstring);
$sth->execute();

# Einfache Ausgabe formatieren
print(
    "%6s%14s%10s%10s%10s%12s%10s%10s%14s\n",
    "ID", "AppName", "User", "Status", "Client",
    "Zuletzt", "RowR", "RowW", "Selektivität"
);

while ( my @row = $sth->fetchrow_array ) {
    print( "%6d%14s%10s%10s%10s%12s%10d%10d%13s\n", @row );
}

$sth->finish;
$dbh->disconnect;
    
```

Abb. 4: Das vollständige Perl-Skript.

```

Datei Bearbeiten Ansicht Terminal Hilfe
# admin_queries.ini
# Grobe Übersicht aller Anwendungen:
[APPL_OVERVIEW]
SELECT app.agent_id, app.appl_name, app.authid,
       app.appl_status, app.client_name as client,
       snapa.uow_comp_status, snapa.rows_read, snapa.rows_written,
       perf.percent_rows_selected

FROM SYSIBMADM.APPLICATIONS app,
     SYSIBMADM.SNAPAPPL snapa,
     SYSIBMADM.SNAPAPPL_INFO info,
     SYSIBMADM.APPL_PERFORMANCE perf

WHERE info.workload_id <> 0
      AND app.agent_id=snapa.agent_id
      AND info.agent_id=snapa.agent_id
      AND perf.agent_id=snapa.agent_id

ORDER BY snapa.rows_read
# Kommentierung ist möglich und erwünscht
#
# Top-Ten nach rows_read:
[ROWS_READ_TOP]
SELECT app.agent_id, app.appl_name, app.authid...
FROM ...
WHERE ...
ORDER BY snapa.rows_read DESC
FETCH FIRST 10 ROWS ONLY

