

ORDIX news

Das IT-Magazin der ORDIX AG



Objektorientierung in Oracle

18 | Objektorientiert entwickeln mit PL/SQL

- 8 | DAO - Data Access Object
Geschichtet für die Wiederverwendbarkeit
- 32 | XEN - Der Stern leuchtet noch
Neue Funktionen der Virtualisierung
- 36 | „Secure by Default“ in Oracle 11g
Die sichere Datenbank out-of-the-box?
- 41 | Perlen (nicht nur) für DB2-DBAs
Best Practice, Erste Hürden



Mehr als 100
Veranstaltungen
im Jahr

DOAG-Mitglieder sind besser informiert



Wissensvermittlung,
Erfahrungsaustausch
und Vernetzung

Die nächsten Termine:

Special Interest Groups

- 23. März SIG BPM
- 14. April SIG Fusion Middleware – Oracle WebLogic Server 11g
- 29. April SIG Development – Rapid Development mit Java
- 03. Mai SIG SOA
- 08. Juni SIG Database



Fachkonferenzen

- 24. – 25. März **DOAG 2010 Applications**
- 04. – 05. Mai OAUG Connection Point Budapest
- 11. Mai DOAG ITIL und Betrieb 2010
- 16. – 18. November **DOAG 2010 Konferenz + Ausstellung**



Berliner Expertenseminare

- 26. – 27. April Oracle Security
- 17. – 18. Mai Fortgeschrittene Performancediagnose und Optimierung mit extended SQL Trace Profiling
- 22. – 23. Juni Optimizing Oracle Performance
- 22. – 23. September Oracle 11g Performance Forecast & Oracle 11g Performance Firefighting
- 05. – 06. Oktober Advanced APEX I & Advanced APEX II



Crisis? What Crisis?¹

Paderborn, Februar 2010

So oder so ähnlich müsste man denken, wenn man das Gezanke und Herumeiern unserer neuen Regierung betrachtet. Können wir entweder froh sein, dass die Krise schon 2008 kam oder ist es völlig egal, welche Regierung man hat, weil ihr wirtschaftspolitischer Einfluss real eher zu vernachlässigen ist?

Auf jeden Fall ist das schon sehr peinlich, was da in den letzten vier Monaten passiert ist. Insbesondere, ich muss Herrn Pinkwart² Recht geben, die abgesenkte Mehrwertsteuer für Hotelübernachtungen und damit die immensen Änderungen bei Reisekostenabrechnungen machen nur Arbeit und bringen definitiv nicht mehr Netto vom Brutto. Wie war das mit versprochen und gehalten?

Aber irgendwie scheint bei uns ja noch genügend Geld vorhanden zu sein. Die Kommunen jammern zwar gewaltig – bei den Schlaglöchern in so mancher Straße nach dem harten Winter mehr als verständlich – aber wir geben einem Land, das einfach gar nicht wirtschaften kann, erstmal wieder eine Gyros- nein natürlich Euro-Spritze. Da warte ich nur darauf, dass andere Länder bald nachziehen werden. Zumindest spricht man schon von den Euro PIIGS³. Und das alles trotz der Kreditklemme, obwohl Kreditklemme stimmt eigentlich auch nicht, wir Mittelständler geben zumindest inzwischen diversen Großkonzernen monatelang Barkredite und das alles ohne Absicherung.

Larry Ellison wiederum hat auch viel Geld ausgegeben und hat es nun geschafft. SUN wurde einverleibt. Wie es da wirklich weitergehen wird, weiß aber noch keiner so recht⁴. Für mich steht damit leider fest, dass der nicht gerade vor Innovationen strotzenden INTEL Phalanx nur noch IBM mit seinen Power Prozessoren entgegensteht. Nur noch zwei Anbieter auf dem Markt verheißt jedoch in den meisten Fällen nichts Gutes. Oder wie finden Sie die Energiepreisentwicklung?

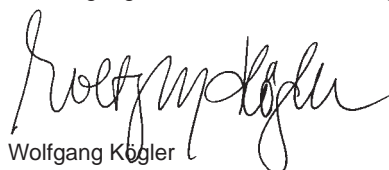
Aber gut, wir haben in dieser Ausgabe auch nur zwei Artikel, die sich mit dem objektorientierten Zugriff auf die Datenbank beschäftigen (Java Best Practice: Data Access Object und Objektorientierung in Oracle). Daneben aber auch noch einige andere interessante Themen aus dem Umfeld von Oracle, die irgendwie immer mit dem Stichwort „Sicher“ zu tun haben (Sicherung: Oracle und NetApp, Sicherheit: Oracle 11g).

Abgerundet wird diese Ausgabe mit Artikeln zum Thema Virtualisierung, Monitoring, Programmierung (JGoodies Forms und Perl mit DB2) und Oracle Migration mit Logical Data Guard. Nicht ganz unwichtig ist die Fortsetzung der Artikelreihe IT Management, dort geht es auch um Orientierung.

Zuletzt finden Sie noch einen Artikel über die Orientierung der ORDIX in ihren ersten 10 Jahren.

Ich wünsche Ihnen, dass sich nach dem langen Winter bei Ihnen langsam Frühlingsgefühle einstellen können, so dass Sie sich mit uns auf ein spannendes und hoffentlich erfolgreiches Jahr eins nach der Krise einstellen können. So wie es aussieht, können wir uns dabei nicht auf die Berliner Politiker verlassen, denn wie schon 2009 ist auch 2010 leider wieder Wahljahr. Selbst eine 1,54-Promille-Fahrt lenkte nur kurz die Schlagzeilen auf ein anderes Thema.

Viel Vergnügen beim Lesen der Schlagzeilen und der ORDIX News 1/2010 wünscht Ihnen



Wolfgang Kögler



¹ Crisis? What Crisis? Freunden der Rockmusik sicher als Album der Band Supertramp ein Begriff. Tatsächlich stammt der Spruch aber aus dem Film „Der Schakal“ von 1973, dem Jahr der Ölkrise (Quelle: Wikipedia)

² Prof. Dr. Andreas Pinkwart ist FDP Minister in NRW und Stellvertreter des Ministerpräsidenten in NRW

³ PIIGS: Portugal Irland Italien Griechenland Spanien.

⁴ Bei meinem letzten Editorial stand das Ergebnis dieses Deals noch auf der Kippe.



8 | Data Access Object (DAO) - Profis für den Zugriff



18 | Objektorientierung in Oracle

Java/JEE

- 8..... **Java Best Practice (Teil VI): Data Access Object (DAO) Profis für den Zugriff**
Das DAO Entwurfsmuster ermöglicht als separate Software-Schicht den Zugriff auf die Datenbank.
- 44..... **JGoodies Forms - Swing in Form bringen**
Einblick in die Erstellung einer Benutzeroberfläche mit dem kostenfreien Paket JGoodies Forms.

Betriebssysteme

- 32..... **Server-Virtualisierung mit XEN (Teil I): XEN - Der Stern leuchtet noch**
Vorstellung neuer Möglichkeiten von XEN, die in den letzten Jahren entwickelt wurden.

Open Source

- 6..... **GroundWork Monitor - Big Brothers neue Staffel**
GroundWork ergänzt Produkte wie das JBoss-Portal, Nagios, Monarch sowie MySQL mit eigenen Skripten zu einer zukunftssträchtigen Monitoring-Lösung.

IT-Strategie

- 12..... **IT-Management (Teil III): Orientierung für die IT - Entwicklung von IT-Strategien**
Der dritte Teil der Reihe erläutert, mit welchen Mitteln und Richtlinien eine IT-Strategie umgesetzt werden kann.

Datenbanken

- 18..... **Objektorientierung in Oracle**
Ein erster Einstieg, wie mit der Programmiersprache PL/SQL objektorientiert entwickelt werden kann.
- 28..... **Oracle auf NetApp (Teil III): NetApp SMO - der SnapManager für Oracle**
Dieser Artikel erläutert, wie Datenbanksicherungen mit dem SnapManager geplant, erstellt, verwaltet und zurückspielt werden können.
- 36..... **Oracle 11g Security (Teil I): „Secure by Default“ in Oracle 11g - Die sichere Datenbank out-of-the-box?**
Vorstellung der neuen Maßnahmen von Oracle, die die Datenbank sicherer gestalten sollen.
- 41..... **Perl und DB2 im Verbund (Teil I): Perlen (nicht nur) für DB2-DBAs**
Diese neue Artikelreihe erläutert, wie Perl-Programmierung im DB2-Umfeld funktioniert.
- 48..... **Rolling Upgrade mit Logical Data Guard**
Beschreibung eines Oracle Migrationsverfahrens mit minimaler Ausfallzeit auf Basis des Logical Data Guard.

ORDIX 2.0

1990 - 2010 | 20 Jahre bessere IT-Beratung



28 | NetApp SMO - der SnapManager für Oracle



32 | XEN - Der Stern leuchtet noch

Aktuell

- 16..... **Rückblick DOAG Konferenz und Schulungstag 2009**
Security Profis treffen Strategen
- 23..... **Neue Seminare 2010: IT-Management**
- 24..... **20 Jahre ORDIX (Teil I): 1990 - 1999**
Ein Rückblick: ORDIX entwickelt sich, die (IT-)Welt auch.
- 40..... **Larry Ratlos: Java Varargs?**

Standards

- 03..... **Editorial**
- 04..... **Inhalt**
- 05..... **Impressum**
- 26..... **Seminarübersicht: März bis August 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: 9.800

Druck: Media-Print Informationstechnologie GmbH, Paderborn

Bildnachweis:

© aboutpixel.de / Apfelschichten © Thorwald Hoffmann
© aboutpixel.de / Donauinsel © Christian A.
© aboutpixel.de / nostalgisches Bildaufnahmegerät4 © Grabschreit
© flickr.de © Achim Schrepfer
© istockphoto.de © lorenzo puricelli
© shutterstock.com © donvictorio

Autoren dieser Ausgabe:

Christian Fertsch, Markus Fiegler, Matthias Heini, Stefanie Heither-Bürger, Kathleen Hock, Matthias Jung, Dr. Stefan Koch, Wolfgang Kögler, Norbert Munkel, Vanessa Prior, Thomas Rohde, Tobias Wink

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.

Die Zeitschrift ORDIX News wird von der ORDIX AG an ausgewählte Kunden verteilt und kann für 2,20 Euro bestellt werden. Sie finden sowohl die neueste Ausgabe als auch ältere Ausgaben 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.

GroundWork Monitor - Big Brothers neue Staffel

Dieser Artikel richtet sich an Administratoren und Entscheider, die sich für Monitoring-Lösungen interessieren.

Neben den bekannten Stars wie HP OpenView, BMC Patrol/Performance Manager und IBM Tivoli betritt jetzt ein neues, auf OpenSource basierendes, Nachwuchstalente die Bühne. GroundWork ist keine eigenständige Software, sondern kombiniert clever verschiedene OpenSource-Programme zu einem großen Ganzen. Es besticht durch eine intuitive Oberfläche und der damit verbundenen, einfachen Konfiguration.

Was ist GroundWork?

GroundWork ist ein Produkt von GroundWork Open Source aus den USA. Es beinhaltet Produkte wie das JBoss-Portal, Nagios, Monarch, MySQL und ergänzt sie mit eigenen Skripten zu einer Monitoring-Lösung mit Potential.

Das Herz bildet Nagios, welches voll in die Oberfläche des JBoss-Portals integriert ist. Zusätzlich zu der von Nagios gewohnten Oberfläche wird dadurch eine weitere Möglichkeit geboten, sich die aktuellen Status anzeigen zu lassen (siehe Abbildung 1).

Die Konfiguration von GroundWork erfolgt, im Gegensatz zu einem reinen Nagios-System, nicht direkt in Textdateien, sondern über die Web-Oberfläche.

In dieser kann durch die Verwendung von Monarch vom Check-Kommando (siehe Abbildung 2) bis zur Service-Gruppe alles konfiguriert werden. Um Nagios die Änderungen an den Einstellungen mitzuteilen, müssen sie „committed“ werden.

Dabei werden die Konfigurationsdateien aus der MySQL-Datenbank, in der alle Einstellungen der Web-Oberfläche gespeichert sind, generiert und einem Preflight-Check unterzogen.

Für den Umstieg von einem bestehenden Nagios-System gibt es die Möglichkeit, eine bestehende Nagios-Konfiguration einzulesen.

Versionen

Es gibt aktuell zwei verschiedene GroundWork-Versionen:

- Community Edition
- Enterprise Edition

Der Community Edition mangelt es, wie zu erwarten, an einigen interessanten Funktionen, wie beispielsweise der Event-Konsole. Ergänzend wird auf Basis der Community Edition das GroundWork Developer's Kit bereitgestellt.

Community Edition

Die Community Edition wird aktuell in den folgenden drei Varianten angeboten:

- VMware Image
- 32-Bit Installer
- 64-Bit Installer

Für einen ersten Eindruck bietet sich die VMware Version an. Sie enthält ein minimal installiertes CentOS Release 5.3 i386 und die aktuelle Version der GroundWork Community Edition, so dass man direkt mit der Konfiguration anfangen kann, ohne sich Gedanken um die Installation machen zu müssen.

Enterprise Edition

Die Enterprise Edition erweitert die Community Edition um einige interessante Funktionen.

Einen gesamten Vergleich zwischen Community- und Enterprise Edition finden Sie unter [2].

Die interessantesten Zusatzfunktionen:

- Die Event-Konsole bietet durch die Verwendung von Syslog-ng einen zentralen Überblick über jegliche aufkommende Log-Meldungen. Die Verwendung von Filtern sorgt für einen schnellen und guten Überblick über die aktuellen Vorkommnisse auf den überwachten Clients.
- Das Erstellen von Reports auf Basis von BIRT zählt als weiteres Feature. Dabei wird zwischen Fehleranalyse, Top 5 Reports, Performance-Reports und SLA-Reports unterschieden.

GroundWork Developer's Kit

Das GroundWork Developer's Kit stellt eine gute Möglichkeit dar, die Community Edition um fehlende Funktionen zu erweitern. Es besteht aus:

- GroundWork Monitor Community Edition
- Eclipse v3.4.2
- Maven v1.1
- Ant v1.7.1
- Notwendiger Quellcode, um API, Framework und Portlet Applikationen neu zu kompilieren

Um einen leichteren Einstieg in die Erweiterung von GroundWork Monitor zu bekommen, wurde ein Wiki mit Informationen zu dem Thema eingerichtet [3].

Fazit

GroundWork ist auf jeden Fall einen Blick wert. Es bietet schon in der Community Edition viele komfortable Möglichkeiten für das Monitoring von verschiedenen Geräten. Durch die Ergänzung des Developer's Kit steht selbstständigen Anpassungen und Erweiterungen nichts im Wege. Die Enterprise Edition kann es darüber hinaus nicht nur im Preis-/Leistungsvergleich mit etablierten Monitoring-Lösungen namhafter Anbieter aufnehmen.

Gerne unterstützen wir Sie bei der Implementierung ihrer Monitoring-Projekte, egal ob mit Open Source oder kommerzieller Software.

Tobias Wink (info@ordix.de).



Abb. 1: GroundWork Status View.

Abb. 2: Kommando editieren.

Glossar

Nagios	Freie Netzwerk-Monitoring Software.
Monarch	Web basierte Engine für die Nagios-Konfiguration
BIRT	Business Intelligence and Reporting Tools. BIRT ist ein BI-Tool der Eclipse Foundation, das auf Java-Technologien basiert.
Maven	Ein auf Java basierendes Build-Management-Tool der Apache Software Foundation.
JBoss Portal	Eine Open-Source-Implementierung eines Portals für den JBoss Application Server
Pre Flight Check	Eine vorab Syntaxüberprüfung

Link

- ▶ [1] GroundWork Webseite: <http://www.groundworkopensource.com/>
- ▶ [2] Vergleich zwischen Community- und Enterprise Edition: <http://www.groundworkopensource.com/products/familycomparison.html>
- ▶ [3] GroundWork Wiki: <http://www.groundworkopensource.com/wiki>
- ▶ [4] ORDIX Seminare zum Thema Nagios und MySQL <http://training.ordix.de>



Data Access Object - Profis für den Zugriff

Dieser Artikel ist für Java-Entwickler und Software-Architekten von Interesse.

Das Entwurfsmuster Data Access Objekt (DAO) [1] beschreibt eine separate Software-Schicht für den Zugriff auf die Datenbank. Wesentliche Vorzüge dieser Lösung sind die Wiederverwendbarkeit des Codes sowie die Abstraktion des Datenzugriffs. Dies allein sind schon gute Gründe, das Entwurfsmuster unter die Lupe zu nehmen.

Java
Best
Practice



Data Access Object

Der Name dieses Entwurfsmusters ist Programm: Es schlägt vor, Klassen zu erstellen, deren einzige Aufgabe es ist, Daten dauerhaft zu speichern oder dauerhaft gespeicherte Daten zur Verfügung zu stellen.

Eine DAO-Klasse enthält Methoden wie

- `Artikel getById(Long id)`, um einen Artikel anhand seiner ID zu laden
- `void saveOrUpdate(Artikel artikel)`, um einen Artikel abzuspeichern
- `void delete(Artikel artikel)`, um einen Artikel zu löschen
- `Collection<Artikel> findArtikelByXyz(..)`, um Artikel anhand bestimmter Kriterien zu laden

Das Objekt der Begierde ist in unserem Beispiel ein Artikel. Seine einzige Aufgabe ist es, Daten strukturiert aufzunehmen, die dauerhaft gespeichert werden sollen.

In der DAO-Schnittstelle werden so genannte Business-Objekte verwendet wie beispielsweise Kunde oder Bestellung. Die Namen der DAO-Klassen werden anhand der Business-Objekte hergeleitet. Kunden-Objekte werden in der `KundenDao`, Bestellungs-Objekte in einer `BestellungDao` abgespeichert.

Klasse DAO

Die Benutzung des DAO-Patterns wird anhand des Kommunikationsdiagramms der Abbildung 1 diskutiert. Als Beispiel wird ein fiktiver Bestellvorgang in einer JEE-Architektur skizziert.

Der Client verfügt über Kunden- und Artikelnummer und kennt die zu bestellende Menge. Mit diesen Informationen wird der Bestellvorgang auf den Weg geschickt. Die **BestellBean** nimmt die Nachricht entgegen und delegiert sie an einen Bestellservice. Dieser wird passend zur Kunden- und Artikelnummer die entsprechenden Kunden und Artikel aus der Datenbank laden.

Dazu verwendet der Service Instanzen von **KundenDao** und **ArtikelDao**. Darauf wird der Service ein Bestell-Objekt erzeugen und dieses mit Hilfe einer **BestellungDao** in der Datenbank ablegen.

Software klar geschichtet

Anhand des Kommunikationsdiagramms (Abbildung 1) lassen sich wesentliche Vorteile der Verwendung des Entwurfsmusters darstellen. Ein Service kann mehrere DAO-Objekte in Anspruch nehmen. Durch die Separation von Service und DAO ist die Wiederverwendbarkeit des Codes für den Zugriff auf die Daten gewährleistet.

Die Dienstleistung der DAO-Klassen wird offensichtlich angeboten: Sind Informationen zu einem Artikel oder einem Kunden erforderlich, welche Klasse muss eingesetzt werden? Die **ArtikelDao** und **KundenDao**, natürlich.

Offensichtlich ist auch, dass die DAO-Klassen eine Schicht bilden. Oberhalb dieser Schicht, zu den Service-Klassen hin, werden Informationen immer als Objekt (wie Kunde oder Artikel) behandelt. Aspekte der Datenbank bleiben verborgen.

In Abbildung 2 wird verdeutlicht, wie sich das DAO-Pattern in die Business Layer eingliedert. In der Business Layer wird die Geschäftslogik umgesetzt. Es wird von einer JEE-Architektur ausgegangen. Die Business Layer ist als Anwendung auf dem Application-Server deployed.

Durch den Einsatz des DAO-Patterns ist es möglich, die Business Layer weiter zu gliedern. Die DAO-Klassen sind dabei das Herzstück der Persistence Layer. In der Regel verwenden diese ein Persistence-Framework wie beispielsweise Hibernate oder eine Implementierung der Java Persistence API. Doch die Implementierung ist der Service Layer nicht bekannt – sie kennt nur die Schnittstellen der DAO-Klassen.

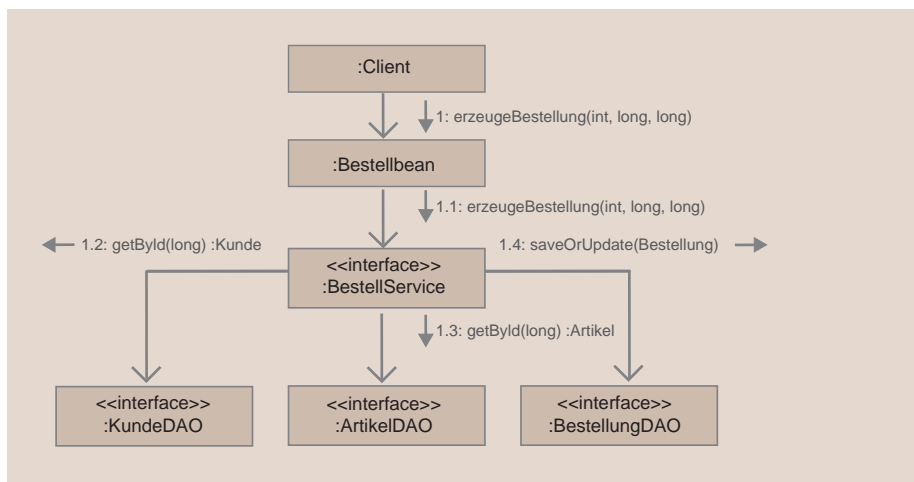


Abb. 1: Kommunikationsdiagramm für einen Bestellvorgang.