### weitere queries ###
[ROWS_WRITTEN_TOP]
SELECT ...
    
```

Abb. 5: Eine SQL::Library-Konfiguration mit zwei Queries.

Die resultierende Abfrage sehen Sie im Beispiel in Abbildung 4.

Das liefert schon eine brauchbare Übersicht über die aktiven Anwendungen einer Datenbank. Um die Systemprozesse auszuschließen, haben wir den Umstand genutzt, dass auch ohne explizite Verwendung des Workload Manager alle Prozesse implizit einer Workload zugeordnet werden. Die Workload „0“ bezeichnet damit Systemprozesse, die uns hier nur die Übersicht zerstören würden.

Viele Queries – große Unordnung

Mit dem Know-how aus dem ersten Teil dieser Reihe [1] sollte es uns nun leicht fallen, ein oder mehrere entsprechende Perl-Skripte zu erstellen. Spätestens ab der dritten Query wird der Quelltext allerdings recht unübersichtlich.

Für jede Query-Anpassung muss wieder der Programmcode editiert werden. Einzelne Abfragen möchte man evtl. auch in anderen Programmen verwenden und beim Editieren der SQL-Statements vergisst man leicht mal ein Anführungszeichen. Eine Lösung besteht darin, die SQL-Abfragen vom eigentlichen Programmcode zu trennen. Abstraktion schafft hier Flexibilität und Übersicht.

Ordnung schaffen mit SQL::Library

Das Perl-Modul `SQL::Library` bietet genau die gesuchte Funktionalität mit einer einfach zu handhabenden Schnittstelle. Sie erlaubt die Verwendung von SQL-Bibliotheken im Stil einer Windows INI-Datei, wobei jeder Abfrage ein eindeutiger Bezeichner zugeordnet wird.

Die ganze Arbeit von `SQL::Library` besteht daraus, einer Perl-Variablen an entsprechender Stelle den Text der Abfrage zuzuweisen.

Die Installation erfolgt entweder über das CPAN-Modul mittels `install SQL::Library` oder über den üblichen Weg mit manuellem Download und der Befehlsfolge `perl Makefile.pl; make; make install`.

Abbildung 5 zeigt einen beispielhaften Auszug aus einer `SQL::Library`-Datei. In Abbildung 4 binden wir eine Abfrage aus der Library in unser Skript ein. Schon bei einer einzigen verwendeten Abfrage ist unser Skript deutlich schlanker und damit wartbarer geworden. Ein guter, erster Schritt.

Mehr Abstraktion, mehr Ordnung

Wir haben jetzt die SQL-Abfragen aus unserem Quellcode entfernt. Allerdings ist unser Skript noch recht starr auf eine Datenbank fixiert. Mit dem bisher Erlernten würden Sie ggf. für jede Datenbank ein Skript mit den jeweils geänderten Zugangsdaten erstellen. Viel einfacher wäre es, die Zugangsdaten für verschiedene Datenbanken in einer zentralen XML-Datei zu speichern und dann nur den Datenbanknamen über die Kommandozeile zu übergeben.

Im nächsten Teil beschäftigen wir uns mit dem Modul `XML::Simple`, das uns einfache Methoden hierfür zur Verfügung stellt.



Norbert Munkel
(info@ordix.de).

ID	AppName	User	Status	Client	Zuletzt	RowR	RowW	Selektivität
23	db2bp	NMK	UOWAIT	sless11	UOWCOMMIT	85	0	75,29
27	db2bp	DB2IADM1	UOWAIT	ubuntu8	UOWROLLBACK	498	0	97,59
31	java	NMK	UOWAIT	ubuntu8		0	43	30,23
32	java	NMK	UOWAIT	ubuntu8	UOWCOMMIT	2446	0	15,12
167	DBI-Skript	DB2IADM1	UOWEXEC	ubuntu8		0	11	0,00

ID	AppName	User	Status	Client	Zuletzt	RowR	RowW	
40	javaw.exe	TESTDB2	UOWAIT	SANTANA	UOWCOMMIT	2326	0	71,63
85	db2bp.exe	NMK	UOWAIT	SANTANA	UOWCOMMIT	221	0	91,85
170	DBI-Skript	DB2IADM1	UOWEXEC	ubuntu8		0	12	0,00

Abb. 6: Der Output unseres Skripts. Deutlich übersichtlicher!

Links

- ▶ [1] ORDIX News Artikel „Perlen (nicht nur) für DB2-DBAs“:
http://www.ordix.de/ORDIXNews/1_2010/Datenbanken/perl_db2_im_verbund.html
- ▶ [2] Dokumentation zu SQL::Library
<http://search.cpan.org/~dgorley/SQL-Library-0.0.3/lib/SQL/Library.pm>
- ▶ [3] Download SQL::Library
<http://search.cpan.org/~dgorley/SQL-Library-0.0.3/>
- ▶ [4] Dokumentation der administrativen Views
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.mon.doc/doc/r0006117.html>
- ▶ [5] Seminarempfehlungen:
Perl-Programmierung Aufbau:
<http://training.ordix.de/siteengine/action/load/kategorie/Programmierung/nr/238/index.html>
IBM DB2 für Linux/Unix/Windows SQL Grundlagen:
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/281/index.html>
IBM DB2 für Linux/Unix/Windows Monitoring und Tuning
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/644/index.html>

Seminarempfehlung: IBM DB2 für Linux/Unix/Windows Administration

- ▶ Informationen/Online-Anmeldung: <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/279/index.html>

In diesem Seminar werden Sie mit der Installation, Konfiguration und Administration eines IBM DB2 Datenbanksystems für Linux/Unix/Windows vertraut gemacht. Sie lernen anhand der vorhandenen Werkzeuge, DB2-Datenbanken zu verwalten und zu analysieren. Das Seminar wird sowohl unter Windows als auch unter Unix/Linux gehalten.

Seminarinhalte

- Die DB2 Architektur
- Einführung in die GUI Tools (Control Center, CLP)
- DB2 Installation, Anlegen einer Datenbank
- Datenbank-Objekte verwalten
- Backup und Recovery
- Reorganisation
- Sperrmechanismen (Locking und Monitoring)
- Security: Benutzerverwaltung, Zugriffsschutz
- Datenbank-Tools
- Tipps und Tricks
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

- 02.08. - 06.08.2010 in Wiesbaden
- 04.10. - 08.10.2010 in Wiesbaden
- 13.12. - 17.12.2010 in Wiesbaden

Seminar-ID: DB-DB2-02

Dauer: 5 Tage

Preis pro Teilnehmer: 1.990,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.791,00 € (zzgl. MwSt.)

Wir führen unsere Seminare auch jederzeit an einem geeigneten Ort Ihrer Wahl durch und bringen, wenn nötig, auch das entsprechende Equipment mit. Informieren Sie sich am besten im Internet über unsere Kundenseminare und die mobilen Schulungen: <http://training.ordix.de>.



Java Server Faces 2.0

Java Server Faces 2.0 – der neue Standard?

Dieser Artikel richtet sich an erfahrene Java-Webentwickler, die bisher schon mit JSF gearbeitet haben, aber auch an Neueinsteiger in die JSF-Welt.

Dies ist eigentlich eine rhetorische Frage, denn Java Server Faces (JSF) 2.0 wird Standard der kommenden Java EE 6 Version sein. Aber es wird wohl auch von den Entwicklern als Standard akzeptiert und benutzt werden, da schon JSF 1.2 sich sehr großer Beliebtheit erfreute und auch schon zahlreiche Ergänzungen zur Standard-Implementierung existieren.

Einführung

Es liegen gut drei Jahre zwischen der Veröffentlichung von JSF 1.2 und der aktuellen Version 2.0. In der Zeit hat sich JSF stark gemauert und ist zu einem sehr guten und weit verbreiteten Framework gereift. Zahlreiche Ergänzungen wie RichFaces, IceFaces und JBoss Seam sind in dieser Zeit entwickelt worden und haben die Web-Entwicklung stark vereinfacht. JSF ist somit zu einer der wichtigsten Technologien in der Java-Web-Entwicklung geworden.

Die vielen zum Teil konkurrierenden Ergänzungen führten aber immer häufiger zu Problemen, da sie teilweise nicht kompatibel zueinander waren (und sind). Durch die Version 2.0 wurde vieles vereinheitlicht und auch bisher nicht standardisierte Funktionen wurden in den JSF-Standard mit aufgenommen.

Facelets

Die auch bisher schon häufig verwendete View-Technologie Facelets wurde nun Teil des Standards. Aber die Entwicklung ging noch weiter. Das bisher bevorzugte JSP wurde vom Thron gestürzt. Facelets ist nun die bevorzugte View-Technologie und JSP in diesem Zusammenhang sogar „deprecated“. Facelets bietet die Möglichkeit, Seiten und Inhalte auf Templates basierend aufzubauen.

Seitenfragmente lassen sich einfach zentral ablegen und in mehrere Seiten integrieren. Das geht sogar so weit, dass man Komponenten erstellen kann, ohne eine Zeile Java-Code zu verfassen.

Auf diesen Aspekt wird im Verlauf des Artikels näher eingegangen, wenn es um die `Composite Components` geht.

Ein Leben ohne faces-config.xml

Für diejenigen, die Konfigurationsdateien noch nie leiden konnten, hier nun eine gute Nachricht: Mit JSF 2.0 ist es möglich, in den meisten Fällen auf die `faces-config.xml` zu verzichten. Dies wird einerseits durch den verstärkten Einsatz von Annotations erreicht, aber auch durch das Prinzip „Convention over Configuration“. Dies bedeutet, dass es ein Default-Verhalten gibt, wenn keine explizite Konfiguration angegeben wurde. Wer weiterhin die `faces-config.xml` benutzen möchte, kann dies natürlich tun.

Wie dies funktioniert, schauen wir uns am folgenden Beispiel an. In Abbildung 1 sehen wir zwei ManagedBeans. Die eine Bean enthält die Daten (Model), die andere die Action-Methode(n) (Controller). In der alten JSF-Version mussten beide Beans in der `faces-config.xml` eingetragen werden, damit sie in den JSF-Seiten benutzt werden konnten. Dies erfolgt nun über die Annotation `@ManagedBean`. Wenn man keinen Namen angibt, über den man die Bean ansprechen kann, wird als Name der Klassenname verwendet, bei dem der erste Buchstabe zu einem Kleinbuchstaben umgewandelt wird. In diesem Fall gilt das für die Klasse `EmailController`, die man in der XHTML-Seite unter dem Namen `emailController` ansprechen kann (siehe Abbildung 2). Der `ManagedBean CustomerBean` haben wir explizit den Namen `mitarbeiterBean` gegeben, unter dem man die Bean ansprechen kann (siehe Abbildung 2).

Mit den Annotations kann man ebenfalls den Scope angeben, in dem eine Bean läuft. In unserem Fall läuft die `CustomerBean` im Session Scope und die Bean `EmailController` im Request Scope (siehe Abbildung 1). Werden keine Annotations angegeben, so unterstellt JSF 2.0 den Request Scope. Neu hinzugekommen ist in JSF 2.0 der View Scope (`@ViewScoped`). Dieser bezieht sich genau auf eine View. Beans, die in diesem Scope abgelegt werden, sind so lange gültig, wie eine View aktiv ist. Dies kann nur einen Request lang dauern oder beliebig viele. Erst wenn auf eine andere View navigiert wird, wird dieser Scope geleert.

Die Annotation `@ManagedProperty` ist ebenfalls neu. In unserem Beispiel ist die `ControllerBean` (EmailController) stateless und liegt im Request-Scope. Beim Instanzieren wird die `CustomerBean` (die im Session-Scope läuft) mit Hilfe der neuen Annotation