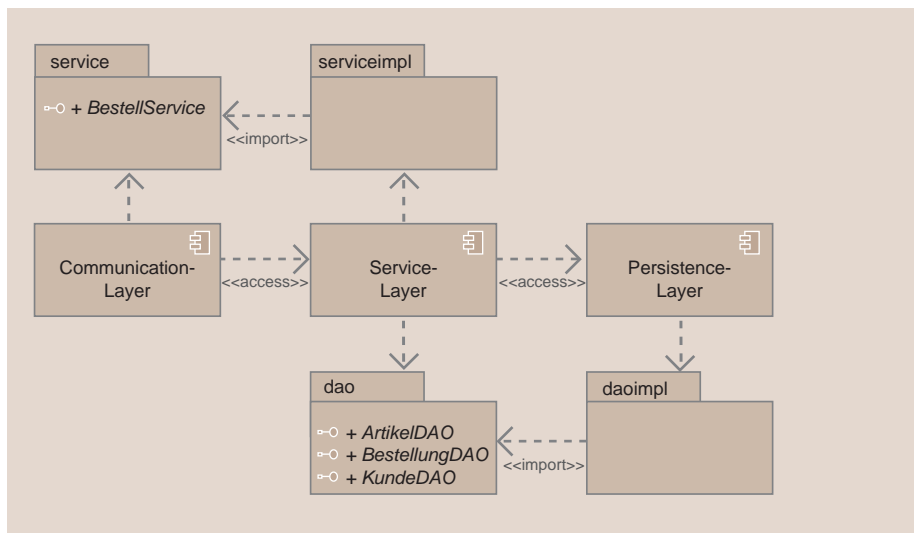


Abb. 2: Schichten-Architektur der Business Layer.

Die Service Layer implementiert die eigentliche Geschäftslogik. Sie ist durch die Persistence Layer aller Aufgaben der Datenpersistenz entledigt und kann sich dadurch auf ihre Kernaufgabe konzentrieren.

Zum Client hin sorgt eine Communication Layer (z. B. in Form einer Session Facade) dafür, dass die Kommunikation zwischen Client und Server problemlos verläuft. Schnittstellen können dadurch stabil gehalten werden, Services können in gewissem Umfang entkoppelt von der Client-Schnittstelle (weiter-)entwickelt werden. Exceptions werden mundgerecht an den Client weitergereicht.

Die Abstraktion der Persistenz führt dazu, dass die DAO-Schicht einfach austauschbar ist. Zu diesem Zweck sind DAO-Interfaces

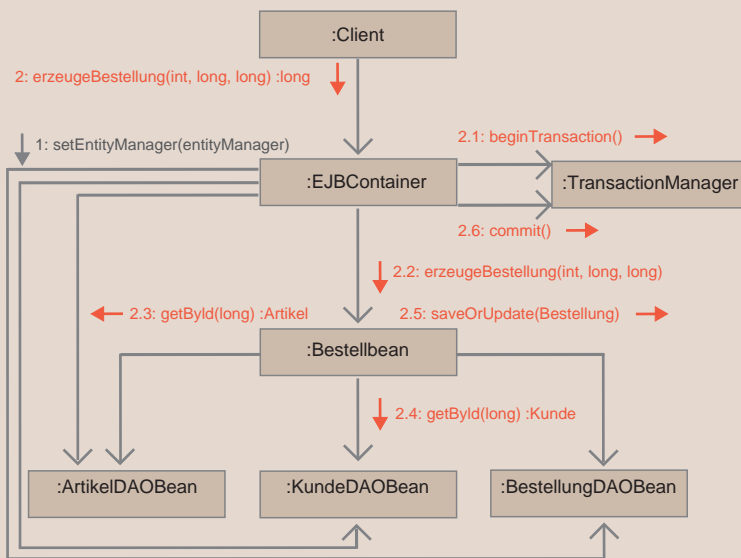


Abb. 3: Implementierung in einer JEE-Architektur.

```
package ejb;

import java.io.Serializable;
import javax.ejb.Local;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import common.Artikel;
import service.ArtikelDao;

@Stateless(name = "ArtikelDao")
@Local(service.ArtikelDao.class)
public class ArtikelDaoBean implements ArtikelDao, Serializable {
    @PersistenceContext(unitName = "containerPersistence")
    private EntityManager entityManager;

    public void delete(Artikel artikel) {
        entityManager.remove(artikel);
    }

    public Artikel getById(Long id) {
        return entityManager.find(Artikel.class, id);
    }

    public void save(Artikel artikel) {
        entityManager.persist(artikel);
    }

    public void update(Artikel artikel) {
        entityManager.merge(artikel);
    }
}
```

Abb. 4: Quellcode einer DAO-Klasse für die Persistenz von Artikel-Objekten.

zu definieren. Die Austauschbarkeit beflügelt den Entwicklungs- und Testprozess, indem einfache Varianten der DAO-Klassen (so genannte Mock-Klassen) eingesetzt werden.

Performance-Probleme stehen nicht selten in Zusammenhang mit Datenbankzugriffen. Die Kapselung dieser Zugriffe ist eine gute Voraussetzung dafür, dass sich Spezialisten genau auf diese Aspekte konzentrieren können.

Wo bleiben die Transaktionen?

Für die Implementierung sind noch wesentliche Aspekte nicht angesprochen worden:

- Woher bekommen DAOs den Zugriff zur Datenbank?
- Wie werden Transaktionen gehandhabt?

Die Beantwortung dieser Fragen richtet sich sehr nach der Umgebung, in der die Implementierung erfolgt. Unabhängig davon sollte eine Lösung aber die folgenden Aspekte berücksichtigen:

- Die Verbindung zur Datenbank ist eine kostbare Ressource. Es muss sichergestellt sein, dass bei einer Verarbeitung (Thread) immer nur eine Verbindung gleichzeitig verwendet wird. Die Ressource muss nach Gebrauch unbedingt wieder freigegeben werden. Bei Missachtung dieser Kriterien droht die Höchststrafe: Sie werden feststellen, dass die Anwendung unter Last buchstäblich zusammenbricht und minutenlang ihren Dienst verweigert. Alle Datenbankverbindungen sind in Verwendung und die Prozesse warten auf die zweite Datenbankverbindung, um weiter zu arbeiten.
- Transaktionen müssen außerhalb der DAO-Schicht geregelt werden. Nur so kann eine Wiederverwendung der DAOs ermöglicht werden. Mit Blick auf die Abbildung 1 fällt diese Aufgabe der Bean oder dem Service zu – wie ist das mit einem gekapselten Datenbankzugriff durch die DAO-Schicht zu verbinden?

Da die Implementierung von DAO zu den Standardaufgaben der Anwendungsentwicklung gehört, gibt es Standardlösungen. Vielfach erprobt ist der Einsatz des Java-Enterprise-Standards oder aber des Spring-Frameworks [2].

Im Folgenden wird die Implementierung von DAOs in einer Java-Enterprise-Architektur weiter verfolgt.

DAO-Implementierung mit JEE

Die Geschäftslogik ist in einer JEE-Architektur in einem EJB-Container auszuführen. Dieser Container stellt unter anderem genau die erforderliche Infrastruktur für die Implementierung des DAO-Entwurfsmusters zur Verfügung: Er verwaltet sowohl die Verbindung zur Datenbank als auch Transaktionen. In Abbildung 3 sind die Aufgaben des EJB-Containers skizziert:

Er erzeugt die DAO-Beans und injiziert diesen einen Entity Manager. Mit diesem kann direkt ein objekt-relationaler Zugriff auf die Datenbank erfolgen. Auch die `BestellBean` wird vom EJB-Container erzeugt und mit den erforderlichen DAOs ausgestattet.

Die Transaktionssteuerung kann und sollte der EJB-Container ebenfalls übernehmen (Container Managed Transaction). Ruft ein Client eine Service-Methode auf, so kümmert sich der EJB-Container um die Transaktion.

In dieser Infrastruktur kann sich jede Schicht auf ihre Kernaufgabe konzentrieren. Die Implementierung des DAO-Patterns wird stark vereinfacht. In Abbildung 4 ist sie exemplarisch für die Klasse `ArtikelDaoBean` dargestellt.

Der Entity Manager stellt die erforderlichen Methoden unmittelbar zur Verfügung. Doch das wird sich im Laufe der Zeit ändern: In der Regel kommen umfangreichere Methoden dazu, wenn die Business Objekte Kollektionen anderer Objekte enthalten oder aber spezielle Datenbanksuchen erforderlich sind.

Schwieriger zu deuten sind die Annotations zu Beginn des Quellcodes von Abbildung 4. Damit der EJB-Container für DAOs genutzt werden kann, müssen diese als EJBs implementiert werden. Die Annotation `@Stateless` konfiguriert diese Klasse als `Stateless-Session-Bean`.

DAOs werden durch `@Local` als lokale EJB konfiguriert: Der Client darf keinen direkten Zugriff auf diese Beans haben. So wird verhindert, dass ein Client seine eigene Applikationslogik implementiert.

Schlussfolgerung

Das Entwurfsmuster Data Access Object gehört für die Implementierung von Geschäfts-

Glossar

Data Access Object (DAO)	Entwurfsmuster, das eine separate Software-Schicht für den Zugriff auf die Datenbank vorsieht.
Design Pattern	Entwurfsmuster. Beschreibt Architekturen von Software-Komponenten, die sich bewährt haben.
EJB (Enterprise Java Bean)	Teil des JEE-Standards. Objekt, das von einem Application Server erstellt und dessen Methoden von einem Client aufgerufen werden.
JEE (Java Enterprise Environment)	Standardisiertes Framework für die Entwicklung von Server-Anwendungen.
JPA (Java Persistence API)	Teil des JEE-Standards. Über diese Schnittstelle erfolgt das dauerhafte Speichern oder Laden von Daten. Wesentliche Eigenschaft ist die Umwandlung von Objekten in Zeilen von Datenbanktabellen.
Mock-Objekt	Diese Objekte werden für die Entwicklung und Modultests eingesetzt. Sie implementieren Service Interfaces in möglichst einfacher Weise. Sie stehen dadurch zeitnah zur Verfügung. Ihr Verhalten ist gut vorhersehbar.
Factory	Fabrik - Entwurfsmuster für Klassen, die Objekte erstellen und zur Verwendung instrumentieren.

Links

- ▶ [1] Entwurfsmuster Data Access Object:
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
- ▶ [2] Spring-Framework:
<http://springframework.org>
- ▶ [3] ORDIX News Artikelreihe Java Best Practice
http://www.ordix.de/ORDIXNews/artikelreihen.html#java_best
- ▶ [4] Java-JEE Seminare der ORDIX AG
<http://training-ordix-de-entwicklung.ordix.de/siteengine/action/load/kategorie/Java-JEE/index.html>

...
Teil V: Das kleine 1x1 der Software-Entwicklung
Teil VI: Data Access Object
Teil VII: Build Management und IDE optimal kombinieren
...



logik mit Datenbankanschluss zum Stand der Technik. Die Herausforderung ist weniger die Entwicklung der Data Access Klassen sondern vielmehr die Einrichtung der erforderlichen Infrastruktur. In unseren JEE-Seminaren [4] erfahren Sie mehr zur Implementierung des DAO-Patterns. Gern unterstützen wir Sie auch beim Entwurf und der Implementierung ihres Entwicklungsprojekts.

Dr. Stefan Koch (info@ordix.de).



Rückblick DOAG Konferenz und Schulungstag 2009

Security-Profis treffen Strategen



Seit Jahren ist die Deutsche Oracle Anwenderkonferenz die Know-how-Plattform der deutschsprachigen Oracle Community. Rund 2.000 Teilnehmer besuchten dieses Jahr die Veranstaltung und freuten sich über rund 300 Vorträge und Schulungen. ORDIX und sein Tochterunternehmen coniatos lieferten ein interessantes Quartett an fachlichen und strategischen Vorträgen und luden an die ORDIX Infoinsel zum Dialog ein.

Technisch versiert ... beim Thema Oracle Security

Zwei technische Vorträge zum Thema Oracle Security [1] lieferten die ORDIX Datenbankspezialisten Klaus Reimers und Kathleen Hock. Um beispielsweise Wettbewerbsvorteile zu sichern oder gesetzlichen Anforderungen gerecht zu werden, müssen viele Unternehmen ihre Geschäftsdaten unter höchsten Sicherheitsanforderungen gegen externe Zugriffe und Missbrauch schützen.

Datenverschlüsselung

Ein möglicher Ansatz ist die Datenverschlüsselung, der sich Klaus Reimers in seinem Vortrag widmete. Oracle bietet hier verschiedene Ansätze, von der Verschlüsselung auf PL/SQL-Ebene bis hin zur Verschlüsselung von Kennwörtern bei DataPump oder von Backup Sets und Archivelog-Dateien in RMAN. Diese Möglichkeiten beleuchtete Klaus Reimers im Einzelnen, zeigte ihre Einsatzgebiete auf und lieferte Beispiele aus der Praxis.

Sicher per Default?

Kathleen Hock nahm einen anderen Sicherheitsaspekt unter die Lupe: „Secure by Default

in Oracle 11g“ [2] (siehe Seite 36). Geht das überhaupt? Oracle stellt in der Tat ein ganzes Set von Out-of-the-box Maßnahmen sowie einen verbesserten Passwortschutz und -management zur Verfügung, das die Hoffnung schürt, per Knopfdruck eine sichere Datenbank in Oracle 11g zu bekommen. Nach eingehender Prüfung kommt man zu der Einsicht, dass diese Maßnahmen aus Security-Sicht zwar eine gute Hilfestellung liefern, der DBA allerdings einige Auswirkungen zu berücksichtigen hat und auch weiterhin Hand anlegen muss, um seine Datenbank kontinuierlich sicherer auszulegen.

... und strategisch durchdacht

Die Kollegen der coniatos AG beschäftigten sich in ihren Vorträgen mit den strategischen Themen Risikomanagement und Enterprise Architecture Management.

Risikomanagement

Wie oben bereits erwähnt, kommen dem DBA eine ganze Reihe an Aufgaben zu, um die Systeme sicher zu gestalten und die Geschäftsdaten zu schützen. Besser ist es jedoch, proaktive Maßnahmen einzuleiten, um das Risiko schon im Vorfeld so gering wie möglich zu halten.

Das Thema „Risikomanagement“ [3] (siehe Abbildung 3) beleuchtete Lars Eisenblatt, Vorstand der coniatos AG. Er erläuterte zahlreiche Methoden und Checklisten, die bei der Identifikation, Bewertung, Umgang und Beherrschung von Risiken helfen. Damit lassen sich nachhaltig die Marktposition oder sogar die Existenz eines Unternehmens sichern und die Kosten kontrollieren. Darüber hinaus zeigte er auf, in welchen IT-Standards Risikomanagement verankert ist.

Agiles Enterprise Architecture Management (EAM)

„Der Blick auf das Ganze ist ein zentraler Aspekt, wenn es um die strategische Planung von IT-Landschaften geht“, so Matthias Heintz von der coniatos AG in seinem Vortrag. Beim EAM [4] + [5] müssen sowohl technologische Entwicklungen, vorgegebene Geschäftsprozesse als auch die Reduzierung der Komplexität der Architektur-Landschaften in Einklang gebracht werden (siehe Abbildung 2). Neben etablierten Frameworks, wie Zachman, FEAT und TOGAF, rücken immer mehr Prozesse und Verfahrensweisen in den Vordergrund, um kostengünstiger agieren zu können. In der Praxis zeigt sich, dass insbesondere Aspekte des agilen EAM in vielen Anwendungsfällen schneller und effizienter zum Ziel führen.

Objektorientierung in Oracle

In einem Auszug aus dem 5-tägigen Seminar [6] brachte Markus Fiegler seinen Teilnehmern beim DOAG-Schulungstag die von Oracle unterstützten, objektorientierten Konzepte wie Vererbung, Polymorphismus und benutzerdefinierte Konstruktoren näher. Die Abbildung von komplexen Strukturen und Klassenbeziehungen ist möglich. Objektorientierte Vorteile, wie Wiederverwendbarkeit, Modularität, Verbesserung der Wartbarkeit und Steigerung der Qualität der Software, können genutzt werden. Den Teilnehmern wurden die Grundprinzipien der Objektorientierung in einer Oracle Datenbank vermittelt. In mehreren Demos zeigte Markus Fiegler, wie man ein objektorientiertes Klassenmodell in Oracle umsetzt und wie man objektorientierte Objekte persistent in einer Datenbank verwaltet.

Stefanie Heither-Bürger (info@ordix.de).

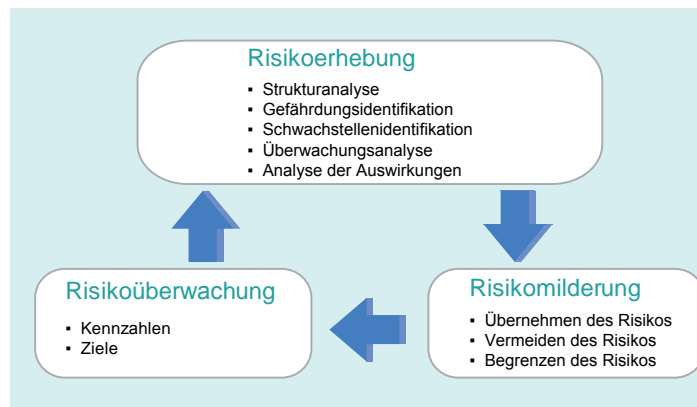


Abb. 1: Risikomanagement.

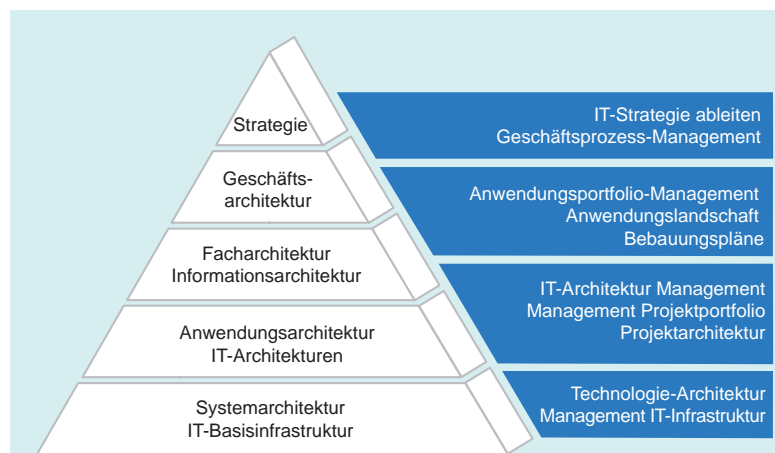


Abb. 2 Architekturpyramide.

Links

- ▶ [1] Seminarempfehlung „Oracle Security“:
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/837/index.html>
- ▶ [2] ORDIX News Artikelreihe „Oracle 11g Security - Secure by Default“
http://www.ordix.de/ORDIXNews/1_2010/Datenbank/secure_by_default.html
- ▶ [3] Seminarempfehlung „IT-Risikomanagement“
<http://training.ordix.de/siteengine/action/load/kategorie/IT-Management/nr/846/index.html>
- ▶ [4] ORDIX News Artikel
„Der Blick auf das Ganze: Management von IT-Unternehmensarchitekturen“
http://www.ordix.de/ORDIXNews/3_2009/IT-Strategie/it_management.html
- ▶ [5] Seminarempfehlung „IT-Architekturen“
<http://training.ordix.de/siteengine/action/load/kategorie/IT-Management/nr/843/index.html>
- ▶ [6] Seminarempfehlung „Objektorientierung in Oracle“
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/816/index.html>
- ▶ [7] Weitere Seminare im ORDIX Trainingsshop
<http://training.ordix.de>
- ▶ [8] Deutsche Oracle Anwendergruppe (DOAG)
<http://www.doag.org/>

15 % Rabatt für
DOAG Mitglieder
auf ORDIX Seminare



Objektorientierung in Oracle

Dieser Artikel richtet sich an Entscheider, Software-Architekten sowie Entwickler, die sich mit Objektorientierung in Oracle vertraut machen wollen.

Die Vorteile der objektorientierten Softwareentwicklung gegenüber der konventionellen, prozeduralen Programmierung ist nicht nur in der Fachliteratur, sondern auch in der Praxis anerkannt. Folglich bieten viele Entwicklersysteme, die in den letzten Jahren vermarktet wurden, die Möglichkeit der objektorientierten Programmierung. Wie sieht es mit der Objektorientierung in Oracle aus? Dieser Artikel gibt einen Einstieg, wie mit der Datenbankprogrammiersprache PL/SQL objektorientiert entwickelt werden kann. Dabei werden die Besonderheiten der Objektorientierung in Oracle aber auch die Möglichkeiten der Speicherung der Objekte in einer Datenbank vorgestellt.

Einführung und Motivation

Mit der objektorientierten Softwareentwicklung lassen sich komplexe Problemstellungen übersichtlich abbilden. Diese Vorgehensweise bietet gegenüber der konventionellen, prozeduralen Programmierung wesentliche Vorteile:

- Verständlicher Programmaufbau
- Geringerer Umfang an Programmcode durch Wiederverwendbarkeit
- Höhere Flexibilität bei Anforderungsänderungen
- Steigerung der Wartbarkeit
- Qualitätsverbesserung der Software

Mit der Version 8 hat Oracle einige objektorientierte Konzepte eingeführt, die mit den folgenden Versionen immer weiter ausgebaut und vervollständigt wurden. Diese objektorientierten Konzepte werden in den folgenden Abschnitten kurz vorgestellt.

Objekttypen versus Klasse

Objektorientierung unter Oracle orientiert sich einerseits an dem Package-Konzept (siehe Abbildung 1) aus PL/SQL und andererseits an objektorientierten Programmiersprachen wie Java.

Die Basis bilden die Objekttypen unter Oracle. Ein Objekttyp [1] ist ein benutzerdefinierter Datentyp und besteht aus einem Namen, mehreren Attributen und den dazugehörigen Methoden.

Eine Methode kann entweder eine Prozedur oder eine Funktion sein. Sie kann auf die Attribute des zugehörigen Objekttyps zugreifen und diese verändern.

Objekttypspezifikation

Die Attribute und Methoden eines Objekttyps werden, ähnlich einem PL/SQL-Paket, in einer Spezifikation (Objekttypspezifikation) deklariert. Die Spezifikation ist eine öffentliche Schnittstelle und kann von anderen Programmen aufgerufen werden.

Objekttypumpf

Die Implementierungen der in der Spezifikation deklarierten Prozeduren bzw. Funktionen befinden sich im Body (Objekttypumpf). Dabei muss jede im Body implementierte Prozedur bzw. Funktion auch in der Spezifikation deklariert werden.

Diese Einschränkung stellt einen Unterschied zu PL/SQL-Paketen dar. Denn hier können Prozeduren und Funktionen in einem Paket-Body implementiert werden, ohne diese in der Paket-Spezifikation deklarieren und somit nach außen sichtbar machen zu müssen.

Darüber hinaus können in einem Objekttyp-Body keine zusätzlichen Variablen deklariert werden. Auch diese Einschränkung stellt einen Unterschied zum PL/SQL Paket-Paradigma dar (siehe Abbildung 2).

STATIC- und MEMBER-Methoden

Bei den Methoden wird grundsätzlich zwischen der **STATIC**- und der **MEMBER**-Methode unterschieden. Ähnlich der anderen objektorientierten Programmiersprachen wird die **STATIC**-Methode unabhängig von Objekten aufgerufen.

Eine **MEMBER**-Methode wird dagegen nur im Zusammenhang mit einem Objekt aufgerufen. Sie verfügt über einen impliziten **SELF**-Parameter. Ein **SELF**-Parameter ist eine Referenz auf das Objekt selbst und kann innerhalb von **MEMBER**-Methoden verwendet werden.

Sichtbarkeit von Attributen und Methoden

Eine Besonderheit der Objektorientierung in Oracle gegenüber anderen objektorientierten

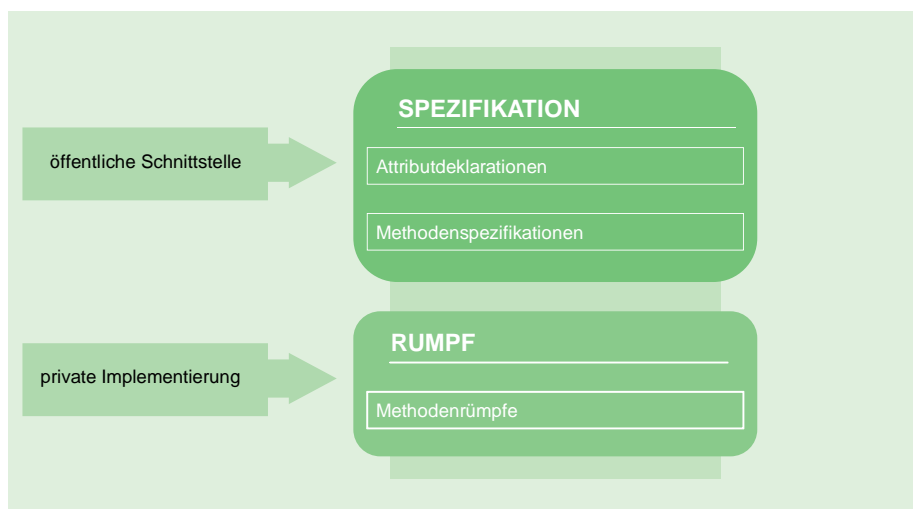


Abb. 1: Struktur eines Objekttyps.

```
CREATE TYPE kunde(
  nr      NUMBER(9),
  name   VARCHAR2(25),
  ort    VARCHAR2(30),
  MEMBER FUNCTION getName RETURN VARCHAR2,
  MEMBER PROCEDURE setName(name IN VARCHAR2)
);
/
CREATE TYPE BODY kunde AS
  MEMBER FUNCTION getName RETURN VARCHAR2 AS
  BEGIN
    RETURN name;
  END getName;

  MEMBER PROCEDURE setName(name IN VARCHAR2) IS
  BEGIN
    SELF.name := name;
  END setName;
END;
/
```

Abb. 2: Objekttyp KUNDE anlegen.

Programmiersprachen ist die Tatsache, dass in Oracle keine Unterscheidung von privaten (**PRIVATE**) und öffentlichen (**PUBLIC**) Attributen und Methoden vorhanden ist.

In PL/SQL kann also direkt, d. h. ohne öffentliche Methoden, auf die Attribute eines Objekts zugegriffen werden.

Speicherung der Objekte in der Datenbank

In einer Oracle Datenbank können Objekte sowohl als Spalten innerhalb einer relatio-

Objekttabelle		Relationale Tabelle		
OID	Objekt	Spalte1	Spalte2	Spalte3
A3160001	Objekt1	Wert1	Wert2	Wert3
A3160002	Objekt2	Wert4	Wert5	Wert6
A3160003	Objekt3	Wert7	Wert8	Wert9

Abb. 3: Speicherung der Objekte in der Datenbank.

```

DECLARE
  kundeObj KUNDE;
BEGIN
  SELECT VALUE(k) INTO kundeObj
  FROM kunde_tbl k WHERE k.nr = 5;
END;
/
    
```

Abb. 4: Auslesen eines Objekts aus einer Objekttabelle mit Hilfe der Funktion VALUE.

```

ALTER TYPE kunde ADD ATTRIBUTE (
  strasse VARCHAR2(50)
) CASCADE;
    
```

Abb. 5: Objekttyp KUNDE wird um ein Attribut STRASSE erweitert.



Abb. 6: Darstellung einer Referenz zwischen Objekttabellen.

```

CREATE TYPE BODY kunde AS
  MAP MEMBER FUNCTION f_sort RETURN NUMBER IS
  BEGIN
    RETURN nr;
  END f_sort;
END;
/
    
```

Abb. 7: Beispiel einer MAP-Funktion.

nen Tabelle, als auch in einer so genannten Objekttabelle (**object tables**) gespeichert werden. Eine Objekttabelle beinhaltet in einer Zeile ein Objekt eines vorgegebenen Objekttyps. Eine Spalte entspricht dabei einem Attribut eines Objekts (siehe Abbildung 3). Ein wesentlicher Vorteil von Objekttabellen gegenüber der Speicherung von Objekten als Spalte (relationale Tabelle) ist, dass auf Objekte in einer Objekttabelle referenziert werden kann (siehe **REF**-Spalten).

Der Zugriff und die Änderung der Daten einer Objekttabelle erfolgt wie bei einer herkömmlichen relationalen Tabelle. Wird allerdings statt der einzelnen Attribute ein Objekt benötigt, so kann mit der Funktion **VALUE** ein gesamtes Objekt ausgelesen werden (siehe Abbildung 4).

Type-Evolution

Sollen bereits vorhandene Objekttypen geändert werden, so kann dies mit Hilfe der so genannten Type-Evolution vorgenommen werden. Mit der Type-Evolution stellt Oracle eine Funktionalität zur Verfügung, mit der Objekttypen trotz abhängiger Datenbankobjekte, wie z. B. andere Objekttypen oder sogar Objekttabellen mit Objekten basierend auf dem Objekttyp selbst, modifiziert werden können.

Dabei können die Änderungen wie z. B. das Hinzufügen von neuen Attributen, das Löschen von bereits vorhandenen Attributen oder das Vergrößern der Länge einzelner Datentypen der Attribute, entweder unmittelbar oder zeitverzögert propagiert werden (siehe Abbildung 5).

REF-Spalten

Eine weitere Besonderheit der Objektorientierung in Oracle stellen die **REF**-Spalten dar. In einer **REF**-Spalte kann eine Referenz auf ein Objekt aus einer Objekttabelle gespeichert werden. Eine Objektreferenz ist dabei ein in der gesamten Datenbank eindeutiger Objektidentifizierer (Object identifier, **OID**).

Der Vorteil der **REF**-Spalte kommt immer dann zum Tragen, wenn statt des gesamten Objektes lediglich nur eine Objektreferenz gespeichert werden soll (siehe Abbildung 6).

Objektorientierte Konzepte

Des Weiteren werden in Oracle ähnlich wie bei den verbreiteten objektorientierten

Programmiersprachen alle herkömmlichen objektorientierten Konzepte unterstützt. Dazu gehören unter anderem:

- Vererbung
- Overloading
- Overriding
- Konstruktor
- Polymorphismus

Sortierung von Objekten

Soll auf Objekte, die in der Datenbank gespeichert sind, eine Sortierung in einem **SELECT**-Statement oder eine Überprüfung auf Gleichheit von Objekten vorgenommen werden, so muss entweder eine **MAP**- oder eine **ORDER**-Methode implementiert werden.

Durch die Verwendung der **MAP**-Methode werden alle Objekte eines Objekttyps in skalare Datentypen umgewandelt und anschließend sortiert (siehe Abbildung 7).

Eine **ORDER**-Methode wird dagegen mehrfach aufgerufen, da mit der **ORDER**-Methode immer nur zwei Objekte verglichen werden können (siehe Abbildung 8).

Wird eine große Anzahl von Objekten sortiert, so ist die **MAP**-Methode zu empfehlen.

Object Views

Mit Hilfe der so genannten Object Views ist eine objektorientierte Sicht auf relationale Daten möglich. Dabei können Object Views entweder auf relationalen Tabellen oder auch auf Objekttabellen basieren (siehe Abbildung 9).

Object Views können z. B. dann verwendet werden, wenn eine objektorientierte Softwareentwicklung auf bereits vorhandenen relationalen Daten realisiert werden soll.

Performance

Zum Thema Performance ist grundsätzlich zu sagen, dass die Objekttabellen die gleichen Möglichkeiten der Performance-Steigerung bieten, wie die relationale Tabelle auch.

Es ist allerdings zu beobachten, dass der Zugriff auf eine Objekttable mit dem standardmäßig, systemgenerierten **OID** langsamer ist, als ein Zugriff auf eine Objekttable mit einem auf dem Primärschlüssel basierenden **OID**.

```
CREATE TYPE BODY kunde AS
ORDER MEMBER FUNCTION f_sort(p_kunde kunde)
RETURN NUMBER IS
BEGIN
IF nr<p_kunde.nr THEN
RETURN -1;
ELSIF nr>p_kunde.nr THEN
RETURN 1;
ELSE
RETURN 0;
END IF;
END f_sort ;
END;
```

Abb. 8: Beispiel einer ORDER-Funktion.

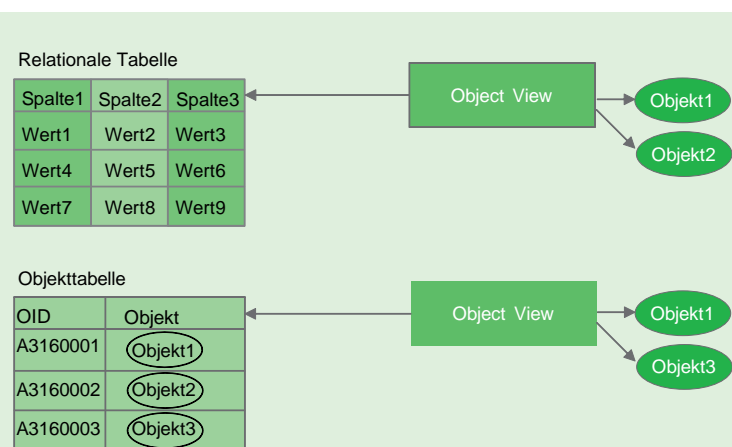


Abb. 9: Funktionsweise von Object Views.

Glossar

- O-/R-Mapping** Object-Relational-Mapping. O-/R-Mapping bezeichnet die Abbildung von objektorientierten Daten auf relationale Daten und umgekehrt.
- STATIC** Mit dem Schlüsselwort **STATIC** kann eine Methode in der Programmiersprache Java als eine Klassenmethode deklariert werden. Das Besondere an einer Klassenmethode ist, dass diese zu einer Klasse und nicht zu einem Objekt gehört. Außerdem ist im Speicher immer nur eine Kopie einer Klassenmethode vorhanden.
- OVERLOADING** Speicherung von gleich benannten Funktionen oder Prozeduren innerhalb der Datenbank, die sich nur in Anzahl, Reihenfolge und/oder Datentyp unterscheiden.
- Package** Stellt eine Sammlung logisch zusammengehöriger Funktionalitäten (Prozeduren, Funktionen, Typdeklarationen, ...) dar.
- PL/SQL** Procedural Language/Structured Query Language. Erweiterung der Abfragesprache SQL um prozedurale Programmier-elemente.

Der Grund hierfür liegt darin, dass die Objekttabellen mit dem systemgenerierten OID zusätzlich zu den Attributspalten noch über eine OID-Spalte (16 Byte) verfügen und somit mehr Speicherplatz belegen, als Objekttabellen mit einem auf dem Primärschlüssel basierenden OID.

Zusammenfassung

Die Objektorientierung in Oracle kann als eine Alternative zu der herkömmlichen prozeduralen Programmierweise verstanden werden. Da die wesentlichen Vorteile der Objektorientierung in der Wiederverwendbarkeit und Verbesserung der Wartbarkeit liegen, können

hier langfristige Kosteneinsparungen erzielt werden. Zudem kann der Arbeitsaufwand signifikant minimiert werden.

Kritisch sind hier allerdings die fehlende Unterstützung der Zugriffsmodifikatoren und damit auch die fehlende Unterstützung von Information Hiding. Da in Oracle zwischen privaten und öffentlichen Attributen nicht unterschieden werden kann, können alle Objektattribute ohne Verwendung der Objektmethoden verändert werden.

Die direkte Möglichkeit der Speicherung der Objekte in einer Objekttabelle stellt auf der anderen Seite aber einen deutlichen Performance-Vorteil gegenüber anderen objektorientierten Programmiersprachen dar, bei denen ein explizites Mapping zwischen Objekten und Tabellen notwendig ist.

Eine Vertiefung zu diesem Thema bietet Ihnen unser 5-tägiges Seminar „Objektorientierung in Oracle“ (siehe unten).

Markus Fiegler (info@ordix.de).

Link

- ▶ [1] ORDIX News Artikelreihe „Oracle Objekttypen von A-Z“
http://www.ordix.de/ORDIXNews/artikelreihen.html#oracle_objekttypen

Seminarempfehlung: Objektorientierung in Oracle

- ▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/816/index.html>

Oracle unterstützt auch objektorientierte Konzepte wie Vererbung, Polymorphismus und benutzerdefinierte Konstruktoren. Auch die Abbildung von komplexen Strukturen und Klassenbeziehungen ist möglich. Objektorientierte Vorteile wie Wiederverwendbarkeit, Modularität, Verbesserung der Wartbarkeit und Steigerung der Software-Qualität können genutzt werden. Dieses Seminar bietet den Einstieg zur Objektorientierung in Oracle.

Seminarinhalte

- Überblick/Architektur
- Design-Betrachtung
- Objekt-Typen mit Methoden
- Speicherung in der Datenbank
- Vererbung, Konstruktoren, Type Evolution
- Multilevel-Collection/Collection
- Ref-Spalten, Overloading, Polymorphismus
- ORDER/MAP-Funktion
- Trigger
- Dynamisches SQL
- Performance

Termine

- 17.05. - 21.05.2010 in Wiesbaden
- 02.08. - 06.08.2010 in Wiesbaden
- 11.10. - 15.10.2010 in Wiesbaden

Seminar-ID: DB-ORA-38

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>.



Neue Seminare 2010: IT-Management

Moderne IT-Technologien sind bereits heute das Fundament für Ihren Unternehmenserfolg. Über die gewinnbringende Ausrichtung von morgen entscheidet immer mehr die Qualität eines strategisch orientierten IT-Managements. Unsere neuen Seminare zeigen, wie ein solches IT-Management aufgebaut bzw. weiterentwickelt werden kann.

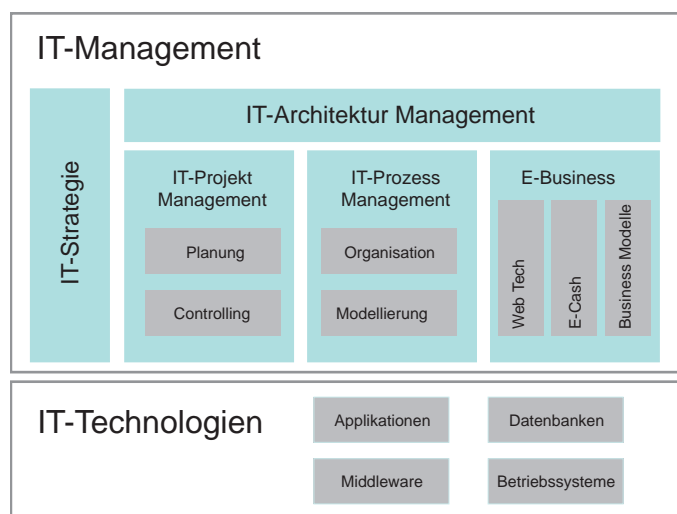


Fachliches Know-how strategisch aufbereitet

Die IT kommt heutzutage ohne strategisches Management nicht mehr aus. Wo früher rein technisches Know-how ausreichte, müssen heute strategische Management-Konzepte einen Rahmen bilden, um die immer komplexer werdenden Anforderungen an die IT überschaubar und strukturiert zu halten.

Mit unserem erweiterten Seminarangebot vermitteln wir Ihnen das Wissen, eigene IT-Strategien zu entwickeln und damit zum Geschäftserfolg Ihres Unternehmens maßgeblich beitragen zu können.

Stefanie Heither-Bürger (info@ordix.de).



Seminarempfehlung: IT-Management

► Informationen/Online-Anmeldung: <http://training.ordix.de/siteengine/action/load/kategorie/IT-Management/index.html>

Diese Seminare vermitteln Ihnen die wichtigsten Kernthemen des IT-Managements. Nach dem fundierten Einblick in die jeweiligen Spezialgebiete nimmt die Vorstellung aktueller Management-Instrumente einen breiten Raum ein.

Wir bieten folgende Seminare neu an:

- IT-Management
- IT-Prozessmanagement
- IT-Architekturen
- IT-Risikomanagement
- Systemisches Strategieentwicklung
- E-Business

CHRONIK 20 Jahre ORDIX

Bereits 1984 hatte der damals 29-jährige Wolfgang Kögler erste Gedanken, sich selbstständig zu machen. Nach seinem Studium zum Diplom-Mathematiker sammelte er erste Berufserfahrungen bei Siemens und landete 1987 auf Umwegen bei Nixdorf in Paderborn. Mit seinen Kollegen - teilweise noch aus der Siemens-Zeit - nahmen die Ideen zur Selbstständigkeit Form an. Das mühsame Tauziehen um Siemens und Nixdorf zu Siemens

Das erste Logo der
ORDIX Software GmbH



ORDIX steht für
Offene Systeme
und relationale Datenbanken unter Unix
und Windows.

Die ORDIX News
im Wandel der Zeit



1993



1998



1999

ORDIX entwickelt sich ...