```
@ManagedBean (name="mitarbeiterBean")
@SessionScoped
public class CustomerBean {
    private String vorname;
    private String nachname;
    private String email;

    //TODO Getter und Setter
}

@ManagedBean
@RequestScope
public class EmailController {

    @ManagedProperty(value="#{mitarbeiterBean}")
    private MitarbeiterBean maBean;

    public String berechneEmail() {
        StringBuffer buffer = new StringBuffer();

        buffer.append(maBean.getVorname().substring(0,1));
        buffer.append(maBean.getNachname().substring(0,2));
        buffer.append("@ordix.de");

        maBean.setEmail(buffer.toString());

        return "ausgabe";
    }
}
```

Abb. 1: Managed Beans.

```
...
<h:body>
    <h:form>
        <h:panelGrid columns="2">
            Vorname:
            <h:inputText value="#{mitarbeiterBean.vorname}"/>

            Nachname:
            <h:inputText value="#{mitarbeiterBean.nachname}"/>
        </h:panelGrid>

        <h:messages/>

        <h:commandButton action="#{emailController.berechneEmail}"
            value "Berechne E-Mail" />
        </h:form>
    </h:body>
...

```

Abb. 2: eingabe.xhtml.


```

...
<h:body>
  <h:form>
    Ordix E-Mail Adresse:
    <h:outputText value="#{mitarbeiterBean.email}"/>
    <h:commandButton action="eingabe" value "Zurück" />
  </h:form>
</h:body>
...

```

Abb. 3: ausgabe.xhtml.

```

...
<h:body>
  <h:form>
    <h:panelGrid columns="2">
      Vorname:
      <h:inputText id="vor" value="#{mitarbeiterBean.vorname}"/>
      Nachname:
      <h:inputText id="nach" value="#{mitarbeiterBean.nachname}">
        <f:ajax execute="@this vor" render="voll"/>
      </h:inputText>
    </h:panelGrid>
    <h:outputText id="voll" value="#{mitarbeiterBean.vorname}
#{mitarbeiterBean.nachname}"/>
    <h:outputText id="email" value="#{mitarbeiterBean.email}"/>
    <h:messages/>
    <h:commandButton action="#{emailControler.berechneEmail}"
      value "Berechne E-Mail">
      <f:ajax execute="@this vor nach" render="email"/>
    </h:commandButton>
  </h:form>
</h:body>
...

```

Abb. 4: eingabeAjax.xhtml.

```

@ManagedBean (name="mitarbeiterBean")
@SessionScoped
public class CustomerBean {
    @NotNull(message="Bitte geben sie einen Vornamen ein")
    private String vorname;
    @NotNull(message="Bitte geben sie einen Nachnamen ein")
    @Size(min=2 message="Der Nachname muss mindestens 2 Zeichen
haben")
    private String nachname;
    private String email;
    //TODO Getter und Setter
}

```

Abb. 5: Validierung.

`@ManagedBean` injiziert, so dass auf sie zugegriffen werden kann (siehe Abbildung 1).

Auch die Navigation ist in vielen Fällen nun ohne Konfigurationsdatei möglich. Der Rückgabewert der Action-Methode kann nun gleich das Navigationsziel sein. In unserem Beispiel ist der Rückgabewert der Methode `berechneEmail()` der String ausgabe. Somit wird nach einer Datei `ausgabe.xhtml` gesucht und dorthin navigiert (siehe Abbildung 1).

Eine weitere Möglichkeit besteht darin, direkt im Action-Attribut des CommandButtons oder der CommandLinks einen View-Namen anzugeben, zu dem navigiert werden soll. In unserem Beispiel ist dies in der Abbildung 3 zu sehen. Dort steht beim `CommandButton` `action="eingabe"`, womit beim Drücken des Buttons zu `eingabe.xhtml` navigiert wird.

AJAX inklusive

Bisher war AJAX in JSF-Programmen nur durch Erweiterungen verfügbar. Allerdings waren die verschiedenen Ansätze nicht kompatibel zueinander.

Aus diesem Grund standardisiert JSF 2.0 den AJAX-Support. Um die neue AJAX-Integration in JSF vorzustellen, ändern wir unser Beispiel so ab, dass die Ausgabe der E-Mail auf der gleichen Seite erfolgt und beim Drücken des Buttons nur das Ergebnisfeld aktualisiert und nicht die ganze Seite neu geladen wird.

Ferner bauen wir noch ein Ausgabefeld ein, in dem der gesamte Name (Vorname + Nachname) angezeigt wird. Dieses Feld wird beim Verlassen des Eingabefeldes für Nachname aktualisiert.

Durch das Tag `f:ajax`, das in der Komponente `inputText` eingebettet ist, wird beim `onchange`-Trigger des Feldes ein AJAX-Request abgesetzt (siehe Abbildung 4). Dies ist das Default-Verhalten. Möchte man einen anderen Trigger des jeweiligen Feldes benutzen, so kann man dies mit dem Attribut `event` angeben.

Das `execute`-Attribut veranlasst die Abarbeitung des JSF-Lebenszyklus für die angegebenen Komponenten (exclusive der Rendering-Phase). In unserem Beispiel ist dies das Feld mit dem Tag `f:ajax` (`@this`) und das Feld mit der `id="vor"`. Mit dem Attribut `render` gibt man an, welche Komponenten der Seite aktualisiert werden, in unserem Fall das Feld `outputText` mit der `id="voll"`.

Auch dem `commandButton` haben wir ein AJAX-Tag verpasst. Hier wird dafür gesorgt, dass nach Drücken des Buttons das Ergebnis im Feld `outputText` mit der `id="email"` neu umgesetzt und angezeigt wird (siehe Abbildung 4).

Man kann mit diesem Tag `f:ajax` nicht nur einzelne Komponenten mit AJAX-Funktionalität versehen, sondern auch ganze Regionen. Dafür reicht es aus, die gewünschten Komponenten in ein `f:ajax`-Tag einzubetten. Die Standard-Trigger der eingebetteten Komponenten werden dadurch mit AJAX-Funktionalität angereichert.

Validierung

Die Validierung ist ein zentrales Thema jeder Business-Anwendung. Zudem kommt sie auch noch in verschiedenen Schichten vor.

Zunächst wird meist in der Oberfläche eine erste Validierung durchgeführt. Anschließend wird aus dem User Interface-Controller eine Business-Funktionalität aufgerufen. In dieser muss natürlich auch nochmals validiert werden. Zu guter Letzt wird eine `Bean` persistiert, was erneut eine Validierung voraussetzt.

In der Vergangenheit kamen auf den verschiedenen Layern zum Teil unterschiedliche Technologien zum Einsatz, was zu Redundanzen oder sogar zu unterschiedlichen Ergebnissen in der Validierung führte. Um diesen Diskrepanzen vorzubeugen, wurde erstmals eine standardisierte API (Bean Validation API) erarbeitet, mit der genau diese Anforderung umgesetzt werden kann. Die Validierungslogik ist an einer zentralen Stelle, nämlich an der Klasse selbst zu finden.

Diese Bean Validation API kann nun auch zusammen mit JSF benutzt werden. Allerdings müssen die dazu notwendigen Bibliotheken ggf. separat in das Projekt mit eingebunden werden. JSF 2.0 schaut beim Validieren von Managed Beans nach, ob an einer Eigenschaft Bean Validation API-kompatible Annotationen vorhanden sind und wertet diese aus.

Wir erweitern nun unser Beispiel um diese Annotationen (siehe Abbildung 5). Beide Eingabeattribute (`vorname`, `nachname`) dürfen nicht Null sein und der Nachname muss mindestens 2 Zeichen lang sein. Die Meldung, die beim Attribut `message` steht, wird im Fehlerfall auf der Oberfläche ausgegeben.