- 03/1990: Wolfgang Kögler gründet die ORDIX Software GmbH.
05/1990: Eintrag der GmbH in das Handelsregister des Amtsgerichts Paderborn HRB 1757 und Aufnahme des Geschäftsbetriebes.
07/1990: Start mit 2 weiteren Mitarbeitern, eine davon ist heute noch bei ORDIX und heißt jetzt auch Kögler.
08/1990: Der ORDIX Schulungsbetrieb wird mit Unix Schulungen aufgenommen.
11/1990: Axel Röber, heute Vorstand für Technik und Entwicklung, arbeitet zunächst als freiberuflicher Mitarbeiter für ORDIX.
-
- 10/1991: Windows- und Arbeitsplatzsupport werden in das ORDIX Portfolio aufgenommen.
Umzug in die Geschäftsräume in der Westernmuer 12 - 16.
Abschluss des ersten vollen Geschäftsjahres. Neben Unix Schulungen werden Informix, SQL und spezielle SINIX Kurse angeboten.
-
- 1992: Mit Peter Seifert und Thomas Schonhoven wird in Köln die YARD Software GmbH gegründet, in diesem Zusammenhang entsteht die Geschäftsstelle Köln der ORDIX Software GmbH. Der Umsatz im zweiten vollen Geschäftsjahr verfehlt nur knapp die 2 Mio. DM Marke.
-
- 1993: Die 1. ORDIX News erscheint. Auflage 500 Stück, als Lose-Blatt-Sammlung, von Wolfgang Kögler eigenhändig zusammengetackert und kuvertiert. Die ORDIX GmbH beginnt mit der Oracle Beratung und wird offizieller Oracle Partner. Software-Entwicklung und Systemmanagement werden ins Portfolio aufgenommen.
-
- 03/1994: Christoph Lafeld, heute Vorstand Personal, tritt nach seinem Studium als Angestellter in die ORDIX ein. Axel Röber wechselt von der freiberuflichen Tätigkeit in die Festanstellung. Mit knapp 10 Mitarbeitern wird ein Umsatz von ca. 2,2 Mio. DM erwirtschaftet.
-
- 1995: ORDIX wird Ausbildungsbetrieb. Markus Fiegler, heute Senior Consultant, ist der erste Auszubildende. Verantwortlich für die Ausbildung ist Ulrike Kögler. Nachdem ORDIX bereits 1992 für SNI SAP R3 portiert hat und seit 1994 für SAP im Oracle Umfeld arbeitete, portieren wir nun für Sequent und NCR.
-
- 1996: ORDIX geht Partnerschaften mit BMC Software und Informix ein. Die Wandlung von der GmbH zur Aktiengesellschaft läutet den nächsten Schritt ein. Am 15. Mai 1997 erfolgt der Eintrag im Handelsregister des Amtsgerichtes Paderborn. Leider wird durch Fehler beim Amtsgericht eine neue HRB Nummer (HRB 2941) vergeben.
-
- 1997: Das erste Geschäftsjahr als Aktiengesellschaft endet am 30.06.1997 nach 9 Monaten mit einem Umsatz von 3,8 Mio. DM bei ca. 20 Mitarbeitern. Die ORDIX Geschäftsstelle/das Trainingszentrum in Wiesbaden wird eröffnet.
-
- 1998: Erste Java Projekte bereichern den Entwicklungsbereich bei ORDIX. Durch die Partnerschaft mit BMC wird ORDIX als Dienstleister Vorreiter beim Thema Systemmanagement. Am 30.06.1998 endet das erste volle Geschäftsjahr als Aktiengesellschaft, Umsatz 6,2 Mio. DM. Die Mitarbeiteranzahl lag bei ca. 30 Mitarbeitern.
-
- 1999: Der ORDIX Standort Nr. 4 wird in Münster eröffnet und wir werden Mitglied der Deutschen Oracle Anwendergruppe (DOAG). ORDIX begegnet dem Börsen- und Y2K-Hype in der IT-Branche skeptisch und wächst nur vorsichtig. Eine richtige Entscheidung, wie sich drei Jahre später herausstellte, als mehrere Firmen, die uns in dieser Zeit „überholt“ hatten, in Konkurs gingen.

19
19
19
19
19
19
19
19
19
19

ORDIX 2.0

1990 - 2010 | 20 Jahre bessere IT-Beratung

Nixdorf erschwerte das Vorhaben, so dass Wolfgang Kögler seinen Traum alleine weiterverfolgte. Im Januar 1990 akquirierte er bei einem Treffen mit einem Ex-Arbeitskollegen auf dem Rastplatz „Kirchheim-Ost“ den ersten offiziellen Auftrag für die noch in Gründung befindliche ORDIX Software GmbH. Die Marke „ORDIX“ war geboren!

... die (IT-)Welt auch!

- 90
91
92
93
94
95
96
97
98
99
- 1990: Tim Berners-Lee entwickelt das WorldWideWeb. Microsoft bringt die erste Windows Benutzer-Oberfläche auf den Markt. Unix bringt die Version 4.3BSD Reno heraus. Der Grüne Punkt als Kennzeichen für recyclebare Verpackungen wird in Deutschland eingeführt. Bei der Fußball-Weltmeisterschaft 1990 in Italien wurde die deutsche Fußballnationalmannschaft mit Trainer Franz Beckenbauer zum dritten Mal in ihrer Geschichte Weltmeister.
- 10/1990: Wiedervereinigung Deutschlands – Die Mauer fällt.
-
- 1991: Linus Torvald stellt seinen Kernel Linux vor. SUN Microsystems veröffentlicht Solaris 1.0. Bei Siemens Nixdorf wird SINIX 5.4 eingeführt. Freddie Mercury von der Band Queen stirbt an AIDS. Der „Soli“ wird eingeführt.
-
- 10/1992: Windows for Workgroups 3.1 wird veröffentlicht.
1992: Der Vertrag von Maastricht als Grundlage der EU wird unterzeichnet. Er tritt am 1. November 1993 in Kraft. Bill Clinton wird Präsident der USA.
-
- 12/1993: Veröffentlichung der Datenbank DB2 z/OS in der Version 1.
1993: Mehrwertsteuer-Erhöhung in Deutschland von 14 % auf 15 %. Deutschland erhält die fünfstelligen Postleitzahlen.
-
- 1994: Nach Urheberrechtsstreitigkeiten zwischen Novell, USL und BSDI wurde sämtlicher auf USL-Entwicklungen basierender Quellcode aus 4.4BSD entfernt, und es kam zur Veröffentlichung von „4.4BSD lite“. Die freien BSDs brachten ebenfalls neue, auf „4.4BSD lite“ basierende Versionen heraus. Die letzten Schulterpolster verschwinden aus der Modewelt.
-
- 1995: Oracle kauft für 100 Millionen Dollar die OLAP-Produktlinie (Express Server) der Information Resources Inc. (IRI). Aus der Deutschen Bundespost wird durch Privatisierung die Deutsche Post AG. Auch die Deutsche Telekom entsteht aus der bereits ausgegliederten Deutschen Bundespost Telekom.
-
- 1996: Die erste Version des Java Development Kit, das JDK 1.0 wird veröffentlicht. Das Schaf Dolly, erstes geklontes Säugetier der Welt, wird geboren. Deutschland wird in England Fußball-Europameister durch ein „Golden Goal“ von Oliver Bierhoff. Mit Deep Blue bezwingt erstmals ein Schachcomputer einen Schachweltmeister, Garry Kasparow, in einer Partie unter Turnierbedingungen.
-
- 1997: Die Open Group veröffentlicht den Standard „Single UNIX Spec/2“, der jetzt Echtzeit, Threads und 64-Bit-Prozessoren unterstützt. Bundespräsident Roman Herzog hält seine bekannte Berliner Rede, in der er fordert, durch Deutschland müsse ein „Ruck“ gehen. Prinzessin Diana stirbt bei einem Autounfall in Paris. Das Tamagotchi wird zu einer Modeerscheinung.
-
- 09/ 1998: Google wird geboren
1998: Mehrwertsteuer-Erhöhung in Deutschland von 15 % auf 16 %. Regierungswechsel in Deutschland. Ende der 16 Jahre dauernden Ära Kohl. Neuer Bundeskanzler ist Gerhard Schröder. Es wird populär, per SMS zu kommunizieren
-
- 01/1999: Das JDK 1.2 wird in „Java 2 platform“ umbenannt.
1999: Einführung der europäischen Gemeinschaftswährung Euro (€) als Buchgeld. Die Anzahl der weltweit lebenden Menschen übersprang nach Berechnungen der UN erstmals die 6-Milliarden-Marke.

Ohne geht es nicht mehr



Linux-Maskottchen



Resultat eines Wettbewerbs

Windows in den Kinderschuhen



Google Logo - Früher



Google Logo - Heute



Fortsetzung folgt ...

Den Rückblick für die Jahre 2000 - 2010 finden Sie in der ORDIX News Ausgabe 2/2010.

März - August 2010

	Dauer	Preis	März		April		Mai		Juni		Juli		August														
			KW 10	KW 11	KW 12	KW 13	KW 14	KW 15	KW 16	KW 17	KW 18	KW 19	KW 20	KW 21	KW 22	KW 23	KW 24	KW 25	KW 26	KW 27	KW 28	KW 29	KW 30	KW 31	KW 32	KW 33	KW 34
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	März		April				Mai				Juni				Juli				August						
			KW 10	KW 11	KW 12	KW 13	KW 14	KW 15	KW 16	KW 17	KW 18	KW 19	KW 20	KW 21	KW 22	KW 23	KW 24	KW 25	KW 26	KW 27	KW 28	KW 29	KW 30	KW 31	KW 32	KW 33	KW 34
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.



Oracle auf NetApp (Teil III)

NetApp SMO - der SnapManager für Oracle

Der Artikel richtet sich an Oracle Datenbankadministratoren, Storage-Verwalter und/oder Mitarbeiter aus Supportteams, die sich thematisch mit dem Sichern und Wiederherstellen von Oracle Datenbanken beschäftigen.

In der ORDIX News Ausgabe 2/2009 [1] hatten wir Ihnen im Rahmen dieser Artikelreihe bereits vorgestellt, wie NetApp Snapshots als Sicherheitsstrategie für Oracle Datenbanken genutzt werden können. In diesem Artikel stellen wir Ihnen das NetApp-Produkt SnapManager für Oracle (SMO) vor, mit dessen Hilfe Sie Datenbanksicherungen planen, erstellen, verwalten und natürlich auch wieder zurückspielen können.

Snapshots leicht gemacht

Mit dem Produkt SMO gibt NetApp den Datenbank- und/oder Storage-Administratoren ein Werkzeug an die Hand, mit dem Oracle Datenbanken einfach und unkompliziert gesichert und wiederhergestellt werden können. Den grundlegenden Prozess einer Datenbanksicherung über NetApp Snapshots haben wir Ihnen ja bereits vorgestellt [1].

Der Snapmanager (SMO) vereinfacht in vielerlei Hinsicht die Steuerung dieses Prozesses. Zum einen erleichtert die grafische Oberfläche von SMO die Erstellung und Verwaltung von Backups erheblich. Zum anderen kann ein DBA über dieses Tool seine Backups auf den Storage-Systemen selbstständig verwalten, ohne explizit Berechtigungen auf den

Files zu erhalten oder permanent Rücksprache mit den Administratoren dieser Systeme halten zu müssen. Neben einer grafischen Oberfläche bietet das Tool natürlich auch ein CLI (Command Line Interface), das sich in Skripte oder z. B. in eine Crontab einbinden lässt.

Integration leicht gemacht

SMO integriert sich nahezu problemlos in die Oracle Architektur. Es spielt keine Rolle, ob der Datenbank-Storage per NFS, iSCSI oder Fibre Channel angebunden ist. Lediglich per CIFS genutzte Storage-Bereiche lassen sich per SMO nicht sichern (dies dürfte in der Praxis aber auch relativ selten vorkommen). Darüber hinaus lassen sich Single Instances ebenso einfach wie RAC-Installationen sichern

und wiederherstellen. ASM stellt SMO ebenfalls vor keinerlei Probleme. Zusätzlich lassen sich per SMO erstellte Backups einfach in den Oracle Recovery Catalog integrieren und können im Bedarfsfall auch über den RMAN eingesehen und verwaltet werden.

Installation Teil I

Die Installation von SMO gestaltet sich recht einfach. Die entsprechenden Binary-Pakete können sich Kunden – sofern sie alle notwendigen Lizenzen besitzen – herunterladen und installieren [3]. Zusätzlich zum SMO-Paket wird ein weiteres Werkzeug benötigt: SnapDrive [4]. SnapDrive dient als Kommunikationsschnittstelle zwischen SMO und dem Filer. SnapDrive übernimmt die Zuordnung von lokal eingehängten Volumes und Devices zu den entsprechenden Filer-Systemen. Das heißt SnapDrive erkennt, welches Device von welchem Filer bezogen wird, bzw. ermittelt, ob ein Verzeichnis überhaupt auf einem NetApp Filer liegt. Zudem übernimmt SnapDrive die Verwaltung der Snapshots auf den entsprechenden Filer-Systemen (siehe Abbildung 1).

Beide Produkte - SMO und SnapDrive - können mit wenigen Befehlen konfiguriert und als Dienst auf dem Datenbankserver in Betrieb genommen werden. Zur Erstellung und Verwaltung von Backups kann dann wahlweise mit CLI-Kommandos oder mit einer Oberfläche gearbeitet werden, die auf Java Webstart basiert.

Zusätzlich zu den Softwarekomponenten benötigen die angebotenen Filer jeweils eine FlexClone- und eine SnapRestore-Lizenz (siehe hierzu auch [2]).

Installation Teil II

Bevor per SMO die ersten Backups erzeugt werden, sind noch weitere Vorbereitungen zu treffen. Die Metadaten der erstellten Backups werden in einer separaten Oracle-Datenbank verwaltet. Im Prinzip erfüllt diese Datenbank die gleiche Aufgabe wie der Oracle Recovery Catalog. Zusätzlich müssen so genannte Profile erstellt werden. Hinter einem Profil verbergen sich wesentliche Informationen über die zu sichernde Oracle Instanz sowie umfangreiche Retention-Einstellungen. Darüber hinaus kann in einem Profil u. a. hinterlegt werden, ob eine Instanz „online“ oder „offline“ gesichert wird und ob eine Teilsicherung (z. B. einzelne Datafiles und/oder

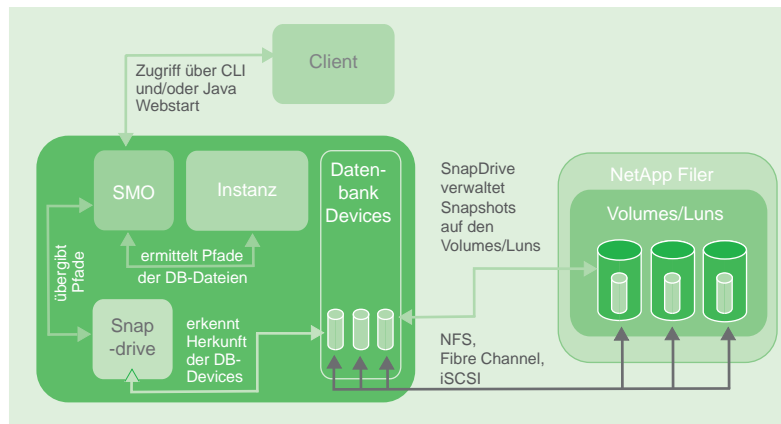


Abb. 1: SMO erfragt die Pfade aller Datenbankkomponenten bei der Instanz.

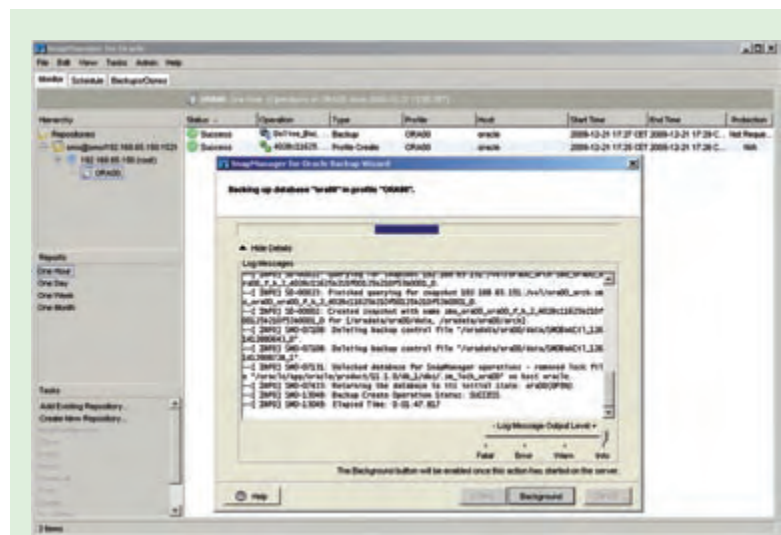


Abb. 2: SMO-Web-Oberfläche, die Erstellung eines Backups erfolgt per Mausklick.

Tablespaces) oder eine Vollsicherung erfolgen soll. Die Profile können gegen unbefugten Zugriff per Passwort geschützt werden.

Umstrukturierung?

Zusätzlich zu den Installationstätigkeiten ist auch datenbankseitig einiges zu bedenken. Das Layout der Datenbankdateien ist ggf. bei der Verwendung von SMO anzupassen (siehe hierzu auch [1]). In den entsprechenden Installationsleitfäden sind Hinweise für alle möglichen Produkt- (RAC, ASM, Single Instance) und Protokollkombinationen (NFS, iSCSI, FC) hinterlegt. An dieser Stelle sollen

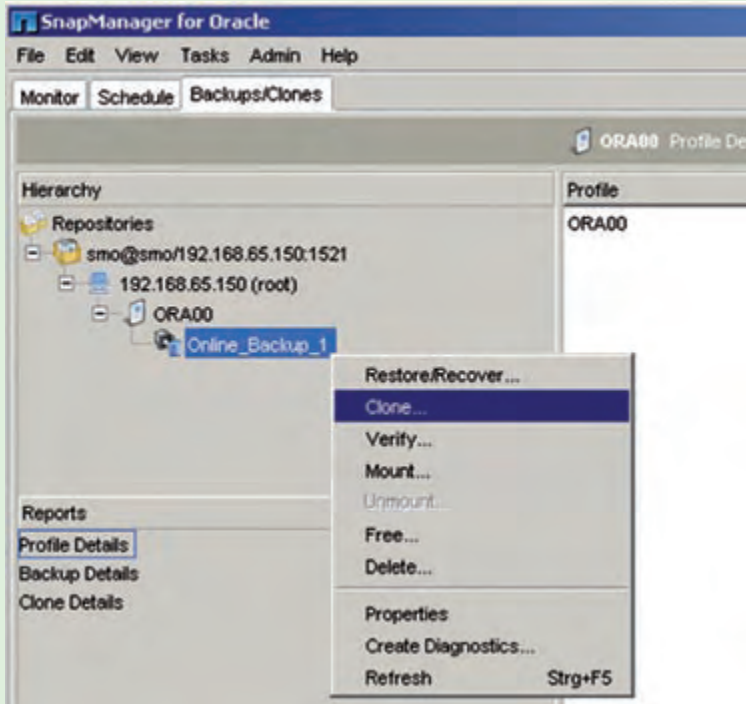


Abb. 3: SMO bietet neben dem reinen Backup- und Recovery-Funktionen noch weitere interessante Möglichkeiten. So können Backups verifiziert und als Basis für Datenbankkopien (Klone) genutzt werden.

Glossar

Retention	Eine Retention ist eine unbedingt einzuhaltende Vorgabe, wie viele oder wie lange Backups aufzuheben sind (Aufbewahrungsfrist). Sicherungen, die aus einer Retention „herausfallen“ werden ggf. automatisch gelöscht.
CLI	Command Line Interface. Wird oft auch als Konsole oder Terminal bezeichnet. Das CLI ist ein Eingabebereich zur Steuerung bzw. zum Aufruf von Programmen.
Crontab	Abgeleitet von den Begriffen „chronos => Zeit“ und „tabula => Tabelle“. Über die Crontab können auf Unix/Linux-Systemen automatisiert wiederkehrende Aufgaben (Programme/Skripte) ausgeführt werden.
FC	Fibre Channel - ein Standardprotokoll aus dem Bereich der Speichernetzwerke.
ASM	Automatic Storage Management. ASM ist ein im Oracle Kernel integrierter Volume Manager. Er ist ab Oracle 10g verfügbar und arbeitet nach dem SAME-Prinzip (stripe and mirror everything). ASM kann als zentraler Storage für RAC-Systeme verwendet werden.

die Vorgaben für einige Oracle Instanzen, deren Datenbankdateien über das NFS-Protokoll angebunden werden, grob skizziert werden.

Für diesen Fall wird dringend empfohlen, dass separate Volumes für Datenbankdateien (DBFs), Control-Files, Online-Redo-Logs und archivierte Redo-Logs bereitgestellt werden.

Dies ist notwendig, da die jeweiligen Komponenten der Datenbanken zu unterschiedlichen Zeitpunkten des Sicherungsprozesses per Snapshot gesichert werden. Bei einer Mischung der Datenbankkomponenten weigert sich SMO im schlimmsten Fall, ein Backup durchzuführen.

Diese Vorgaben bei einer Neuinstallation einzuhalten, ist – verglichen mit dem Vorhaben, eine mehrere Terabyte große Datenbank den Konventionen entsprechend umzustrukturieren – recht einfach.

Zweigleisig zum Ziel

Wie bereits erwähnt, kann SMO sowohl über Kommandos als auch per Weboberfläche genutzt werden. Im Folgenden werden wir jedoch mehr auf die Weboberfläche eingehen. Das Programm präsentiert sich sehr aufgeräumt und intuitiv. Auf einen Blick lassen sich alle Profile eines Repositories erfassen (siehe Abbildung 2). In unserem Beispiel sehen wir eine Repository-Datenbank, in der sich lediglich ein Profil für die Instanz ORA00 auf dem Datenbankserver 192.168.65.150 befindet.

Funktionen, Funktionen, Funktionen

Die erstellten Profile bilden die Grundlage für diverse Funktionen. Neben dem Erstellen und dem Zurücksichern von Backups bietet SMO noch einige andere interessante Anwendungen (siehe Abbildung 3).

- Erstellen von Backup-Zeitplänen (inkl. Aufbewahrungsfristen/Retentions)
- Verifizierung von Backups über das Oracle Werkzeug „dbverify“
- Klonen von Datenbanken. Hier dient als Basis für einen Klon ein erstelltes Backup. (siehe Abbildung 3)
- Mounten von Backups (also des entsprechenden Snapshots)

Bei einer entsprechend sorgsamem Vorbereitung der Datenbanken und einer sehr genauen Lektüre der Vorgaben des entsprechenden Best-Practice-Guides [5] lassen sich alle Funktionen problemlos nutzen. Probleme entstehen meist durch ein fehlerhaftes Datenbanklayout (z. B. Ablage von Controlfile und Datafiles in einem Volume) oder durch Nachlässigkeiten bei der Anbindung des Storage an den Datenbankserver (z. B. Nutzung oder Weglassen bestimmter NFS Mount Optionen).

Fazit

Mit dem SnapManger für Oracle stellt NetApp ein leistungsfähiges Tool zur Erstellung und Verwaltung von Oracle Datenbank-sicherungen bereit. Die Installation ist – bei entsprechender Lektüre der jeweiligen Handbücher – einfach und die Nutzung des Werkzeugs intuitiv.

Mit Hilfe dieses Produktes können Oracle Datenbanken auch ohne großes Hintergrundwissen gesichert, geklont und wiederhergestellt werden. Das Produkt ist daher z. B. gut geeignet für den First-Level-Support oder für sehr große Oracle-Umgebungen mit vielen Instanzen, deren Backups zentral gesteuert und auch überwacht werden sollen.

Matthias Jung (info@ordix.de).

Links

- ▶ [1] ORDIX News Artikel „Oracle auf NetApp (Teil I): Bitte recht freundlich: Snapshots als Sicherungskonzept“
http://www.ordix.de/ORDIXNews/2_2009/Datenbanken/oracle_netapp.html
- ▶ [2] ORDIX News Artikel „Oracle auf NetApp (Teil II): Spieglein, Spieglein an der Wand ...“
http://www.ordix.de/ORDIXNews/4_2009/Datenbanken/oracle_netapp2.html
- ▶ [3] Oracle/NetApp SnapManager
<http://www.netapp.com/us/products/management-software/snapmanager-oracle.html>
- ▶ [4] Oracle/NetApp SnapDrive Unix/Windows
<http://www.netapp.com/us/products/management-software/>
- ▶ [5] Oracle/NetApp Best Practice
<http://media.netapp.com/documents/tr-3369.pdf>



NetApp SnapManager für Oracle (SMO) im Expertencheck

1. Einrichtung	■■■■■■■■■□□
2. Konfiguration (GUI)	■■■■■■■■■□□
3. Konfiguration (CLI)	■■■□□□□□□□
4. Administrierbarkeit (GUI, CLI)	■■■■■■■■■□□
5. Performance	■■■■■■■■■□□
6. Funktionsumfang	■■■■■■■■■□
7. Dokumentation	■■■■■■■■■□□

positiv: ■■■■■■■■■■

negativ: □□□□□□□□



„SMO ist eine interessante Alternative zu anderen Oracle Backup Produkten. Gerade in großen Oracle-Umgebungen, mit dedizierten Teamstrukturen und Supporteinheiten, kann SMO mit seiner grafischen Oberfläche glänzen. Backups und Restores lassen sich so - auch von fachfremden Personen - einfach erstellen und verwalten.“

Matthias Jung, ORDIX AG, Paderborn



Neue Reihe: Server-Virtualisierung mit XEN – Neues und Spezielles (Teil I)

XEN – Der Stern leuchtet noch

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

In der ORDIX News 3/2006 haben wir eine Artikelreihe zum Thema XEN begonnen, um über die Vorzüge dieser OpenSource Virtualisierungslösung zu berichten. Mittlerweile schreiben wir das Jahr 2010. In den letzten vier Jahren haben wir weitere Projekte und Schulungen zu diesem Thema durchgeführt. Mittlerweile gibt es kommerzielle XEN-Versionen und XEN hat die Version 3.4.2 mit zahlreichen neuen Funktionen erreicht. Mit diesem Artikel gehen wir auf neue Möglichkeiten von XEN ein, die in den letzten Jahren entwickelt wurden.

Textdateien – Gibt es die noch?

Klassischerweise legt man für jeden XEN-Gast eine textbasierte Konfigurationsdatei an, in der die für den Gast notwendigen Parameter, wie die Menge des zugewiesenen RAM, Anzahl der CPU's, usw. festgelegt werden. Mit dem Befehl `xm create <Name der Konfigurationsdatei>` kann nun der Gast mit den eingetragenen Einstellungen gestartet werden.

Eine besondere Funktion von XEN ist, dass fast jeder Parameter online verändert werden kann. So kann zum Beispiel mit dem Befehl `xm mem-set suse1 1024` der zugewiesene Arbeitsspeicher auf 1024 MB vergrößert werden oder mit `xm vcpu-set suse1 4` die Anzahl der zugewiesenen CPU's geändert werden.

Das Problem bei dieser Art der Administration ist, dass der online zugewiesene neue Wert

beim Herunterfahren des Gastes verloren geht. Soll bei zukünftigen Starts des Gastes weiterhin die höhere Menge an Arbeitsspeicher verwendet werden, muss der Administrator zusätzlich die Konfigurationsdatei des Gastes anpassen. Das Problem vervielfacht sich spätestens dann, wenn man mit mehreren XEN-Servern arbeitet und die Funktion der Live-Migration nutzt. Hier müssen bei Änderungen der Konfiguration auf allen beteiligten Systemen die notwendigen Anpassungen durchgeführt werden.

XenStore – Registry für XEN?

Zur Beseitigung des Problems mit den Konfigurationsdateien haben die Entwickler eine Lösung programmiert und diese „XenStore“ getauft. XenStore ist eine zentrale Konfigurationsdatenbank, auf die alle XEN-Domains zu-

greifen können. Um XenStore nutzen zu können, benötigt man zum Anlegen des Gastes nur eine Konfigurationsdatei. Hinterher wird sie nicht mehr benötigt und kann gelöscht werden. Für die durch XenStore verwalteten Gäste haben die Entwickler einige zusätzliche Verwaltungsbefehle implementiert, mit denen diese Art von virtuellen Gästen administriert werden (siehe Abbildung 1).

Die XenStore-Datenbank könnte man mit der Registrierungsdatenbank von Windows vergleichen. Es gibt verschiedene Hauptschlüssel, unter denen sich verschiedene Informationen über die Domains auslesen und verändern lassen. Für den Zugriff auf die XenStore-Datenbank stehen Befehle zum Auslesen und Verändern von Schlüsseln und Werten zu Verfügung. Diese Datenbank enthält für die Administration 3 Hauptschlüssel, die im Folgenden beschrieben werden:

- **/vm**
Unterhalb des Schlüssels `/vm` gibt es für jede Domain einen Eintrag. Die Domains sind hier aber nicht mit dem eigentlichen Namen aufgelistet, sondern mit einer eindeutigen UUID. Die Domain 0 hat immer dieselbe UUID: 00000000-0000-0000-0000-000000000000, während für die Domain U eine zufällige erzeugt bzw. der vom Administrator definierte Wert verwendet wird. Mit `xm list -l suse1` kann der Wert für einen Gast ermittelt werden.
- **/local/domain**
Unter dem Schlüssel `/local/domain` sind weitere Konfigurationseinstellungen der einzelnen Domains abgelegt. Allerdings sind diese hier nicht über eine UUID ansprechbar, sondern über eine ID. Die Domain 0 hat automatisch immer die ID 0. Startet man den ersten Gast, bekommt er die ID 1. Wird der Gast nun neu gestartet, wird danach für den Gast die ID 2 angezeigt. Mit dem Befehl `xm list` kann man sich für alle Domains die ID's anzeigen lassen, mit dem Befehl `xm domid suse1` wird die ID für eine bestimmte Domain angezeigt.
- **/tool**
Der Schlüssel `/tool` enthält standardmäßig nur einen leeren Unterschlüssel mit dem Namen "xenstored". Der Schlüssel ist für Einträge von Drittanbieterwerkzeugen gedacht, kann aber auch für eigene Zwecke genutzt werden.

Stellt man von den bisher verwendeten Konfigurationsdateien auf die XenStore basierte Konfiguration um, gibt es noch eine Beson-

<code>xm new</code>	Fügt einen Gast mit Hilfe einer Konfigurationsdatei der XenStore-Datenbank hinzu
<code>xm delete</code>	Löscht einen Gast wieder aus der XenStore Datenbank
<code>xm start</code>	Startet einen XenStore-basierten Gast
<code>xm suspend</code>	Hält die Ausführung des Gastes an
<code>xm resume</code>	Führt einen angehaltenen Gast wieder aus

Abb. 1: Verwaltungsbefehle für Gäste, die auf XenStore basieren.

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	1706	2	r-----	15418.4
suse1	5	512	1	-b----	9.9
suse2		512	1		7.5
suse3		512	1		4.2
winxp	2	512	1	-b----	225.5

Abb. 2: Auflistung der Domains mit `xm list`.

derheit zwischen diesen beiden Verfahren. Arbeitet man mit Konfigurationsdateien, wird ein Gast, wenn er nicht gestartet ist, mit `xm list` auch nicht angezeigt. Erst wenn er mit `xm create suse1` gestartet wurde, bekommt er eine ID zugeordnet und wird in der Auflistung auch angezeigt. Ein XenStore-Gast wiederum wird immer mit `xm list` angezeigt. Anhand der vorhandenen ID wird ersichtlich, ob der Gast gestartet ist oder nicht. Abbildung 2 zeigt die Ausgabe des Befehls `xm list`. Die Gäste `suse2` und `suse3` sind aktuell nicht gestartet, werden aber trotzdem angezeigt, da sie von XenStore verwaltet werden.

Änderungen an einem Gast können nun entweder mit den üblichen Befehlen online und dauerhaft vorgenommen werden oder aber mit XenStore-Befehlen. Ein Beispiel für eine Änderung per Befehl wäre `xm rename suse1 suse11`, bei der ein Gast umbenannt wird. Für einige Konfigurationsparameter gibt es bisher keine Möglichkeit, diese mit einem Befehl zu ändern, so dass immer eine Änderung der Konfigurationsdatei mit anschließendem Neustart des Gastes notwendig war. Mit den XenStore-Befehlen wiederum lassen sich alle Parameter ändern. Die Änderungen sind ohne Neustart des Gastes sofort aktiv. Im Folgenden nun die wichtigsten Befehle mit Beispielen.

xenstore-ls

Mit dem Befehl `xenstore-ls` (ohne weitere Argumente) wird dem Administrator die komplette XenStore-Datenbank ange-

```
# xenstore-ls /vm/f1fc911d-744c-70a2-f02a-60861440db16
.....
on_xend_stop = "ignore"
shadow_memory = "0"
uuid = "f1fc911d-744c-70a2-f02a-60861440db16"
on_reboot = "destroy"
start_time = "1264323716.0"
on_poweroff = "destroy"
bootloader_args = "-q"
on_xend_start = "ignore"
on_crash = "restart"
xend = ""
  restart_count = "0"
vcpus = "1"
vcpu_avail = "1"
bootloader = "/usr/bin/pygrub"
name = "suse1"
```

Abb. 3: Anwendung und Ausgabe des Befehls `xenstore-ls`.

```
# xenstore-read /vm/f1fc911d-744c-70a2-f02a-60861440db16/
name
suse1
# xenstore-read /vm/f1fc911d-744c-70a2-f02a-60861440db16/
on_poweroff
destroy
```

Abb. 4: Abfrage von Schlüsselwerten mit `xenstore-read`.

```
# xenstore-exists /vm/f1fc911d-744c-70a2-f02a-
60861440db16/on_poweroff
# echo $?
0
# xenstore-exists /vm/f1fc911d-744c-70a2-f02a-
60861440db16/Schluessel_gibt_es_nicht
# echo $?
1
```

Abb. 5: Ausgabe des Befehls `xenstore-exists`.

```
# xenstore-write /vm/f1fc911d-744c-70a2-f02a-60861440db16/name
sles11
# xenstore-write /vm/f1fc911d-744c-70a2-f02a-60861440db16/
on_poweroff restart
```

Abb. 6: Verändern von Werten mit `xenstore-write`.

```
# xenstore-rm /tool/ f1fc911d-744c-70a2-f02a-60861440db16/
Gastkommentar
```

Abb. 7: Löschen eines Schlüssels mit `XenStore-rm`.

zeigt. Mit `xenstore-ls /vm/f1fc911d-744c-70a2-f02a-60861440db16` kann dies auf eine bestimmte Domain beschränkt werden. Abbildung 3 zeigt eine gekürzte Ausgabe des Befehls.

xenstore-read

Soll nur ein bestimmter Wert aus der Datenbank ausgelesen werden, hilft der Befehl `xenstore-read`, bei dem der Name des anzuzeigenden Schlüssels angegeben werden muss. Abbildung 4 zeigt zwei Beispiele, die abgefragt werden.

xenstore-exists

Zur Prüfung, ob ein bestimmter Schlüssel vorhanden ist, kann `xenstore-exists` (siehe Abbildung 5) ausgeführt werden. Hier wird mit Hilfe von Shell-Rückgabewerten dargestellt, ob ein Schlüssel vorhanden ist oder nicht.

xenstore-write

Um den Wert eines Schlüssels nun zu verändern, wird der Befehl `xenstore-write` (siehe Abbildung 6) verwendet. Hierbei müssen der Name des Schlüssels und ein gültiger Wert bekannt sein. Ein bestehender Wert wird hierbei einfach mit einem neuen Wert überschrieben. Im ersten Beispiel wird der Name des Gastes geändert, was sofort mit `xm list` geprüft werden kann. Im zweiten Beispiel bewirkt ein Herunterfahren innerhalb des Gastes jetzt kein Ausschalten mehr, sondern dass der Gast automatisch wieder gestartet wird.

xenstore-rm

Der Vollständigkeit halber wurde von den Entwicklern ein Werkzeug spendiert, mit dem Einträge aus der Konfigurationsdatei entfernt werden können. In der Regel wird man Löschungen maximal im `/tool`-Zweig von XenStore durchführen. Abbildung 7 zeigt die Syntax des Befehls.

Resumee

Mit Hilfe der XenStore-Gastverwaltung haben die Entwickler eine praktische Möglichkeit geschaffen, Gäste noch einfacher zu verwalten. Besonders in Verbindung mit Live-Migrationen muss nicht mehr geprüft werden,

ob die Konfigurationsdatei auf dem Zielsystem schon die aktuellen Werte besitzt, da beim Migrieren automatisch die aktuelle Konfiguration mitgezogen wird. Mit Hilfe der Befehle `xenstore-*` kann jetzt praktisch jeder Wert von Hand oder per Skript geändert werden, ohne dass ein Neustart benötigt wird. Im nächsten Teil dieser Reihe beschäftigen wir uns mit dem Thema Festplattenverwaltung unter XEN.

Christian Fertsch (info@ordix.de).

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.
Live-Migration	Eine Funktion von XEN, bei dem ein laufender Gast online von einem physikalischen Wirt auf einen anderen migriert wird. Anwendungen und Netzwerkverbindungen werden weiterhin ausgeführt.
UUID	Universally Unique Identifier. Eine UUID wird genutzt, um Informationen in verteilten Systemen ohne zentrale Koordination eindeutig kennzeichnen zu können.

Link

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

Seminarempfehlung: Server-Virtualisierung mit Xen

▶ **Informationen/Online-Anmeldung:** <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Betriebssysteme/nr/579/index.html>

Server-Virtualisierung - ein Konzept der Mainframes - wird auch auf PC-bzw. Unix-Architekturen zunehmend populärer. In diesem Seminar werden Sie mit der Installation, Konfiguration und Administration von XEN-basierten Systemen vertraut gemacht und sind danach in der Lage, ein XEN-System aufzusetzen. Viele praktische Übungen und Beispiele helfen Ihnen dabei, die Seminarinhalte zu verstehen und anzuwenden.

Seminarinhalte

- Aufbau und Architektur von XEN
- Systemvoraussetzungen von Hard- und Software
- Installation und Konfiguration von XEN
- Erstellung eines eigenen XEN Kernel
- Konfiguration des Bootmanagers und XEN-Parameters für verschiedene Einsatzszenarien
- Wichtige Dateien und Verzeichnisse von XEN
- Installation von Linux Gastsystemen
- Netzwerkkonfiguration und XEN Switch
- Routing und Subnetzbildung virtueller Gäste
- Live-Migration von virtuellen Gästen
- XEN-Troubleshooting
- Andere XEN-Gäste
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

26.05. - 28.05.2010 in Wiesbaden
16.08. - 18.08.2010 in Wiesbaden
18.10. - 20.10.2010 in Wiesbaden
13.12. - 15.12.2010 in Wiesbaden

Seminar-ID: BS-13

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>.

„Secure by Default“ in Oracle 11g – Die sichere Datenbank out-of-the-box?

Dieser Artikel richtet sich an Datenbankentwickler, Sicherheitsbeauftragte und Datenbankadministratoren, die für den Bereich Security verantwortlich sind.

Oracle hat viel unternommen, um die aktuelle Datenbank Version 11g sicherer zu gestalten. Sie liefert zahlreiche neue und verbesserte Security-Funktionen. Einen entscheidenden Anteil nimmt hierbei die neue Initiative „Secure by Default“ ein, die es dem Datenbankadministrator erleichtern soll, eine sicherere Datenbank out-of-the-box zu installieren. Mit der Initiative „Secure by Default“ starten wir eine neue Artikelreihe, in der wir unter anderem Maßnahmen für eine erhöhte Passwortsicherheit in 11g vorstellen und später auf Veränderungen im Security-Bereich eingehen, die Release 2 mit sich bringt.

Überblick

Unter „Secure by Default“ versteht Oracle eine ganze Reihe von out-of-the-box-Maßnahmen, die die Oracle 11g Datenbank sicherer machen sollen als eine vergleichbare 10g-Installation. Diese Maßnahmen möchten wir Ihnen in diesem Artikel vorstellen:

- Standardmäßiges Auditing
- Änderungen am Standard-Benutzerprofil
- Verbesserte Überprüfung der Passwort-Komplexität
- Delayed Failed Logins
- Fine Grained Access Control (FGAC) bei Netzwerk-Callouts aus der Datenbank

Daneben spielen der verbesserte Passwortschutz und ein erweitertes Passwortmanagement eine zentrale Rolle unter den neuen Security-Funktionen, die Oracle 11g standardmäßig mit sich bringt. Dieses Thema wird jedoch erst Bestandteil des nächsten Artikels sein.

Ob und wie die hier eingeführten Maßnahmen wirklich zu einer out-of-the-box sichereren Datenbank führen und mit welchem Aufwand dies für den Datenbankadministrator zu erreichen ist, wird im Folgenden näher beleuchtet.

„Enhanced Default Security Settings“ im DBCA 11g

Gleich zu Beginn des Arbeitens mit Oracle 11g R1 kommt der Anwender mit der neuen Initiative

„Secure by Default“ in Berührung. Denn schon der DBCA fragt sowohl beim Anlegen als auch beim nachträglichen Ändern der Datenbank, ob die „Enhanced default security settings“ eingeschaltet werden sollen:

Oracle empfiehlt, diese Frage mit dem ersten Auswahlpunkt zu beantworten (siehe Abbildung 1). Dies hat Anpassungen des Standard-Benutzerprofils sowie ein standardmäßiges Auditing bestimmter Operationen zur Folge, was in den folgenden zwei Absätzen näher erläutert wird.

Übrigens bietet Oracle mit 11g Release 2 diese Wahl nicht mehr an. Die „Enhanced Default Security Settings“ sind hier somit gesetzt und können im DBCA nicht mehr abgewählt werden.

Standardmäßiges Auditing

Über das Standard-Auditing lassen sich Aktionen von Benutzern in den drei Bereichen Objekte, Kommandos und Privilegien protokollieren.

Werden die „Enhanced Default Security Settings“ unter Oracle 11g eingeschaltet, so ändert sich der Default dieses Standard-Auditing folgendermaßen:

Der Parameter AUDIT_TRAIL steht dann auf DB, statt wie bisher auf NONE.

Viele Aktionen werden nun zwangsweise per Standard-Auditing (AUDIT...BY ACCESS)

in die Audit-Tabelle SYS.AUD\$ protokolliert. Hierzu zählen Aktionen, die unter Nutzung der in Abbildung 2 dargestellten Privilegien durchgeführt werden.

Was aus Gründen der Sicherheit eine gute Maßnahme darstellt, kann sich allerdings schnell zum Problem entwickeln, da es keinen Automatismus für das Löschen der Audit-Daten in der Audit-Tabelle (AUD\$) gibt. Die Tabelle wird schnell zunehmend größer, was je nach Konfiguration bis zum Stillstand der Instanz führen kann. Die Tabelle sollte also in einen dedizierten Tablespace gelegt und die entstandenen Audit-Daten regelmäßig verarbeitet werden.

Zu diesem Zweck lohnt es sich, einen Blick auf das mit Patchset 11.1.0.7 eingeführte Paket DBMS_AUDIT_MGMT zu werfen; es bietet hilfreiche Prozeduren zur Administration (Verschieben, Bereinigen, etc.) der Audit-Tabellen.

Es empfiehlt sich außerdem, die obigen Audit-Einstellungen zu überprüfen und ggf. an die eigenen Bedürfnisse anzupassen; einige sind gewiss zu viel des Guten (z. B. CREATE SESSION... WHENEVER SUCCESSFUL), andere könnten sicherlich noch hinzugefügt werden.