```
...
<h:body>
  <composite:interface>
    <composite:attribute name="vName" required="true"/>
    <composite:attribute name="nName" required="true"/>
  </composite:interface>
  <composite:implementation>
    <h:panelGrid columns="2">
      Vorname:
      <h:inputText value="#{cc.attrs.vName}"/>
      Nachname:
      <h:inputText value="#{cc.attrs.nName}"/>
    </h:panelGrid>
  </composite:implementation>
</h:body>
...
```

Abb. 6: vollerName.xhtml.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:oka="http://java.sun.com/jsf/composite/meineKomponenten">
  ...
  <h:body>
    <h:form>
      <oka:vollerName vName="#{mitarbeiterBean.vorname}"
        nName="#{mitarbeiterBean.nachname}"/>
      <h:messages/>
      <h:commandButton action="#{emailControler.berechneEmail}"
        value "Berechne E-Mail" />
    </h:form>
  </h:body>
  ...
</html>
```

Abb. 7: eingabeKomp.xhtml.

```
<f:view contentType="text/html">
  <f:metadata>
    <f:viewParam name="nachname" value="#{mitarbeiterBean.nachname}"/>
  </f:metadata>
  <h:body>
    <h:form>
      <h:panelGrid columns="2">
        Vorname:
        <h:inputText value="#{mitarbeiterBean.vorname}"/>
        Nachname:
        <h:inputText value="#{mitarbeiterBean.nachname}"/>
      </h:panelGrid>
      <h:messages/>
      <h:commandButton action="#{emailControler.berechneEmail}"
        value "Berechne E-Mail" />
    </h:form>
  </h:body>
</f:view>
```

Abb. 8: View Parameter.

Glossar

AJAX	Asynchronous JavaScript and XML. Ajax ist ein Konzept, das den asynchronen Austausch von XML-Nachrichten zwischen Client und Server erlaubt, ohne dass eine Webseite komplett neu geladen werden muss.
Annotation	Anmerkungen im Java Source Code, die zur Laufzeit des Programms ausgewertet werden können. Annotations sind seit Java 5 Bestandteil von Java. Als geistigen Vater dieser Annotations kann man das Open Source Projekt XDoclet ansehen. XDoclet erlaubte es, auch schon in früheren Java-Versionen mit Annotations zu programmieren.
Java EE	Java Plattform, Enterprise Edition; ist die Spezifikation einer Software-Architektur für die transaktionsbasierte Ausführung, der in Java programmierten Anwendungen und insbesondere Web-Anwendungen.
JSP	Java Server Page. JSP ist die Java-Komponente einer Web-Anwendung, die durch den JEE-Standard definiert ist. Sie ähnelt vom Aufbau einer HTML-Seite mit integriertem Java-Code. Sie wird zuerst vom Web-Container in ein Servlet kompiliert. Anschließend nimmt das Servlet Client-Anfragen in Form eines HTTP-Requests entgegen und generiert dynamisch eine Antwort, die es über eine HTTP-Response an den Client zurückschickt.

`/resources/<Name der Komponentensbibliothek>` abgelegt sind. Wir speichern also unsere Datei `vollerName.xhtml` im Verzeichnis `/resources/meineKomponenten` ab. Die Komponente steht dann ohne weitere Konfiguration zur Verfügung.