Änderungen am Standard-Benutzerprofil

Durch das Einschalten der „Enhanced Default Security Settings“ werden nicht nur die Einstellungen für ein standardmäßiges Auditing aktiviert, sondern auch Änderungen am Standard-Benutzerprofil vorgenommen (siehe Abbildung 3).

Drei der Profileinstellungen, PASSWORD_LOCK_TIME, PASSWORD_GRACE_TIME und PASSWORD_LIFE_TIME, werden durch die automatische Anpassung nun restriktiver behandelt. So bedeutet die PASSWORD_LIFE_TIME von 180, dass sämtliche Benutzer nach 180 Tagen ihr Passwort ändern müssen!

Was aus Security-Sicht sicherlich der richtige Ansatz ist, führt jedoch bei auf die Datenbank zugreifenden Programmen (Skripte, Batch, Application Server, etc.), die ihre Kennwörter fest verdrahtet haben und diese nicht selbstständig ändern können, zum Absturz nach 180 Tagen!

Um dies zu verhindern, empfiehlt es sich, vor Ablauf der 180 Tage ein spezielles Profil anzulegen, in dem PASSWORD_LIFE_TIME auf

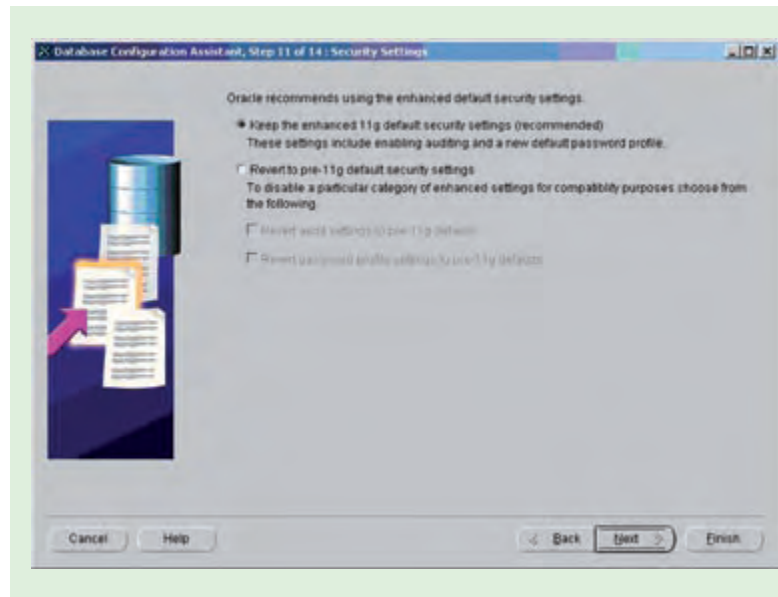


Abb. 1: Abfrage im DBCA, ob die „Enhanced default security settings“ eingeschaltet werden sollen.

ALTER ANY PROCEDURE	CREATE ANY JOB	DROP ANY TABLE
ALTER ANY TABLE	CREATE ANY LIBRARY	DROP PROFILE
ALTER DATABASE	CREATE ANY PROCEDURE	DROP USER
ALTER PROFILE	CREATE ANY TABLE	EXEMPT ACCESS POLICY
AUDIT ROLE BY ACCESS	CREATE EXTERNAL JOB	GRANT ANY OBJECT PRIVILEGE
ALTER SYSTEM	CREATE PUBLIC DATABASE LINK	GRANT ANY PRIVILEGE
ALTER USER	CREATE SESSION	GRANT ANY ROLE
ALTER SYSTEM	CREATE USER	
AUDIT SYSTEM BY ACCESS	DROP ANY PROCEDURE	

Abb. 2: Vordefinierte Einstellungen des Standard-Auditing unter Oracle 11g.

UNLIMITED gesetzt wird, und das Profil nur den Benutzern zuzuteilen, deren Passwort aus genannten Gründen nie ablaufen soll.

„Enhanced Default Security Settings“ ein-/ausschalten

Die „Enhanced Default Security Settings“ werden über die Ausführung des Skriptes \$ORACLE_HOME/rdbms/admin/secconf.sql eingeschaltet. Dieses Skript wird automatisch bei neu unter 11g erstellten Datenbanken – unabhängig davon, ob sie per Skript (!) oder DBCA erstellt wurden – ausgeführt.

Lediglich der Gang über den DBCA ermöglicht es, diese Einstellungen über den ent-

```
SQL> select RESOURCE_NAME, LIMIT
       from DBA_PROFILES
       where PROFILE = 'DEFAULT'
         and RESOURCE_TYPE = 'PASSWORD';
```

RESOURCE_NAME	LIMIT
FAILED_LOGIN_ATTEMPTS	10
PASSWORD_LIFE_TIME	180
PASSWORD_REUSE_TIME	UNLIMITED
PASSWORD_REUSE_MAX	UNLIMITED
PASSWORD_VERIFY_FUNCTION	NULL
PASSWORD_LOCK_TIME	1
PASSWORD_GRACE_TIME	7

Abb. 3: Änderungen am Standard-Benutzerprofil beim Einschalten der „Enhanced Default Security Settings“.

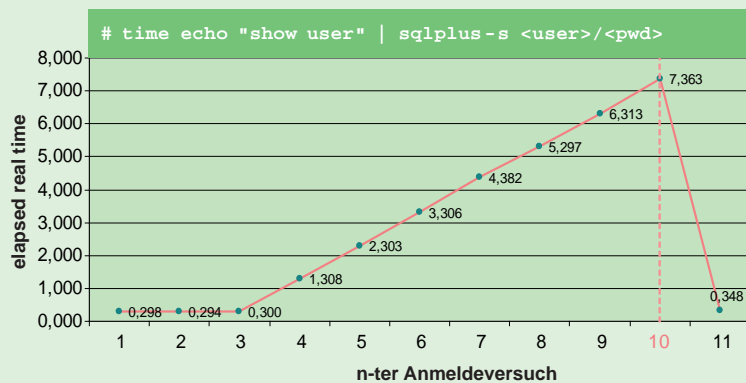


Abb. 4: Delayed Failed Login.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.CREATE_ACL (
    acl      => 'test_acl.xml',
    description => 'Network permission for UTL_INADDR for user TEST_ACL',
    principal => 'TEST_ACL'
    is_grant  => TRUE,
    privilege => 'resolve');    -- 'connect | resolve'
END;
/

BEGIN
  DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL (
    acl      => 'test_acl.xml',
    host     => 'SLES10.ordix.de');
END;
/

commit;
```

Abb. 5: ACL anlegen und einem Netzwerk Service zuweisen.

sprechenden Bildschirm (siehe Abbildung 1) einzeln oder in Gesamtheit auszuschalten. Hierfür ruft der DBCA die beiden Skripte `undoaud.sql` und `undopwd.sql` im Verzeichnis `$ORACLE_HOME/rdbms/admin` auf,

um die Änderungen am Standard Auditing und Profil nach dem Anlegen der Datenbank wieder rückgängig zu machen.

Bei auf 11g migrierten Datenbanken werden die „Enhanced Default Security Settings“ grundsätzlich nicht eingeschaltet.

„Secure by Default“ – Verbesserte Passwort-Verifizierungsfunktion

Oracle liefert mit dem erweiterten Passwort-Management, das bereits in Oracle 8i eingeführt wurde, folgendes Skript aus:

```
$ORACLE_HOME/rdbms/admin/utlpwdmg.sql
```

Über die Ausführung dieses Skriptes wird in der Datenbank eine PL/SQL-Funktion angelegt, die neu eingegebene oder geänderte Benutzerkennwörter auf ausreichende Komplexität überprüft.

In der Oracle Version 11g wurde das Skript `utlpwdmg.sql` um die zusätzliche Funktion `VERIFY_FUNCTION_11G` erweitert, in der die Anforderungen an die Komplexität des Passwortes signifikant erhöht wurden:

- Minimale Passwortlänge von 8 Stellen (bisher: 4 Stellen)
- Passwort ungleich Username(1-100) (neu)
- Passwort ungleich Username rückwärts geschrieben (neu)
- Passwort ungleich Datenbankname, Datenbankname(1-100) (neu)
- „Einfache“ Passwörter werden geprüft (z. B. `welcome1`, `database1`, `user1234`, etc.)
- Passwort ungleich oracle(1-100) (neu)
- Passwort muss mindestens 1 Buchstabe und mindestens 1 Zahl beinhalten (bisher: 1 Buchstabe, 1 Zahl und 1 Sonderzeichen)

Da die neue Verifizierungsfunktion auch in 11g standardmäßig nicht eingerichtet ist, muss das Skript `utlpwdmg.sql` weiterhin explizit ausgeführt werden.

Die Ausführung des Skriptes hat neben der Anlage der neuen Funktion aber auch Änderungen am Standard Benutzerprofil zur Folge: Der Profil-Parameter `PASSWORD_VERIFY_FUNCTION` wird auf `VERIFY_FUNCTION_11G` geändert. Darüber hinaus werden die auf die Ressource `PASSWORD` bezogenen Parameter gemäß den „Enhanced Default Security“-Einstellungen neu gesetzt.

„Secure by Default“ – Verzögertes Login bei falscher Passworteingabe

Der Versuch, sich mehrfach mit falschem Kennwort an einen existierenden Benutzer in der Datenbank anzumelden, löst ab Oracle 11g eine verzögerte Bearbeitung weiterer Login-Versuche an diesen Benutzer aus. Diese Maßnahme, mit deren Hilfe Brute-Force-Attacken erschwert werden sollen, gehört mit zum Standard in 11g und wird nach 3 Fehlversuchen mit einer maximalen Verzögerungszeit von jeweils 10 Sekunden wirksam.

Die verzögerte Login-Bearbeitung greift auch dann, wenn zudem versucht wird, die Verbindung von einer anderen IP-Adresse oder Host aus aufzunehmen.

FAILED_LOGIN_ATTEMPTS vs. SEC_MAX_FAILED_LOGIN_ATTEMPTS

Die soeben beschriebene Verzögerung bei fehlerhaften Anmeldeversuchen setzt sich bis zur Sperrung des betroffenen Accounts in der Datenbank fort. Die Anzahl der hier möglichen Versuche ist auch in 11g weiterhin durch die Angabe FAILED_LOGIN_ATTEMPTS im Passwortprofil des Benutzers begrenzt (siehe Abbildung 4).

Im Gegensatz dazu bezieht sich der in Oracle 11g eingeführte Initialisierungsparameter SEC_MAX_FAILED_LOGIN_ATTEMPTS nicht auf einen bestehenden Account in der Datenbank, sondern auf jeglichen fehlerhaften Anmeldeversuch mit unbekanntem Benutzer und Passwort. Bleiben 10 dieser wahlfreien Angriffsversuche (= Intruders) ohne Erfolg, wird die Verbindung abgebrochen; eine Funktionalität, die allerdings nur im Kontext einer OCI-Applikation zum Tragen kommt.

Access Control Listen (ACLs) – Kontrolle der Netzwerkzugriffe aus der Datenbank

In der Oracle Datenbank existieren bereits seit längerer Zeit eine Reihe von PL/SQL-Packages, mit deren Hilfe Informationen aus der Datenbank heraus an beliebige, im Netzwerk erreichbare Server geschickt werden können. Zu diesen „Netzwerk“-Paketen zählen:

- UTL_TCP
- UTL_SMTP
- UTL_MAIL
- UTL_HTTP
- UTL_INADDR

Glossar

DBCA	Dababase Configuration Assistant. Grafischer Assistent zur Erstellung und Konfiguration von Oracle Datenbanken und anderen Oracle Komponenten.
Datenbankprofil	Hierüber wird die Nutzung von Ressourcen eingeschränkt sowie verschiedene Optionen zur Verwaltung von Passwörtern gesetzt. In Oracle ist jedem Benutzer genau ein Profil zugeordnet.
Standardprofil	Profil mit dem Namen DEFAULT, das jedem Benutzer zugeordnet wird, wenn beim CREATE USER kein spezielles Profil angegeben wurde.
OCI	Oracle Call Interface. Client-seitige Low-Level-Programmierschnittstelle zum Zugriff auf die Oracle Datenbank, bestehend aus einer Sammlung von in C geschriebenen Software APIs.
DB-Link	Ein Datenbank-Link ermöglicht den Zugriff aus einer Datenbank heraus über Oracle Net auf eine beliebige, andere Datenbank.
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.

Links

- ▶ [1] Buch „HOWTO Secure and Audit Oracle 10g and 11g“, Autor: Ron Ben Natan, ISBN-10: 1420084127, ISBN-13: 978-1420084122
- ▶ [2] Buch „Oracle Database 11g New Features für DBAs and Developers“, Autor: Sam R. Alapati, Charles Kim, ISBN-10: 1590599101, ISBN-13: 978-1590599105
- ▶ [3] Präsentation „Oracle 11g Security“ von Pete Finnigan; <http://www.petefinnigan.com>
- ▶ [4] ORDIX News Artikel „Oracle Database 11g Release 1: Advanced Security“ http://www.ordix.de/ORDIXNews/2_2008/Datenbanken/kr_oracle11g_t3.html
- ▶ [5] Seminarempfehlung „Oracle Security“ <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/837/index.html>

Diese Pakete sind bis Oracle 10g durch das EXECUTE-Privileg, das standardmäßig an PUBLIC(!) vergeben ist, zur Ausführung freigegeben. Diese uneingeschränkte und allgemeingültige Zugriffsmöglichkeit auf das gesamte Netzwerk erscheint praktisch, stellt aber ein erhebliches Sicherheitsrisiko dar.

Dem begegnet Oracle 11g mit einem neu eingeführten Sicherheitskonzept: Netzwerkzugriffe, die durch PL/SQL-Pakete erfolgen, müssen vom DBA sowohl für die einzelnen Ziele als auch für die einzelnen Benutzer bzw. Rollen separat freigegeben werden.

Diese Freigabe erfolgt über so genannte Access Control Listen (ACLs). Programmtechnisch werden die ACLs mit dem neuen Package DBMS_NETWORK_ACL_ADMIN erzeugt und verwaltet (siehe Abbildung 5).

Aber Achtung: ACLs werden als XML-Dokumente unter Verwendung der XML DB-Komponente abgelegt. Damit ein Arbeiten mit diesen „Netzwerk“-Paketen also auch weiterhin funktioniert, muss die XML DB in der 11g Datenbank installiert sein.

Fazit

Festzuhalten bleibt, dass es mit der neuen Oracle Initiative „Secure by Default“ wirklich leichter wird, eine sicherere Datenbank out-of-the-box zu installieren. Die Maßnahmen sind aus Security-Sicht sinnvoll und holen mit einigen wichtigen Industriestandards in punkto Sicherheit auf.

Es bleiben jedoch Aufwände für den Administrator bestehen. Sie liegen hauptsächlich in der Berücksichtigung der Auswirkungen sowie der Anpassung an die eigenen Bedürfnisse dieser out-of-the-box-Maßnahmen. Nur das Zusammenspiel der von Oracle gelieferten Default-Sicherheitsmaßnahmen mit den manuellen Konfigurationen des DBA führen zu einer wirklich sicheren Datenbank.

Im nächsten Artikel stellen wir Ihnen u. a. die Änderungen im Bereich des Passwortmanagements vor.

Kathleen Hock (info@ordix.de).



Larry Ratlos

Java Varargs?

Larry hat den Java-Aufbaukurs gemeistert (siehe ORDIX News 4/2009). Nun will sein Chef wissen, ob er das Erlernte auch anwenden kann. Im Java-Kurs lernte Larry die Varargs kennen. In seinem neuen Projekt kann er dieses Feature gut gebrauchen.

Neues Projekt - neue Probleme

Da er sich an die Funktionsweise von Varargs nicht mehr hundertprozentig erinnern kann, schiebt er sich zunächst ein kleines Testpro-

gramm (siehe Abbildung 1). Anschließend ruft er seine Methode mehrfach mit verschiedenen Werten auf (siehe Abbildung 2). Doch einige Ausgaben auf der Konsole irritieren Larry. Wer kann ihm das Verhalten erklären?

```
public void varargs(Object... values) {
    for (Object o : values) {
        System.out.print(o);
    }
    System.out.print("\n");
}
```

Abb. 1: Larry's Testprogramm.

```
1) varargs("ORDIX");
2) varargs(1);
3) varargs(1, 2, 3, 4);
4) Integer[] i = { 1, 2, 3, 4, 5 };
   varargs(i);
3) varargs("ORDIX ", 4711);
4) String[] s1 = { "Larry ", "ist ", "ratlos." };
   varargs(s1);
5) String[] s2 = { "Larry ", "ist ", "ratlos " };
   varargs(s2, "und ", "verwundert.");
```

Abb. 2: Mehrere Methodenaufrufe mit unterschiedlichen Werten.

Können Sie Larry helfen?

Schicken Sie Larry jeweils die Ausgaben, die die verschiedenen Methodenaufrufe liefern und erklären Sie ihm seinen Denkfehler der Ausgabe 5. Auf Ihre Lösungen freut sich Larry bis zum 14. April 2010 an kniffel@ordix.de.

Lösung der Aufgabe aus 4/2009

Viele Leser konnten Larry helfen, so dass er den Java-Aufbaukurs besuchen durfte. Er bedankte sich mit einem kleinen Weihnachtspaket bei Herrn Mueller von der Baumann GmbH + Co. KG, der ihm als erster die richtige Lösung sendete.

Die richtige Antwort lautete: a, d und f.

Neue Reihe: Perl und DB2 im Verbund (Teil I)

Perlen (nicht nur) für DB2-DBAs

Maskottchen sind beliebt. Bei Perl ist es seit langem das Kamel. IBM setzt seit DB2 Version 9.1 auf Schlangen (Viper, Cobra). Tiere, die sich eher aus dem Weg gehen, möchte man glauben. Das sie sich mit ein wenig gutem Zureden zumindest in der IT gut vertragen, soll dieser erste Artikel unserer neuen Reihe aufzeigen. Auch Perl-Entwickler, die mit anderen Datenbanken arbeiten, sind herzlich zum Lesen eingeladen.

Diese Reihe richtet sich an DB2-Administratoren, die die Möglichkeiten von Perl für ihre tägliche Arbeit kennenlernen möchten und an Perl-Entwickler, die erstmalig mit einer DB2-Datenbank arbeiten möchten.

Warum Perl?

„Folgende 12 Abfragen hätte ich gerne jeden Tag auf dem Linux-Server ausgeführt und als Excel-Arbeitsmappe auf dem FTP-Server des Kunden abgelegt“. So oder ähnlich lauten Aufträge unserer Kunden. Nun, den ersten Teil der Aufgabe könnte man auch mit einem Shell-Skript erledigen. Aber wie geht es weiter, wenn die Anforderungen an den Output ständig variieren?

Nach mittlerweile gut 20 Jahren Entwicklungszeit hat sich Perl in vielen Bereichen etabliert und vor allem im Unix-Umfeld bei einer wachsenden Anzahl von Administratoren die Shell-Programmierung abgelöst. Die große Anzahl frei verfügbarer Bibliotheken deckt viele Spezialfälle ab, die mit einem Shell-Skript entweder gar nicht oder nur sehr mühsam abzubilden wären. Zudem kann man sich bei aktuellen Unix-Derivaten (bei Linux sowieso) i. d. R. darauf verlassen, dass zumindest eine Perl-Installation mit den Standard-Bibliotheken vorhanden ist. Da sich viele Aufgaben innerhalb der Programmierung ständig wiederholen, ist die Möglichkeit der Wiederverwendbarkeit von Code sicher für viele praktisch. Wenn man ein wenig Acht gibt, bleibt der Code auch ohne allzu ausschweifende Kommentierungen wartbar.

Die Qual der Wahl

Die Suche nach „DB2“ im CPAN liefert eine Vielzahl von entsprechenden Perl-Modulen [1]. Der Entwicklungsstand dieser Module ist jedoch recht unterschiedlich. So wurden z. B. die DBIx-Module bereits seit 2006 nicht mehr aktualisiert. Die verbreitetste Technologie dürfte die Verwendung der DBI/DBD-Schnittstelle sein, auf die wir im Folgenden eingehen.

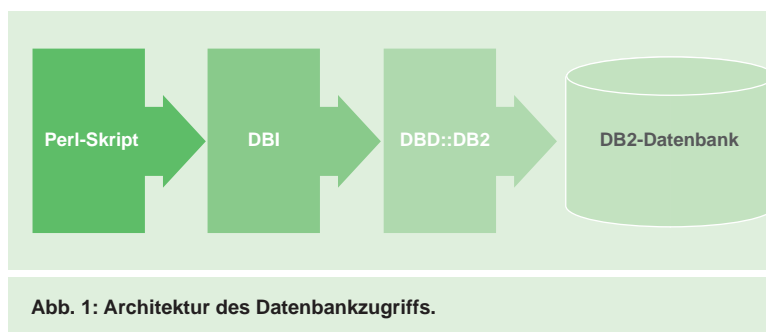


Abb. 1: Architektur des Datenbankzugriffs.

DBI und DBD

Zuständig für die Anbindung beliebiger Datenbanken unter Perl ist das Database Independent Interface (DBI). Für die Anbindung an DB2 ist weiterhin ein spezifischer Datenbanktreiber (DBD) notwendig (siehe Abbildung 1). Beide sind nicht Bestandteil der Perl-Standardbibliothek und müssen somit nachinstalliert werden.

Diese Kombination liefert uns eine SQL-Schnittstelle. Da mit jeder Version von DB2 mehr Konfigurationsparameter über die SYSIBMADM-Views zugreifbar werden, ist dies ein guter Einstiegspunkt. Das DB2::Admin-Modul verspricht dagegen Zugriff auf Nicht-SQL-Aktionen (Event-Snapshots, Katalog-Verwaltung usw.), leidet aber an mehreren Einschränkungen. Die Installation ist an viele Vorbedingungen geknüpft und die vollständige Funktionsweise wurde bisher lediglich für RedHat-basierte Systeme bestätigt. Zudem ist eine ständig größer werdende Anzahl von administrativen Aufgaben mit den SYSIBMADM-Views und einfachem SQL übersichtlich handhabbar.

```

source $DB2INSTANCE/sqlllib/db2profile
export DB2_HOME=/opt/ibm/db2/v9.7/
gunzip DBD-DB2-1.78.tar.gz
tar -xvf DBD-DB2-1.78.tar
cd DBD-DB2-1.78
perl Makefile.PL
make
make install

```

Abb. 2: Beispielhafte Installation des DBD::DB2-Treibers.

```

#!/usr/bin/perl -w
use strict;      # zwingt uns u. a. zur Deklaration unserer Variablen
use DBI;
use DBD::DB2::Constants;
use DBD::DB2;

my $database_name = "dbi:DB2:ordix";
my $database_user = "testdb2";
my $password      = "1234";
my $dbh;          # Diese Variable wird das Objekt mit dem Database-Handle enthalten

$dbh = DBI->connect(      # Verbindung aufnehmen
    $database_name,
    $database_user,
    $password,
    {
        db2_txn_isolation => SQL_TXN_READ_UNCOMMITTED,
                                # aus DBD::DB2::Constants
        db2_info_programname => "perl-dbi-test"
    }
) or die "Datenbankverbindung gescheitert:", $dbh->errstr;

$dbh->do(
    "CREATE TABLE ORDIX_TEST (id bigint not null, seminar varchar(200), dozent varchar(30))"
);
# DDL-Operationen mit "do"

my $insertstring = "INSERT INTO ORDIX_TEST VALUES(?,?,?)";
my $sth          = $dbh->prepare($insertstring);
$sth->execute( 1, 'Perl Programmierung Grundlagen', 'Norbert Munkel' );
$sth->execute(
    3,
    'IBM DB2 für Linux/Unix/Windows Administration',
    'Michael Spoden'
);

my $selectstring = "SELECT seminar, dozent from ordix_test WHERE dozent LIKE ?";
$sth = $dbh->prepare($selectstring);
$sth->execute("%Munkel%");

while ( my @row = $sth->fetchrow_array ) {
    print join( ' ', @row );
    print "\n";
}

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

```

Abb. 3: Ein einfaches DBI/DBD-Skript.

Wir werden uns in dieser Reihe vornehmlich mit dem „klassischen“ Zugriff per DBI/DBD beschäftigen. Da der Großteil der verwendeten Methoden direkt aus dem DBI-Modul stammt, wird erfahrenen Perl-Entwicklern

einiges bekannt vorkommen. Bitte beachten Sie in diesem Fall die spezifischen Connect-Parameter. Ihr DBA wird es Ihnen danken!

Beschaffung und Installation

Bei den meisten Linux-Distributionen ist zumindest DBI als Paket installierbar. Dieser Weg wird auch empfohlen, da benötigte Abhängigkeiten automatisch aufgelöst werden. Spätestens beim DB2-Treiber ist aber Handarbeit angesagt. Zwingende Vorbedingung hierfür ist das Vorhandensein eines C-Compilers (z. B. gcc) sowie mindestens eine DB2-Client-Installation. Für die Verbindung zur Datenbank benötigen Sie weiterhin zwingend einen lokalen Katalogeintrag.

Die Installation erfolgt sinnvollerweise als User `root`. Weitere Vorbedingung ist das Setzen des Umgebungsparameters `DB2_HOME` auf das IBM-Installationsverzeichnis. Abbildung 2 zeigt eine beispielhafte Installation des DBD::DB2-Treibers.

Der erste Kontakt

Der User, unter dem das Skript laufen soll, sollte die `db2profile`-Datei der entsprechenden Instanz laden, damit die Umgebung passt. Kurz: Der User sollte den `db2`-Client auf dem Rechner verwenden können. Für die administrativen Aufgaben sollte der User auf der Datenbankseite mindestens `SYMON`- oder `SYSCTL`-Berechtigung erhalten. In der Regel wird man ohnehin den Instanzen-User verwenden, der diese Berechtigung aufweist. Der Verbindungsaufbau ist in Abbildung 3 dargestellt.

Für den Einstieg gehen wir davon aus, dass die Datenbank lokal katalogisiert ist. Eine Verbindung zu einer nicht-katalogisierten Datenbank ist unter Angabe des Host-Namens und der Port-Nummer ebenfalls möglich, aber für unsere Reihe an dieser Stelle eher unnötig kompliziert.

Attribute für Datenbank-Connections

Jeder Connect-String kann einen zusätzlichen Parameter enthalten. Hier kann eine (anonyme) Referenz auf einen Attribut-Hash hinterlegt werden.

Mögliche Attribute sind in der DBI als auch der DBD::DB2-Dokumentation aufgeführt. An

dieser Stelle gehen wir auf zwei wesentliche Attribute beispielhaft ein.

- `db2_txn_isolation`
- `db2_info_programname`

Mit `db2_txn_isolation` kann der Isolation Level des Skripts geändert werden. Gerade für Admin-Skripte bietet sich hier der Wert `SQL_TXN_READ_UNCOMMITTED` an, da z. B. Systemtabellen oft gesperrt sind.

Mit `db2_info_programname` kann wiederum ein Applikationsname vergeben werden. Leider stellen sich Perl-Programme in der Ausgabe von `list applications` ansonsten lediglich mit dem Schlüsselwort `perl` dar, was bei mehreren parallel laufenden Skripten ein wenig unübersichtlich sein kann.

Mit Datenbank-Connections arbeiten

Um den „Hello World“-Charakter dieses ersten Teils nicht zu sprengen, werden wir uns auf ein paar grundlegende Operationen beschränken. Wir werden eine Tabelle erzeugen, einige Werte eingeben und diese dann auslesen. Somit haben wir sowohl DDL- als auch DML-Operationen. Für die DDL-Operationen verwenden wir die Methode `do`. Für INSERT, UPDATE und SELECT verwenden wir ein mehrstufiges Verfahren. Die Vorgehensweise ist - unabhängig von den verschiedenen speziellen Methoden - im Wesentlichen wie folgt:

1. `connect`
2. `prepare` (Statement wird kompiliert)
3. `execute` (Statement wird ausgeführt, Parameter-Marker werden mit Werten belegt)
4. `fetch` (hier gibt es verschiedene Methoden, die in der DBI-Dokumentation beschrieben sind. Wir verwenden im Beispiel die Methode `fetchrow_array`, welche jede Zeile des Abfrageergebnisses in ein Array schreibt.
5. `disconnect`

Parameter-Marker

Parameter-Marker werden durch ein Fragezeichen dargestellt. Man könnte die VALUES- und WHERE-Klauseln auch direkt mit Werten belegen. Hierdurch erzeugt man allerdings einen unnötigen Overhead. Bei Verwendung der Parameter-Marker wird das Statement nur ein einziges Mal kompiliert.

Glossar

DBI	Database Independent Interface: Die universelle Datenbankschnittstelle in Perl
DBD	Database Driver: Datenbankspezifischer Treiber für DBI
CPAN	Comprehensive Perl Archive Network: Online-Repository für Perl-Module, Anwendungen und Dokumentation

Links

- ▶ [1] DB2-Module im CPAN:
<http://search.cpan.org/search?query=db2&mode=all>
- ▶ [2] Perl-Programming with DB2 Universal Database:
<http://bit.ly/7CkWvs>
- ▶ [3] DBI-Dokumentation:
<http://bit.ly/PxO87>
- ▶ [4] DBD::DB2-Dokumentation:
<http://bit.ly/aVlCpy>
- ▶ [5] Seminarempfehlungen:
Perl Programmierung Aufbau
<http://training.ordix.de/siteengine/action/load/kategorie/Programmierung/nr/238/index.html>

IBM DB2 für Linux/Unix/Windows Administration
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/279/index.html>

IBM DB2 für Linux/Unix/Windows Monitoring und Tuning
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/644/index.html>

Für Abfragen, bei denen man sicher davon ausgehen kann, dass nur eine Zeile zurückgeliefert wird, gibt es z. B. die Methode `selectall_array`. Hier werden die Phasen `prepare/execute/fetch` in einem Statement zusammengefasst und die erste Zeile des Ergebnisses einem Array zugewiesen. Ein komplettes, funktionstüchtiges Skript zeigt Abbildung 3.

Fazit

Der erste Schritt ist getan. Im nächsten Teil der Reihe zeigen wir, wie man mit `SQL::Library` Ordnung in seine Abfragen bringt und entwickeln eine aussagefähigere Alternative zu `LIST APPLICATIONS`.

Norbert Munkel (info@ordix.de).

Swing in Form bringen

Dieser Artikel ist für Java-Entwickler interessant, die Benutzeroberflächen mit Swing entwickeln.

Die Benutzeroberfläche ist das Aushängeschild einer Anwendung. Sie muss funktionalen und ästhetischen Ansprüchen genügen und ist Gegenstand langer Diskussionen und häufiger Änderungen. Der Java-Welt steht Swing als Werkzeug zur Verfügung, um anspruchsvolle Benutzer zufrieden zu stellen. Das kostenfreie Paket JGoodies Forms [1] kann dabei helfen, Dialoge gefälliger und einfacher zu gestalten. Dieser Artikel vermittelt einen ersten Eindruck, was von dieser Bibliothek erwartet werden kann.

JGoodies Forms

Aufgabe dieses Pakets ist es, die Dialoggestaltung von Swing-Anwendungen zu unterstützen. Eine wesentliche Komponente dient der Vereinfachung des Layout-Managements. Ziel dieses Pakets ist es, exzellente Ergebnisse bei kurzen Entwicklungszeiten zu erreichen.

Dabei ist JGoodies Forms [1] mit der BSD-Lizenz geschützt, Lizenzkosten fallen nicht an.

Layout-Management

Der Benutzer erwartet von Dialogen:

- Berücksichtigung unterschiedlicher Fenstergrößen
- Einheitlichkeit der Gestaltung und der Benutzung
- Übersichtliche Anordnung

Die Standard-Layout-Manager von Swing sind Spezialisten zur Lösung der ersten Anforderung. Dialogelemente werden in Abhängigkeit von der Fenstergröße dimensioniert und positioniert. JGoodies-Forms unterstützt darüber hinaus auch die übrigen Anforderungen.

Allgemein wird ein Dialog als übersichtlich empfunden, wenn Abstände zum Fensterrand eingehalten werden, gleiche Abstände zwischen den Elementen vorliegen und Dialogelemente einheitlich ausgerichtet sind.

Abbildung 1 zeigt einen entsprechenden Dialog, der im Folgenden als Beispiel dient.

Am Anfang war die Tabelle

Grundlage für die Gestaltung des Dialogs ist eine Tabelle (siehe rote Linien in Abbildung 2). Dialogelemente werden in die Zellen dieser Tabelle positioniert. Dieses sorgt für klare Linien.

Der Quellcode zur Gestaltung des Dialogs aus Abbildung 1 ist der Abbildung 3 zu entnehmen.

Die Tabelle wird im Konstruktor-Aufruf der Klasse `FormLayout` definiert: Das erste Argument legt den Aufbau der Spalten fest, das zweite Argument die Zeilen.

... dann die Zellengröße ...

Anhand der Argumente des `FormLayout`-Konstruktors lässt sich die Struktur der Tabelle bereits erahnen. Jede Beschreibung einer Zeile oder Spalte ist durch ein Komma von seinem Nachfolger getrennt. Die Beschreibung selbst kann aus mehreren Elementen bestehen, die durch Doppelpunkte voneinander getrennt sind.

Verpflichtend ist dabei nur die Angabe der Zeilen- bzw. Spaltengröße. Diese kann durch absolute Angaben wie Pixel, Punkt, Inch, Millimeter oder Zentimeter festgelegt werden.

Besonders wertvoll ist die Möglichkeit, die Abmessungen relativ zum Systemfont anzugeben. Dazu bietet JGoodies-Forms die

Einheit Dialog Units (dlu) an, die mit der Systemschriftgröße skaliert.

Die dritte Option zur Angabe der Größe einer Tabellenzelle nimmt die Dialogelemente selbst als Maßstab. Die Abkürzung `pref` steht für preferred size: dieser Spalte oder Zeile wird so viel Platz eingeräumt, dass die größte Komponente mit ihrer preferred size angezeigt werden kann.

Durch die Funktionen `max` und `min` lassen sich maximale oder minimale Zeilenhöhen oder Spaltenbreiten festlegen.

Der Abstand zwischen den Elementen wird durch – später leer gelassene – Zellen definiert (siehe Abbildung 2). Es ist sinnvoll, diese Abstände in der Einheit Dialog Units anzugeben.

... danach die Ausrichtung ...

Die Label der Eingabefelder sind nicht gleich groß (siehe Abbildung 1). Solche Elemente müssen in ihrer Zelle ausgerichtet werden. Dazu dient die Information vor der Größenangabe. Als Ausrichtungsoption stehen zur Wahl: `left`, `right`, `top` und `bottom`.

Diese Attribute werden der Größenangabe vorangestellt, z. B. `right:pref` ist demnach eine Spaltenbeschreibung, die dafür sorgt dass die größte Komponente der Zeile mit der preferred size dargestellt wird, die übrigen Komponenten werden rechtsbündig positioniert.

Wird keine Angabe gemacht, so sollen die Komponenten wie Eingabefelder oder Schaltflächen möglichst den gesamten Raum ihres Bereiches einnehmen. Das Verhalten wird auch durch die Ausrichtungsoption `fill` erreicht.

... und schließlich das Resizing-Verhalten

Eine Veränderung der Fenstergröße erlaubt dem Benutzer, den Platz auf seinem Bildschirm optimal für verschiedene Anwendungen aufzuteilen. Steht mehr Platz zur Verfügung, so soll dieser optimal genutzt werden.

Es kann beispielsweise ein größerer Teil einer Tabelle angezeigt werden oder aber das Freitextfeld den zur Verfügung stehenden Platz nutzen.

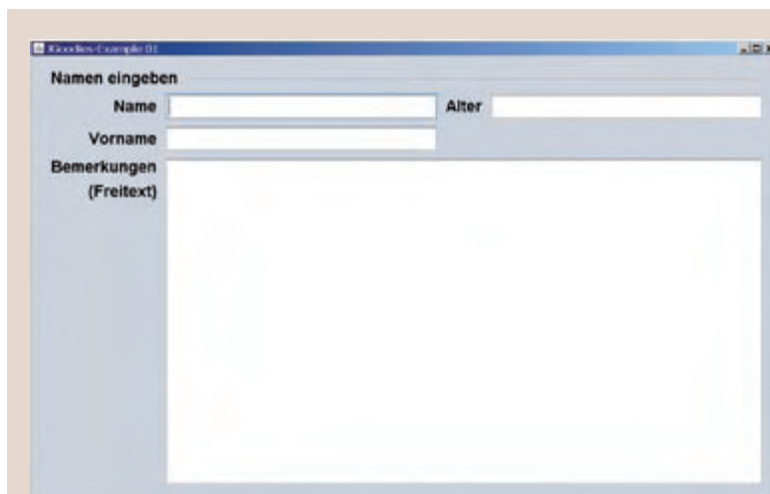


Abb. 1: Dialog zur Erfassung von Personendaten.

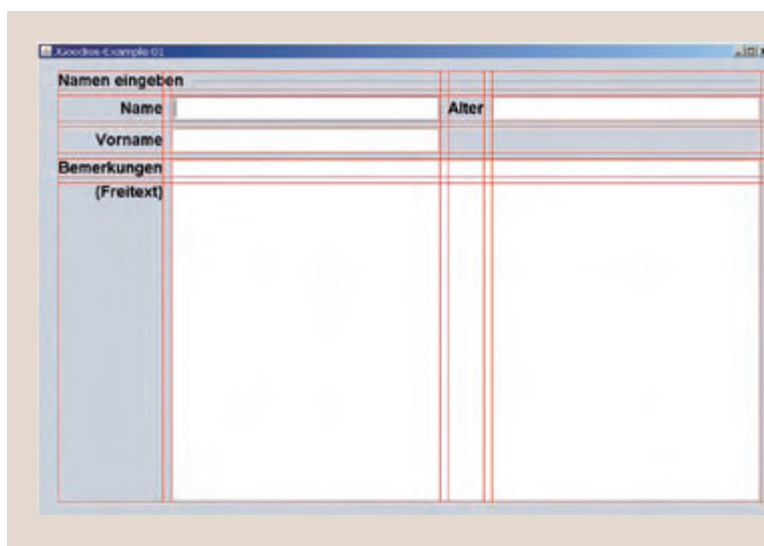


Abb. 2: Tabellen-Struktur des Dialogs (durch JGoodies-Forms dargestellt).

Standardmäßig werden Komponenten beim Resizing in ihren Abmessungen nicht verändert. Wird das Attribut `grow` gesetzt, passt sich die Spaltenbreite oder Zeilenhöhe an die Fenstergröße an. Sollen gleich mehrere Zeilen oder Spalten wachsen, so kann das Attribut `grow` mit einem Gewichtungsfaktor versehen werden.

```

private static JPanel addDialog() {
    JTextField nameField = new JTextField();
    JTextField vornameField= new JTextField();
    JTextField alterField = new JTextField();
    JTextArea bemerkungen = new JTextArea();

    FormLayout formLayout = new FormLayout(
        "right:pref"                // Spalte 1
        + ", 3dlu"                  // Spalte 2
        + ", fill:max(100dlu;pref):grow" // Spalte 3
        + ", 3dlu"                  // Spalte 4
        + ", right:pref"            // Spalte 5
        + ", 3dlu"                  // Spalte 6
        + ", max(100dlu;pref)"      // Spalte 7
        + ", 3dlu"                  // Spalte 8
        ,
        "pref"                       // Zeile 1
        + ", 3dlu"                   // Zeile 2
        + ", pref"                   // Zeile 3
        + ", 3dlu"                   // Zeile 4
        + ", pref"                   // Zeile 5
        + ", 3dlu"                   // Zeile 6
        + ", pref"                   // Zeile 7
        + ", 3dlu"                   // Zeile 8
        + ", max(50dlu;p):grow"     // Zeile 9
    );

    PanelBuilder builder = new PanelBuilder(formLayout);
    builder.setDefaultDialogBorder();

    CellConstraints cc = new CellConstraints();
    builder.addSeparator("Namen eingeben", cc.xyw(1, 1, 7));
    builder.addLabel("Name", cc.xy(1, 3));
    builder.add(nameField, cc.xyw(3, 3, 1));
    builder.addLabel("Alter", cc.xy(5, 3));
    builder.add(alterField, cc.xy(7,3));
    builder.addLabel("Vorname", cc.xy(1, 5));
    builder.add(vornameField, cc.xyw(3, 5, 1));
    builder.addLabel("Bemerkungen", cc.xy(1, 7));
    builder.addLabel("(Freitext)", cc.xy(1, 9, "right,top"));
    builder.add(bemerkungen, cc.xywh(3, 7, 5, 3));

    return builder.getPanel();
}

```

Abb. 3: Quellcode zur Erzeugung des Layouts aus Abbildung 1.