Eine Komponente besteht immer aus einer Schnittstelle (Interface) und der Implementierung. In unserem Beispiel (siehe Abbildung 6) hat unsere Schnittstelle (dargestellt durch das Tag `<composite:interface>`) zwei Attribute: `vName` und `nName`. In der Implementierung (dargestellt durch das Tag `<composite:implementation>`) kann auf die übergebenen Attribute über die vordefinierte `cc-Variable` zugegriffen werden (in unserem Fall `cc.attrs.vName` und `cc.attrs.nName`).

Die Verwendung unserer neuen Komponente ist ebenfalls sehr einfach. Zunächst muss die neue Bibliothek in den Namespace aufgenommen und mit einem Präfix (in unserem Fall `oka`) verknüpft werden (siehe Abbildung 7). Danach kann die Komponente über das Tag `oka:vollerName` verwendet werden.

Composite Components

Ein großer Kritikpunkt an JSF 1.x war, dass das Erstellen eigener Komponenten keine triviale Sache war. In JSF 2.0 wird das Ganze sehr vereinfacht, dank der Integration von Facelets und der darin enthaltenen **Composite Components**.

Eine **Composite Component** kann ohne viel Aufwand angelegt werden und somit die Wiederverwendbarkeit von **Code snippets** in Form von Tags erreicht werden.

Entwickler erhalten durch die Verbindung von Facelets und Ressourcen die Möglichkeit, Komponenten aus beinahe beliebigen Seitenfragmenten aufzubauen. Eine **Composite Component** ist im Grunde nichts anderes als ein XHTML-Dokument, das in einer Ressourcenbibliothek abgelegt ist und die Komponenten deklariert.

Wie dies funktioniert, zeigen wir an unserem Beispiel, indem die Eingabe der Namen als eigene **Composite Component** erstellt wird und somit wiederverwendbar ist. Als erstes erstellen wir eine Komponente, die wir in der Datei `vollerName.xhtml` speichern.

Durch das neue Resource Loading in JSF 2.0 werden Komponenten (und Resource Bundles) automatisch erkannt, wenn sie im Verzeichnis

View Parameter

Zum Abschluss noch eine kleine, aber trotzdem sehr wichtige Neuerung. Ein häufiges Szenario in Anwendungen ist es, Url-Parameter einlesen zu müssen. Das ist meist dann der Fall, wenn eine Anwendung von einer anderen Anwendung aus aufgerufen wird und Übergabewerte als Url-Parameter übergeben werden, z. B. `localhost:8080/meinProgramm/ingabe.jsf?nachname=Kaluza`.

In JSF 1.2 war es prinzipiell zwar möglich, auf diese Parameter zuzugreifen, allerdings war dieser Zugriff ein wenig „hacky“. Man musste über den **ExternalContext** gehen und die gesamte Syntax sah wenig elegant aus.

Hier kommt in JSF 2.0 das Tag `f:viewParam` zum Tragen. Mit Hilfe dieses Tags wird der übergebene Parameter automatisch ausgelesen und wie in unserem Beispiel gezeigt direkt in der Managed Bean `mitarbeiterBean` gespeichert (siehe Abbildung 8). Auf diesen übergebenen Wert wird die Datenkonvertierung und Datenvalidierung angewendet.

Kommt es bei der Übergabe zu einem Fehler (ist der Wert z. B. keine 2 Zeichen lang, wie in unserem Beispiel gefordert), so wird eine entsprechende Fehlermeldung ausgegeben.

Fazit

Dieser Artikel konnte viele Dinge nur anreißen und einige Neuerungen konnten auch gar nicht erwähnt werden. Dennoch wurde deutlich, dass es sich bei JSF 2.0 um eine konsequente Weiterentwicklung des bisherigen Standards handelt und dieser nun entgültig erste Wahl ist,

Link

▶ [1] JSF: <http://java.sun.com/javaee/javaserverfaces/>

wenn es um einfache und effiziente Web-Entwicklung mit Java geht.

Oliver Kaluza (info@ordix.de).



Java Server Faces (JSF) im Expertencheck

1. Entwicklungsaufwand	■■■■■■■■□□
2. Browser-Unterstützung	■■■■■■■■■□
3. Funktionsumfang und verfügbare Erweiterungen	■■■■■■■■■■
4. Einarbeitung für den Entwickler	■■■■■■■■□□
5. Dokumentation	■■■■■■■■■□
6. Weiterentwicklung und Bugfixing	■■■■■■■■■■
7. Reifegrad des Frameworks	■■■■■■■■■□
8. Zukunftspotenzial	■■■■■■■■■■
9. Gesamteindruck	■■■■■■■■■□

positiv: ■■■■■■■■■■

negativ: □□□□□□□□



„JSF hat in den letzten Jahren bei der Web-Entwicklung mit Java einen kometenhaften Aufstieg hinter sich. Die Version JSF 2.0 wurde stark erweitert, vereinfacht und standardisiert, so dass man bei der Web-Entwicklung mit Java nicht mehr um die neue JSF-Version herumkommt.“

Oliver Kaluza, ORDIX AG, Paderborn

Seminarempfehlung: Web-Anwendungen mit Java Server Faces (JSF)

▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Java-JEE/nr/555/index.html>

JavaServer Faces (JSF) ist der in JEE spezifizierte Standard für die Entwicklung von Web-Anwendungen mit Java. In diesem Seminar werden Sie mit den grundlegenden Techniken vertraut gemacht, die für die Entwicklung einer Web-Anwendung mit JSF erforderlich sind.