```

JButton speichernButton = new JButton("Speichern");
JButton abrechenButton = new JButton("Abbrechen");
JPanel panel = ButtonBarFactory.buildRightAlignedBar(new JButton[] {
    speichernButton, abrechenButton });

builder.add(panel, cc.xyw(3, 11, 5));

```

Abb. 4: Beispielquellcode zur Erzeugung eines Button-Panels.

Proportional zu diesem Faktor wird dann der zusätzliche Platz verteilt.

Damit sind alle Konfigurationstypen zur Beschreibung der Layouttabelle dargestellt. Durch das *Ausrichtung:Größe:Resizing-Verhalten* sind Zeilenhöhen und Spaltenbreiten der Tabelle definiert.

Den Zellen der Tabelle werden abschließend die Dialogelemente zugeordnet.

Jeder auf seinen Platz

Es gibt unterschiedliche Möglichkeiten, Dialogelemente einer Zelle zuzuweisen. Im Quellcode der Abbildung 3 ist ein explizites Verfahren gewählt, in dem die Koordinaten der Zelle angegeben werden, die für das Element bestimmt sind.

Dazu werden `CellConstraints` erstellt, die die Layout-Eigenschaften eines Dialogelements aufnehmen. Neben den Koordinaten kann auch die Ausdehnung, durch die Anzahl der Zeilen bzw. Spalten, angegeben werden. Es lassen sich hiermit Ausrichtung und Resizing-Verhalten einzelner Dialogelemente angeben.

Das Element wird zusammen mit dem `CellConstraint` dem `PanelBuilder` zugewiesen. Sind auf diese Weise alle Elemente registriert, so wird das Ergebnis-Panel mit `PanelBuilder.getPanel` zurückgegeben.

Uniformität ist gewünscht

Die Einheitlichkeit von Elementen in einem Dialog wird oftmals gewünscht. Ein prominentes Beispiel sind die Schaltflächen, die in der Regel am unteren Rand eines Dialoges zum *Bestätigen* oder *Abbrechen* positioniert sind.

Damit die Ausrichtung dieser Schaltflächen in der gesamten Applikation gleich ist, sollte eine Factory das Panel erzeugen, auf dem die Schaltflächen untergebracht werden.

JGoodies-Forms bietet zu diesem Zweck unter anderem die `ButtonBarFactory`. Abbildung 4 enthält einen Code-Schnipsel, in dem die Verwendung der Button-Bar gezeigt wird.

Die Schaltflächen *Speichern* und *Abbrechen* werden unter den Dialog gesetzt (siehe Abbildung 5).

Schlussfolgerung

Der Artikel zeigt, dass das Paket JGoodies-Forms ein nützliches Werkzeug für die Gestaltung von Swing-Dialogen ist. Die Qualität der Dialoge, die Verständlichkeit und die Wartbarkeit des Quellcodes lassen sich damit erheblich verbessern.

Testen Sie das Paket doch einfach mal für Ihre Swing-Entwicklung. Wir unterstützen Sie gern dabei.

Übrigens, das Thema Layout-Management wird Ihnen auch in unserem Seminar „Java GUI Entwicklung mit Swing“ (siehe Seminar-empfehlung unten) nahe gebracht.

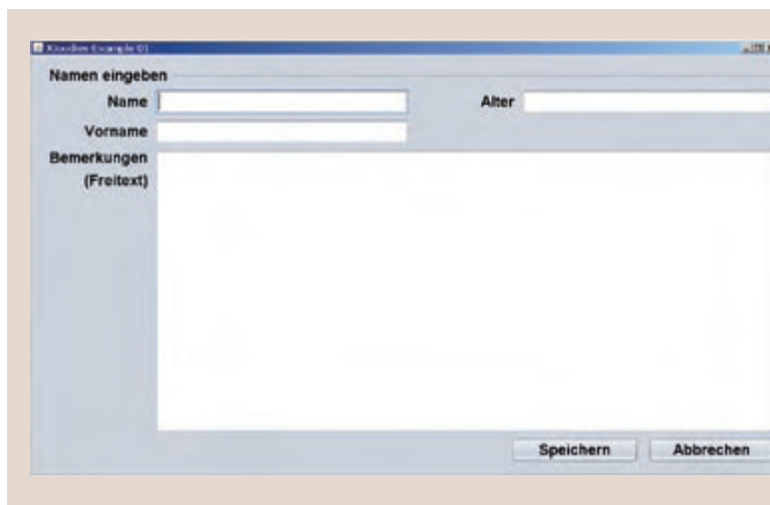


Abb. 5: Beispiel zur Erzeugung eines Button-Panels.

Link

- ▶ [1] Copyright (c) 2002-2008 JGoodies Karsten Lentzsch
<http://www.jgoodies.com>

Dr. Stefan Koch (info@ordix.de).

Seminarempfehlung: Java GUI Entwicklung mit Swing

▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Java-JEE/nr/79/index.html>

Dieses Seminar vermittelt Ihnen die Erstellung komplexer, grafischer Oberflächen mit Hilfe der Java Bibliothek Swing. Sie lernen dabei sowohl die Container, Komponenten und Layoutmanager als auch fundamentale Aspekte wie das Event-Handling kennen. Sehr viel Wert wird dabei auf den Praxisteil mit einer durchgehenden Übungsaufgabe gelegt.

Seminarinhalte

- Entstehung der Grafikbibliothek Swing und Unterschiede zu AWT
- Grundlegende Swing-Container und -Komponenten
- Einfache und komplexe Layout-Manager
- Event-Handling: Interfaces und Adapterklassen
- Vorstellung des Model-View-Controller (MVC) Entwurfsmusters und dessen Umsetzung (Observer und Observable)
- Komplexe Swing-Container und -Komponenten
- Nutzung einer Tabelle in Java
- Look and Feel
- Internationalisierung
- Übungen mit Hilfe einer durchgehenden Praxisaufgabe

Termine

28.06. - 02.07.2010 in Wiesbaden
06.09. - 10.09.2010 in Wiesbaden
29.11. - 03.12.2010 in Wiesbaden

Seminar-ID: P-JAVA_02

Dauer: 5 Tage

Preis pro Teilnehmer: 1.690,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.521,00 € (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>.

Rolling Upgrade mit Logical Data Guard

Dieser Artikel richtet sich an Entscheider und Datenbankadministratoren, die eine Möglichkeit für ein Upgrade mit minimaler Ausfallzeit suchen.

Administratoren und Systembetreuer haben den Anspruch, geplante Ausfallzeiten in Datenbanksystemen auf ein Minimum zu reduzieren. Diese Funktionalität bietet das Feature „Rolling Upgrade“, welches Oracle mit der Version 10.1.0.3 eingeführt hat. Dieser Artikel beschreibt, wie man mit Hilfe eines Logical Data Guard, der nach einem Switchover die Rolle der Primärdatenbank annimmt, ein Upgrade mit minimaler Ausfallzeit durchführen kann (hier von Oracle 11.1.0.6 nach Oracle 11.1.0.7). Außerdem wird erläutert, welche Vorüberlegungen getroffen werden müssen.

Konzept des Rolling Upgrade im Überblick

Zu Beginn des Artikels geben wir Ihnen einen groben Überblick über das Konzept eines Rolling Upgrade. Anschließend beleuchten wir die Voraussetzungen, die dafür gegeben sein müssen und gehen danach auf den detaillierten Ablauf ein.

Als Basis für ein Rolling Upgrade benötigt man zusätzlich zur Primärdatenbank eine logische Standby-Datenbank. Zum Abgleich der beiden Datenbestände werden alle Änderungen in der Primärdatenbank mit Hilfe des SQL-Apply an die Standby-Datenbank übertragen. In Abbildung 1 wird der Ablauf des Rolling Upgrade in vier Teilschritten dargestellt:

1. Aufsetzen der Logischen Standby-Datenbank
2. Anhalten des SQL-Apply und Upgrade der Standby-Datenbank
3. Reaktivierung des SQL-Apply im Mischbetrieb (11.1.0.6 der Primärdatenbank und 11.1.0.7 der Standby-Datenbank)
4. Switchover: Nachdem alle Transaktionen erfolgreich an die logische Standby-Datenbank übertragen wurden und beide Datenbanken den gleichen Stand beinhalten, folgt der Switchover. Ab jetzt sind die Rollen der Datenbanken getauscht. Die alte Primär-

datenbank besitzt die Rolle der Standby-Datenbank und die alte Standby-Datenbank agiert nun mit dem aktuellen Oracle-Release als neue Primärdatenbank.

Nach dem Switchover ist der SQL-Apply noch nicht aktiviert. Somit können wir analog zum vorherigen Upgrade die neue Standby-Datenbank (die alte Primärdatenbank mit dem alten Oracle Release) aktualisieren. Dieser Vorgang beeinflusst den Betrieb der Primärdatenbank nicht. Der Anwender nimmt somit nur die Ausfallzeit während des Switchover wahr.

Voraussetzungen und Vorüberlegungen

Um ein Rolling Upgrade erfolgreich durchführen zu können, bedarf es einiger Voraussetzungen und Vorüberlegungen.

- Der `archiveLog`-Modus auf der Primärdatenbank muss aktiviert sein.
- Die Datenbankeinstellung `Force Logging` muss aktiviert sein, damit alle DML-Operationen in den Redolog-Dateien protokolliert werden.
- Damit alle Datensätze beim SQL-Apply eindeutig identifiziert werden können, bedarf es der Datenbankeinstellung `supplemental Logging` (siehe Abbildung 2).
- Es muss untersucht werden, ob es Tabellen gibt, die keinen Primärschlüssel ha-

ben. Ein Primärschlüssel wird bei einer logischen Standby-Datenbank, anders als bei einer physikalischen Standby-Datenbank, zur Identifizierung eines Datensatzes genutzt. Besitzt eine Tabelle keinen Primärschlüssel, sollte ein solcher angelegt werden oder zumindest ein Unique-Spaltenindex mit **NOT NULL**-Wert in der Tabelle vorhanden sein, damit die Datensätze eindeutig identifiziert werden können. In der View `dba_logstdby_not_unique` kann diese Anforderung überprüft werden. Falls diese Anforderung nicht realisiert werden kann und kein Primary Key oder Unique Index für Tabellen angelegt werden kann, sollte das **Supplemental Logging** für alle Spalten aktiviert werden. Hierbei muss allerdings mit Performance-Einbußen gerechnet werden.

- Des Weiteren werden nicht alle Objekte durch das SQL-Apply in der logischen Standby-Datenbank nachgefahren. Nicht unterstützte Objekte und Datentypen werden in den Views `DBA_LOGSTDBY_UNSUPPORTED` und `DBA_LOGSTDBY_SKIP` ermittelt. Die Basistechnologie für das SQL-Apply ist der Oracle **LogMiner**. Dieser unterstützt jedoch nicht alle bei Oracle möglichen Datentypen. Diese Datentypen werden gegen Ende des Artikels im Abschnitt „Möglichkeiten und Grenzen“ aufgeführt.
- Der Data Guard Protection Mode muss entweder auf **Maximum Performance** oder **Maximum Availability** stehen. Dieses kann in der Spalte `PROTECTION_LEVEL` in der Tabelle `v$DATABASE` überprüft werden.
- Die Parameter `COMPATIBLE` der Primärdatenbank und der Standby-Datenbank muss identisch sein (in unserem Beispiel 11.1.0.6).

Nach dem groben Überblick und dem Klären der nötigen Voraussetzungen für ein Rolling Upgrade, gehen wir nun tiefer ins Detail und stellen Ihnen die einzelnen Schritte ausführlich vor.

Implementierung des logischen Data Guard

Wie anfangs bereits erwähnt, benötigt man eine logische Standby-Datenbank, um überhaupt ein Rolling Upgrade durchführen zu können. Der erste Schritt zum Erzeugen einer logischen Standby-Datenbank ist das Anlegen einer physikalischen Standby-Datenbank. Aus der physikalischen Standby-Datenbank

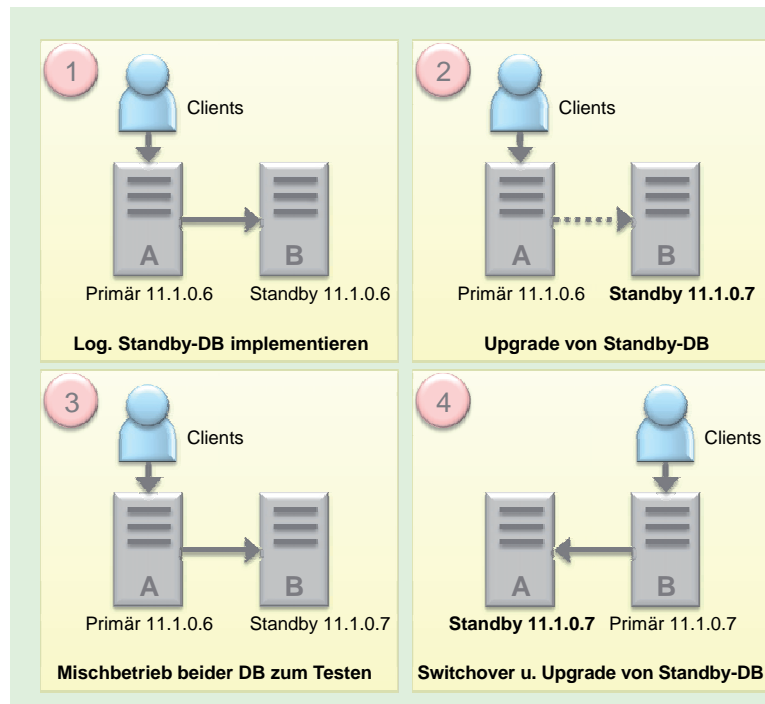


Abb. 1: Ablauf des Rolling Upgrade.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE KEY) COLUMNS;
```

Abb. 2: Supplemental Logging einschalten.

```
ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

Abb. 3: Befehl zum Stoppen des SQL-Apply.

wird dann im weiteren Verlauf eine logische Standby-Datenbank. Bevor nun das Rolling Upgrade durchgeführt werden kann, muss die logische Standby-Datenbank noch einmal auf ihre Konfiguration hin überprüft werden.

- Werden alle Objekte vollständig übertragen und/oder gibt es einen Workaround für Objekte, die nicht übertragen werden können?
- Sind `PROTECTION_MODE` und `PROTECTION_LEVEL` richtig gesetzt?
- Werden die Informationen von der Primärdatenbank korrekt an die Standby-Datenbank übertragen?
- Wurde die letzte Log-Datei erfolgreich übertragen?

```
ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

Abb. 4: Befehl zum Starten des SQL-Apply.

```
ALTER DATABASE PREPARE TO SWITCHOVER TO LOGICAL STANDBY;
```

Abb. 5: Die Primärdatenbank kündigt die Bereitschaft für den Switchover an.

```
ALTER DATABASE PREPARE TO SWITCHOVER TO PRIMARY;
```

Abb. 6: Vorbereitung der Standby-Datenbank auf den Switchover.

```
Primary → ALTER DATABASE COMMIT TO SWITCHOVER TO LOGICAL STANDBY;  
Standby → ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
```

Abb. 7: Switchover und Rollentausch.

```
ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE NEW PRIMARY  
dblink;
```

Abb. 8: Befehl zum Starten des SQL-Apply.

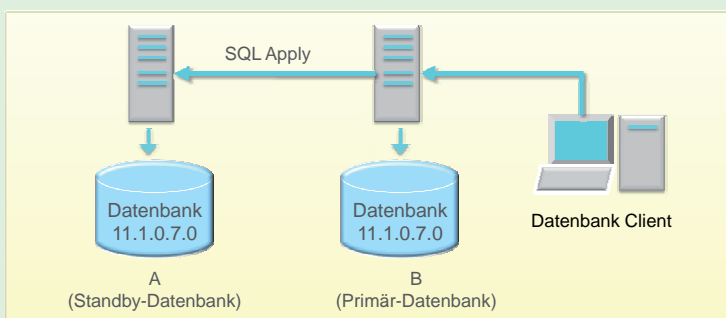


Abb. 9: Ausgangssituation.

Ablauf des Rolling Upgrade im Detail

Die Ausgangssituation ist Schritt 1 (siehe Abbildung 1). Nachdem die logische Standby-Datenbank erfolgreich implementiert ist, muss der SQL-Apply gestoppt werden (siehe Abbil-

dung 3). Die Standby-Datenbank wird heruntergefahren und das Upgrade wird auf Oracle 11.1.0.7 anhand der Oracle Patch-Dokumentation durchgeführt.

Danach wird der SQL-Apply wieder gestartet (siehe Abbildung 4). Zu diesem Zeitpunkt arbeiten beide Datenbanken mit unterschiedlichen Oracle-Versionen.

Mit Hilfe der View `DBA_LOGSTDBY_EVENTS` lässt sich überprüfen, ob der Ablauf innerhalb der logischen Standby-Datenbank fehlerfrei ist. Ein fehlerfreier Ablauf wird als letzter Eintrag in der View `DBA_LOGSTDBY_EVENTS` wie folgt dokumentiert:

```
ORA-16111: log mining and apply setting up
```

Im nächsten Schritt folgt der Switchover bzw. das eigentliche Rolling Upgrade. Als erstes kündigt die Produktionsdatenbank ihre Bereitschaft zum Switchover an (siehe Abbildung 5). In der Spalte `SWITCHOVER_STATUS` in der View `V$DATABASE` prüfen wir auf der Standby-Datenbank den Switchover Status der Standby-Datenbank.

Im zweiten Schritt wird die Standby-Datenbank auf den Switchover vorbereitet (siehe Abbildung 6)

Daraufhin kann der eigentliche Switchover sowie der Rollentausch beider Datenbanken durchgeführt werden (siehe Abbildung 7).

Nach dem Upgrade der nun neuen Standby-Datenbank kann der SQL-Apply mit Hilfe eines Datenbank-Links wieder gestartet werden (siehe Abbildung 8 und 9).

Möglichkeiten und Grenzen

Die Benutzung des logischen Data Guard bringt viele Möglichkeiten mit sich, beinhaltet aber auch Einschränkungen. Der entscheidende Vorteil ist, dass die Ausfallzeit während eines Upgrades einer Datenbank auf ein Minimum reduziert wird. Mit keinem anderen Verfahren von Oracle ist es möglich, diese minimale Ausfallzeit zu erreichen.

Die größte Einschränkung des logischen Data Guard ist, dass nicht alle Datentypen unterstützt werden. Aufgrund des SQL-Apply, der den Oracle `LogMiner` benutzt, werden nicht alle Datentypen übertragen, da der `LogMiner` nicht in der Lage ist, komplexere Datentypen nachzubilden.

Folgende Datentypen können in Oracle 11gR1 nicht übertragen werden:

- BFILE
- Collections (VARRAYS und Nested Tables)
- Multimedia Datentypen (Spatial, Image und Oracle Text)
- ROWID und UROWID
- Benutzerdefinierte Datentypen
- LOBs gespeichert als SecureFiles
- XMLType gespeichert als Object Relational
- Binary XML

Des Weiteren werden keine Objekte vom User SYS übertragen.

Fazit

Die Ausfallzeit minimiert sich bei der Anwendung eines Rolling Upgrades auf die Rollentausch- bzw. Switchover-Zeiten. Insgesamt kann als Ausfallzeit für das Switchover eine Zeitspanne von zwei Minuten eingerechnet werden, wenn alle Vorkehrungen getroffen werden. Das Upgrade-Verfahren der Standby-Datenbank läuft im Hintergrund ab und hat keinen Einfluss auf dem Betrieb der Primärdatenbank. Für Sie in der Praxis hat das vor allem den Nutzen, dass alle Anwender und Anwendungen nach nur sehr kurzer Zeit wieder produktiv auf die Datenbank zugreifen können. Dies ist mit kaum einem anderen Verfahren möglich.

Glossar

DML	Data Modification Language. Über DML-Kommandos werden Daten (Zeilen) in Tabellen eingefügt, gelöscht oder geändert.
Force Logging	Datenbankeinstellung, die garantiert alle DML-Operationen in den Redolog-Dateien protokolliert.
Redolog-Datei	Transaktionslog-Datei für Oracle Datenbanken, die zyklisch beschrieben und ggf. gesichert wird.
Supplemental Logging für Logical Data Guard	Primary Key und Unique Index Informationen eines Oracle-Datensatzes (nicht nur die betroffenen Spalten bei z. B. Spaltenänderung) werden in den Log-Dateien protokolliert.

Links

- ▶ [1] ORDIX News Artikelreihe „Oracle Data Guard“:
<http://www.ordix.de/ORDIXNews/artikelreihen.html#dataguard>
- ▶ [2] Seminarempfehlung „Oracle Migration und Patching“:
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/840/index.html>

Sie wollen mehr über das Rolling Upgrade oder andere Migrations- und Upgrade-Verfahren erfahren? Dann besuchen Sie das 3-tägige ORDIX Seminar „Oracle Migration und Patching“ [2].

Vanessa Prior (*info@ordix.de*).

Seminarempfehlung: Oracle Migration und Patching

▶ **Informationen/Online-Anmeldung:** <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/840/index.html>

In diesem Seminar werden Sie mit der Durchführung von Migrationen bei Oracle vertraut gemacht. Ein weiterer Schwerpunkt liegt auf der Vorgehensweise beim Einspielen von Patches.

Seminarinhalte

- Migration
 - Überblick Migrationsverfahren
 - Migration per exp/imp - Datapump
 - In-Place Migration (manuell oder per dbua)
 - Rolling Upgrade mit Data Guard
 - RMAN Convert
- Patches
 - Abgrenzung der verschiedenen Patchtypen
 - Vorgehensweise beim Patchen
- Praktische Übungen und Beispiele

Termine

22.03. - 24.03.2010 in Wiesbaden
03.05. - 05.05.2010 in Wiesbaden
06.09. - 08.09.2010 in Wiesbaden
02.11. - 04.11.2010 in Wiesbaden

Seminar-ID: DB-ORA-39

Dauer: 3 Tage

Preis pro Teilnehmer: 1.290,00 € (zzgl. MwSt.)

Frühbucherpreis: 1.161,00 € (zzgl. MwSt.)

Wir verbinden Welten

Wir schlagen die Brücke zwischen IT-Management und IT-Technik!

Unsere qualifizierten Consultants zeigen Ihnen die bedeutenden Trends in allen strategisch wichtigen IT-Technologien und unterstützen Ihr Business mit dem dazu passenden IT-Management.

Wir beraten, trainieren und begleiten Sie bei der Weiterentwicklung moderner IT-Strategien, im Enterprise Architecture Management und bei der Umsetzung Ihrer IT-Governance.

Und dies nicht nur auf der fachlichen, sondern ebenso auf der sozialen Ebene durch effiziente Kommunikationstrainings und pro-aktives Konfliktmanagement.

Sprechen Sie uns einfach an - um mit uns gemeinsam Brücken für Ihren geschäftlichen Erfolg zu bauen!



coniatos AG
Kreuzberger Ring 13
65205 Wiesbaden

Tel: 0611 77840-00
E-Mail: info@coniatos.de
Internet: www.coniatos.de