Seminarinhalte

- Grundlagen
- JSF-Implementierungen
- Architektur von JSF-Anwendungen
- Managed Beans
- Phasen eines JSF-Requests
- Validierung von Eingaben
- Verarbeitung von Fehlermeldungen
- Konvertierung von Eingaben
- Value Binding
- Method Binding
- Facelets
- Navigationskonzept
- Internationalisierung
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

- 13.09. - 17.09.2010 in Wiesbaden
- 15.11. - 19.11.2010 in Wiesbaden

Seminar-ID: P-JEE-05

Dauer: 5 Tage

Preis pro Teilnehmer: 1.590,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.431,00 € (zzgl. MwSt.)

Wir führen unsere Seminare auch jederzeit an einem geeigneten Ort Ihrer Wahl durch und bringen, wenn nötig, auch das entsprechende Equipment mit. Informieren Sie sich am besten im Internet über unsere Kundenseminare und die mobilen Schulungen: <http://training.ordix.de>.

Wie der Blitz raus und rein

Dieser Artikel richtet sich an Datenbankadministratoren, Data Warehouse-Spezialisten und Entwickler.

Welcher Datenbankadministrator hatte noch nie die Aufgabe, einige Millionen Datensätze zu laden oder zu entladen? Abhilfe schafft hier das Schreiben einer Prozedur oder eines kleinen Programmes. Eine weitere Lösungsmöglichkeit bietet die Suche nach einem Werkzeug des Datenbanksystemherstellers oder eines kostenpflichtigen Werkzeuges von Drittanbietern. Informix bot bezüglich Laden und Entladen schon immer eine große Vielfalt an Lösungen: SQL-Erweiterungen im dbaccess wie Load/Unload, das Werkzeug dload, entladen oder laden im proprietären Page Format (onload/onunload) und nicht zuletzt den Turbo High Performance Loader (HPL oder onpload). Diejenigen, die sich mit der XPS-Variante von Informix beschäftigt haben, kennen auch die Funktion „External Table“, die mit der Informix-Version 11.5xC6 - angeblich sogar ohne Aufpreis - ausgeliefert wird.

Auslagern / Einlagern aber wie?

Wann immer man mit Reorganisationen, Migrationen, Auslagerung oder Archivierung von Daten beschäftigt ist, stehen zeitliche Aspekte im Vordergrund. Je schneller Lade- oder Entladevorgänge ablaufen desto besser. Informix bietet zwar mit dem HPL hier durchaus eine gute Lösung, die aber viele Datenbankadministratoren scheuen, weil sie komplex zu konfigurieren ist und bei auftretenden Fehlern die Analyse oder Reaktion auf diese Fehler nicht immer trivial sind.

Deshalb werden trotz teilweise extremer Laufzeiten bei Informix weiterhin sehr häufig die komfortablen SQL-Erweiterungen **LOAD** und **UNLOAD** im Werkzeug **dbaccess** genutzt. Kenner von Informix XPS wissen deshalb die Funktion der so genannten **EXTERNAL TABLES** zu schätzen, die über SQL Syntax angelegt werden und dann einfach über **SELECT** und **INSERT** in beide Richtungen angesprochen werden können.

Die Arbeitsweise bei einem **LOAD** von einer **EXTERNAL TABLE** ist in Abbildung 1 dargestellt.

Einfach anlegen und loslegen

Mithilfe einfacher SQL-Erweiterungen beim **CREATE TABLE** wird eine **EXTERNAL TABLE** angelegt und schon kann man sie zum Entladen nutzen (oder bei vorhandenen dementsprechenden Datenfiles auch zum Laden). Die SQL-Syntax zum Anlegen einer Tabelle finden Sie in den Abbildungen 2 a und b, die Syntax zum Laden oder Entladen ist in Abbildung 3 dargestellt.

Zusätzliche Optionen lassen die Verwandtschaft mit dem HPL erkennen. Die Syntax zeigt die einfache Verwendung auch in Anwendungen oder Skripten.

Neue Systemtabellen

Für die Implementierung dienen drei neue, obligatorische Systemtabellen:

- **sysexternal**
- **sysextdfiles**
- **sysextcols**

Ob diese Systemtabellen bei einem Upgrade auf die Version xC6 einfach angelegt werden,

muss von unserer Seite noch geprüft werden. Da aber Versionswechsel bei Informix bislang immer recht problemlos vonstatten gingen, gehen wir davon aus, dass auch bei diesem Wechsel keine Probleme auftreten werden. Wir werden darüber in einer der nächsten ORDIX News berichten.

Und die Performance?

Nach Informix Angaben soll das Verfahren mit External Tables sowohl beim Unload wie auch beim Load deutlich schneller als mit dem HPL sein. Diese Aussage werden wir in den nächsten Wochen intensiv verifizieren und die Ergebnisse in einer der nächsten ORDIX News vorstellen. Bei dem Vergleich der beiden Verfahren greifen wir auf gute Erfahrungen und viele erfolgreiche Projekte mit dem HPL zurück.

Sollten die angekündigten Verbesserungen im Bereich Methoden und Werkzeuge zutreffen, werden wir uns nicht gegen diesen „Trend“ verschließen.

Was nun?

Schon auf den ersten Blick macht die Technologie einen guten Eindruck. Sie scheint nicht nur einfacher als der HPL einsetzbar zu sein, sondern auch gut zu funktionieren, selbst wenn wir noch nicht alle Funktionalitäten ausführlich testen konnten.

Sollte sich auch die Performance-Aussage bewahrheiten, wird es schwer, noch für den HPL zu argumentieren, auch wenn wir von ORDIX mit unserem Aufsatzprodukt die Nutzung des HPL deutlich vereinfachen. Wir behalten die Funktion **EXTERNAL TABLE** im Auge und werden in den nächsten ORDIX News noch die eine oder andere Frage beantworten.



Ulf Papenfuß
(info@ordix.de).



Michael Skowasch
(info@ordix.de).

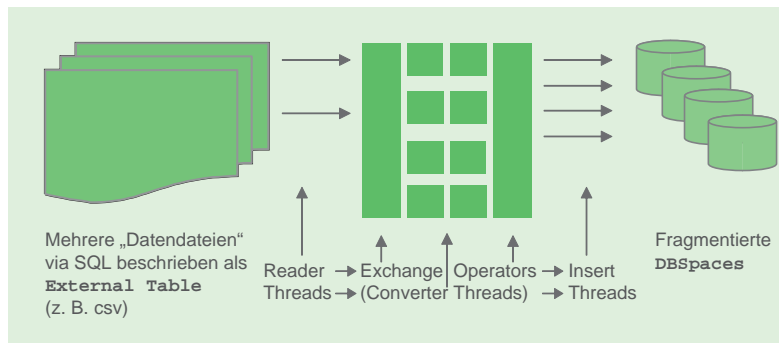


Abb. 1: Generelle Arbeitsweise External Tables beim Lesen von External und Einfügen in normale Tabellen.

```
CREATE EXTERNAL TABLE <tablename>
( <columnlist> )
USING (DATAFILES( x1 [,x2,... ] ), <table-options> )
```

Abb. 2a: Einfaches SQL zum Anlegen der External Table mit expliziten Spalten ...

```
CREATE EXTERNAL TABLE <tablename> SAMEAS <templatetable>
USING (DATAFILES( x1 [,x2,... ] ) , <table-options> )
```

Abb. 2b: ... oder noch einfacher auf Basis einer existierenden Tabelle.

```
SELECT <column-list> FROM <external-table>
-- Selektieren von Daten
INSERT INTO <base-table> SELECT <column-list> FROM <external-table>
-- Laden von Daten
INSERT INTO <external-table> SELECT <column-list> FROM <base-table>
-- Entladen von Daten
```

Abb. 3: So einfach geht nun Selektieren, Laden oder Entladen.

Links

- ▶ [1] IBM IDS 11.5 Infocenter:
<http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp>
- ▶ [2] International Informix User Group (IIUG)
<http://www.iiug.org>
- ▶ [3] Informix User Group
<http://www.iug.de/>
- ▶ [4] Informix Zone
<http://www.informix-zone.com>
- ▶ [5] ORDIX Seminare zum Thema Informix
http://www.ordix.de/trainingsshop/index_informix.htm

Kick IT!



**Verschaffen Sie Ihrem TEAM einen Vorteil -
Training ist der Schlüssel zum Erfolg!**

Kommen Sie zu uns ins Trainingslager und bilden Sie sich weiter!

Ohne das entsprechende fachliche und organisatorische Know-how scheitern viele Projekte.

Mit unserem Trainingsangebot sorgen wir dafür, dass Sie nichts dem Zufall überlassen müssen, sondern Ihre Kenntnisse gezielt in den Bereichen aufbauen und erweitern, die für Ihre Projekte notwendig sind.

Wir trainieren Sie in den Bereichen:

- Datenbanken
- Programmierung
- Java/JEE
- Web- und Applikations-Server
- Betriebssysteme
- Systemmanagement
- Projektmanagement
- IT-Management

Informationen, Download und Buchung unter <http://training.ordix.de>

Fordern Sie die Seminarbroschüre an unter training@ordix.de